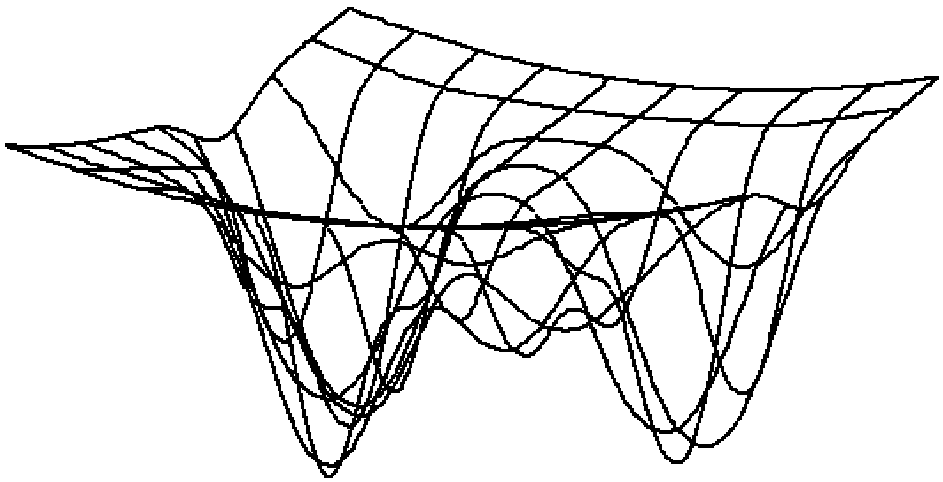
NEURAL RELAXATION DYNAMICS

MATHEMATICS AND PHYSICS OF RECURRENT NEURAL NETWORKS
WITH APPLICATIONS IN THE FIELD OF COMBINATORIAL OPTIMIZATION



JAN VAN DEN BERG

NEURAL RELAXATION DYNAMICS

MATHEMATICS AND PHYSICS OF RECURRENT NEURAL NETWORKS
WITH APPLICATIONS IN THE FIELD OF COMBINATORIAL OPTIMIZATION

(Neurale Relaxatie-Dynamica
Wis- en Natuurkunde van Recurrente Neurale Netwerken
met toepassingen op het terrein van de combinatorische optimalisering)

PROEFSCHRIFT
TER VERKRIJGING VAN DE GRAAD VAN DOCTOR AAN DE
ERASMUS UNIVERSITEIT ROTTERDAM OP GEZAG VAN DE
RECTOR MAGNIFICUS

PROF. DR P.W.C. AKKERMANS M.A.

EN VOLGENS BESLUIT VAN HET COLLEGE VOOR PROMOTIES

DE OPENBARE VERDEDIGING ZAL PLAATSVINDEN OP
VRIJDAG 21 JUNI 1996 OM 13.30 UUR

DOOR

JAN VAN DEN BERG
GEBOREN TE ROTTERDAM.

Promotiecommissie

Promotor:	prof. dr A. de Bruin
Copromotor:	dr J.C. Bioch
Overige leden:	prof. dr E.H.L. Aarts
	prof. dr ir R. Dekker
	prof. dr ir A. Nijholt

To Anneke

Contents

Preface

The background

Creating a thesis is no sinecure. As is often lamented, it is a project which takes much energy and a substantial amount of time. In fact, the time needed for producing this dissertation has faulted my expectations in two ways. On the one hand, the actual work took a rather short time: the starting point was about three and a half years ago when the discipline of neural networks, as yet unknown to me, made a vague but challenging impression on me. On the other hand, I must say that the time required has been considerable: almost four and a half decades of my life have passed in order to arrive at this point. As you might presume, many reasons can be given for this. Now, contemplating them, I think two issues have been all-important and, actually, conditions sine qua non for the realization of this work. The first one relates to the question how I have come in the position to collect enough *knowledge*, the second one relates to the case of how I got the opportunity to accumulate enough *self-confidence* in order to first start, and then to finish the project. Very many people have helped me in this process and I would like to thank some of them explicitly here.

Growing up along the borders of the river 'Wantij', I discovered many secrets of nature. My parents offered me much freedom in going my own way, in exploring and in finding out, using the things I came across. I was surrounded by many friends of my age. Besides having the usual games, we constructed large piers in the river used for swimming, fishing and mooring. By doing this, we learned as a matter of course the basic principles of mechanical engineering. At high school, certain teachers were able to strike the right note in order to rouse my love for mathematics and physics. I still remember the explanations on algebra and geometry by my mathematics teacher when I was 13 years old. Likewise, I still recollect the presentation by my teacher of physics on the differential equation of a simple harmonic motion

$$m\frac{\mathrm{d}^2u}{\mathrm{d}t^2} + cu = 0,$$

having a sinuous function as solution. Ever since, I have loved differential equations and, curiously enough, in a way the said equation plays a part in this thesis!

During my student times at the Technical University Delft, I was often more engaged on student politics and the social impact of science than mathematics and physics itself (it was in the seventies ...). Nevertheless, I was taught many basic principles of theoretical physics and mathematics. I also learned how computers

could be used, as it was done in those days. For my master's thesis, I worked in the field of numerical analysis, and again differential equations played a big part. My working career started in 1977, almost 19 years ago. Since that point of time, I stayed at various places (see my curriculum vitae) and I learned many different subjects. But, whatever my activities were, science continued to attract me. And fortunately, there have been many opportunities to augment my scientific knowledge. E.g., caused by the enormous automation in society, computer science started to cross my path more and more.

Looking back now, I might say I have been quite lucky to be able to constantly improve my knowledge and skills during my life. In relation to my scientific background, this process has taken place in three fields especially, namely, mathematics, physics and computer science, all of which have been indispensable for realizing this thesis. Moreover, I have been able to increase my self-reliance at the same time, although in a different way. I still remember very well the moment of finishing my master's thesis, when I did not feel strong enough to continue in research: scientific work seemed to be a privilege for other, smarter people. Besides, another even more difficult task announced itself: our first child was coming and would soon attract much attention and energy. But ever since, by these and other experiences – like during the Mozambican adventure – my self-confidence could grow, slowly but eventually to a sufficient measure. The intensive contact with so many colleagues, students, and, above all, friends have been a crucial factor here.

Acknowledgements

Of all people, I would like to thank you, Anneke, first. About 28 years ago, we became close friends and we still are. Of everyone, I am most indebted to you. We have lived to see incredibly many things together, with the creation of our family of 3 sons as undoubtedly the most wonderful experience and the most radical decision. Yet, you also gave me a wide berth for finding out much on my own. More specifically, referring to this thesis, you have seen all my moods on it, all progress, doubts, attempts and struggles, in other words, the whole weal and woe of this project. Thanks very much for everything! Next, I would like to thank my parents and my parents-in-law for giving much confidence and support during so many years, in spite of the numerous, in their eyes sometimes rather wild adventures I attempted. It is really a wonderful notion to have you at the ceremony, soon.

It is impossible to thank by name every one of the friends I encountered during my life. However, some can't possibly be passed over. I thank my friends from Delft, certainly Gerrit, Rob, and Romke, among other things, because of the exciting walking and sailing tours we made, and Gerrit together with Tineke on account of the many years of intense friendship. Particular thanks also go to Peter and Elly for the innumerable lovely hours experienced, first together and later separately, quite especially with Elly in Amsterdam. There, I also met other people within a large network of friends. Of all of those, I would like to thank Els for the enjoyable moments we witnessed. I thank Nannie and Raymond from Eindhoven for all the nice times together and our talks (including those using electronic mail). Finally, Marc, thank you very much for the many moments of discussion (and of silence!) during the years we occupied the same office-room, as well as for your

patience in improving my written English. I feel very happy about the immediate willingness of you and of Raymond to be my ushers during the ceremony.

Considering the actual realization of this PhD-study, I first and above all thank you, Cor, for having been my advisor and copromotor. Your critical and inspiring comments have been indispensable. They stimulated me to look for answers to new questions, to express myself clearly, and to improve my mathematical analyses. In particular, I remember our joint efforts to unravel the mathematical statements of the Simic's article [77]. I think your contribution at that time was the seed for the decisive break-through of the 'most general framework', later on. Second, I want to thank you, Jock, for the many discussions we have had during the work on your master's thesis [35] concerning the elastic net algorithms. Some of your ideas you will see again in this dissertation, although in a somewhat different version. The support of both of you has inspired me to use the word 'we' instead if 'I' almost everywhere in this thesis. You might interpret this as my feeling to be supported by either of you two during the process of entrusting the paper with my statements.

I thank Arie de Bruin for his willingness to be my promotor and for his critical comments on certain parts of this dissertation, as well as the theses belonging to it. I thank Emile Aarts for his spontaneous agreement to participate in the PhD-committee and for the thorough discussion on the contents of this work-piece. I thank Rommert Dekker for his confidence offered during the final phase of the creation of the manuscript. I further thank my old college friend Anton Nijholt: some 22 years ago, we worked intensively together in the students' movement, now you are one of my reviewers and opponents, once again showing the fact that life may house pleasant surprises. I further would like to thank Jan Brinkhuis and Joost Kok for their immediate willingness to oppose at the ceremony of the next 21-st of June. I thank Gert-Jan Lokhorst for his readiness to peruse this manuscript on statements concerning the logic of science. Hans de Bruin, thank you for making the 'style files' of your thesis available to me (it really saved me a lot of time and tiresome work), and Reino de Boer, thank you for explaining the intricacies of LaTeX I needed.

Finally, as I have promised you, Mark, Paul, and Erik, you find your names in this book. Dear sons, thanks for showing that you understood my engagement during the preparation of this thesis. I hope that, in times to come, we will have more unlaboured opportunities to see beautiful things together. Likewise, I wish with all my heart that some day, in one way or another, my work will be a source of inspiration for you.

Rotterdam, April 1996

Chapter 1

Introduction

This thesis refers to an analysis of various models of recurrent artificial neural networks and to how they might be applied in order to solve certain optimization problems. It therefore seems most appropriate to start by highlighting the position of the specialty neural networks among other areas of science, to present a historical sketch of its development, to explain what is actually meant by an artificial neural network, and to describe how it can be applied. We shall then shortly dwell upon the central theme of this study, by presenting a general mental image of the notion of relaxation and by explaining how this may take place in a recurrent neural network. Next, the general research objectives are formulated, including a short sketch of how the project got started and gradually evolved. The subject of the succeeding section is the methodology used. It also covers a justification of the chosen working-method. This introductory chapter is concluded by an exposition of the structure of the rest of the dissertation.

1.1 Artificial neural networks

1.1.1 Artificial neural networks and AI

Artificial neural networks (ANNs) are part of the much wider field called artificial intelligence (AI). AI can be defined as 'the study of mental faculties through the use of computational models' [23]. A related definition is that 'AI is the study of intelligent behavior' including 'the implementation of a computer program which exhibits intelligent behavior' [32]. In yet another characterization it is noted that 'the objectives of AI are to imitate by means of machines, normally electronic ones, as much of human activity as possible, and perhaps eventually to improve upon human abilities' [67]. An unavoidable difficulty of these and similar definitions is that they are always based on other notions whose precise meaning is hard to state[1]. E.g., in the second description, it is difficult to define precisely the notion

[1]This concerns a well known problem in science: definitions are always based on other notions. At a certain level, one should accept some 'primary' terms [53].

of intelligent behavior. Notwithstanding this, it is clear from the given definitions that usually, within AI, computers are applied to imitate the mental faculties of our brain which, among other things, comprises of vision, olfaction, language comprehension, thinking, reasoning, searching, remembering, learning, sensing, and controlling. Besides, the fundamental question arises: which modelling approach is chosen by AI researchers? Roughly speaking, two main streams can be distinguished in the ways AI is modelled[2], namely *symbolism* and *connectionism* [32]. The most fundamental difference between the two approaches concerns the *representation of knowledge*. In case of symbolism, this is done by using 'physical symbols'. In this approach, knowledge is represented and manipulated in a structured way, e.g., by means of a computer language like Prolog or Lisp. Logic plays a great part here, and classical expert systems are a well known example. On the other hand, in the 'connectionistic' approach, the representation of knowledge is numerical, where the weight values between the interconnected neurons (see below) represent knowledge in a distributed and generally unstructured way [54]. In this case, calculus and probability theory are important tools.

1.1.2 Inspiration from the biological brain

In quite a bit of AI research, the qualities of our brains are the source of inspiration. More specifically, within the study of ANNs, the way our cerebra are composed is directly taken into account: the biological neural network is imitated by an artificial one where certain architectonic elements of the cerebra are taken over. The following convenient brain characteristics are often put forward as reasons to study its workings [44, 79]:

- It is fault-tolerant: damage (to individual so-called neurons) can occur without a severe degradation of its overall performance.
- It is flexible: adjustment to a new environment is easily done through learning.
- It is highly parallel: many neurons process the (locally available) information simultaneously.
- It is anarchic: there is no specific area which controls the overall working of the brain and the neurons process the incoming information autonomously.
- It can deal with fuzzy, probabilistic, noisy, and even inconsistent information.
- It is small, compact, and dissipates little power.

Comparing the real brain and all man-made devices, it should be clear that any element of the latter group enjoys only a tiny subset of the brain properties mentioned above.

[2] In the background of the modelling problem, an intense philosophical discussion rages on what human intelligence actually is and, related to this fundamental question, on whether a machine like a computer can really have a mind (becoming apparent by, for example, the ability to 'feel' pain and pleasure). There exist various elaborated points of view on this intriguing subject some of which can be found in [26, 45, 61, 67].

The detailed working of the brain has been barely understood. Yet, during the last decades, both in the symbolic and in the connectionistic camp, many computational models have been proposed, which proved to be able to imitate certain elementary mental functions. The construction of those models is usually based on knowledge from many areas of science. E.g., in the area of natural language comprehension, specialists in linguistics, computer science and cognitive psychology make important contributions. In robotics, mechanical and electronic engineering play a big part. Constructing a theorem-proving device requires knowledge of mathematics, and building an expert system demands, besides knowledge of logic, the elicitation of quite a bit of 'domain knowledge' from experts in the field. When composing devices which can see or hear, one uses knowledge from physics, and when constructing an artificial olfactory organ, one requires knowledge of chemistry too. An example might give some idea of the variety of information that should be collected to construct a model with only one specific function. In the 'signal channelling attention network' for modeling so-called 'covert attention' (a certain, not overtly visible selective process of sampling the visual environment by the eye used, e.g., to select future targets for eye fixation), four different disciplines have been applied: biology (neurophysiology), psychology (psychophysics), physics (statistical mechanics), and computer science (parallel computation) [73].

In the *general* ANN approach, the focus is firstly mathematical: we try to catch the working of the brain in abstract mathematical models, which can be analyzed by means of mathematical specialties like dynamic systems theory, probability theory and statistics, or computational learning theory. Certain elements of the anatomy and physiology of our brain as studied in neurobiology act as source of inspiration in the modelling process, but they are, in general, merely points of departure. Theoretical physics is relevant in offering several well studied models which have proven to be useful, and computer science can be used to perform simulation studies on the ANN models in question. Last but not least, electronic engineering can be applied in order to construct, test, and apply (successful) ANN models in hardware.

Moreover, there is a reverse side to the coin. Artificial models of the brain often involve new paradigms and in their turn, may be adopted to solve (old and new) practical problems in a (completely) new way. Thus, in this way, nature shows us how to tackle difficult problems. This surprising, reversing effect may lead to a nice spin-off of the study of artificial intelligence. In fact, part of the study as described in this thesis exhibits an example of this recoiling effect: we have tried to solve combinatorial optimization problems in an alternative way using ANNs.

1.1.3 A brief historical sketch

During the early forties, abstract models concerning the working of a neuron were introduced [60]. A few years later, a law was proposed that explains how a network of neurons can learn [39]. Approximately at the same time, the symbolic approach was applied by scientists who made proposals on the construction and implementation of chess-playing computers (for a more detailed historical overview,

we refer to [27]). Another example of the symbolic way to grapple with AI, was the creation of a theorem-proving program [63]. Later, it was recognized that the logic-oriented approach of this program – precisely like in the event of chess-playing machines – should be broadened to a knowledge-based approach where, besides a certain inference strategy, the acquisition and representation of domain knowledge in a so-called knowledge base is considered to be crucial. The processing of this knowledge is performed by a separated inference engine and is symbolically oriented. In the mid-eighties, many expert systems having this architecture were constructed with the objective of simulating human experts intelligence.

In the mean time, the connectionistic approach had gone through a severe crisis. Often, the book of Minsky and Papert [62] (published in 1969) is taken as the root of all the trouble around connectionism in the seventies. It describes certain strong theoretical limitations of simple perceptrons (a class of certain ANNs). It also expresses the opinion that an 'interesting learning theorem for a multilayer machine' might never be found. Yet, some researchers persevered and in the eighties, neural networks returned to the scene. The backpropagation algorithm as popularized by Rumelhart et al. [75] has been an important stimulus just like the contribution by Hopfield using the idea (from physics) of energy minimization [46, 47]. A few years later, neural networks became a quite popular area of research with hundreds of conferences every year and the genesis of dozens of journals.

Due to the theoretical improvements, ANNs became a new tool in resolving practical problems. A functional classification yields four application areas [32], namely 'classification' (assignment of the input data to one of an (in)finite number of categories), 'association' (retrieval of an object based on part of the object itself or based on another object), 'optimization' (finding the best solution), and 'self-organization' (structuring received data). Within any of these classes, many subclasses can be distinguished, each in its own stage of development. Classification is probably the best-known and largest class with numerous application areas like speech recognition [21], handwritten digit recognition [58], control [6], prediction of time series, image compression, and others (for an overview, we refer to [44]). A collection of applications in the field of optimization and association will be given at the end of the next chapter.

Nowadays, the symbolic as well as the connectionistic camps have run up against certain barriers of their approach and seem more prepared to merge and also to integrate with other promising areas like genetic algorithms [36, 56] and fuzzy systems [54]. In a recent textbook [32], this tendency of integration is extensively described and illuminated with examples. The three fields neural networks, genetic algorithms, and fuzzy systems together are sometimes termed *computational intelligence* [29] (see further section 2.2.3).

1.1.4 An overview of ANNs

Nowadays, there are many textbooks[3] on ANNs, all using a certain taxonomy.

[3]The most important one for this thesis has been the book by Hertz, Krogh, and Palmer [44]. A classic is the book by Rumelhart et al. [75], another classic that of Hecht-Nielsen [41]. Still other general

Defining neural networks

The basic building block of all networks is a neuron (also referred to as a unit, node, processing element, or threshold logic unit). Various neurons are interconnected in differently organized topologies corresponding to different architectures. A central goal of ANN research is to understand the *global* behavior of a given ANN based upon the *individual* deportment of the neurons and its interconnections. A precise definition of an ANN is hard to give. The general definition by Hecht-Nielsen [40] (re-stated in [79]) gives several basic qualities:

> "A neural network is a parallel, distributed information processing structure consisting of processing elements (which can possess a local memory and carry out localized information processing operations) interconnected together with unidirectional signal channels called connections. Each processing element has a single output connection which branches ('fans out') into as many collateral connections as desired (each carrying the same signal – the processing element output signal). The processing element output signal can be of any mathematical type desired. All of the processing that goes within each processing element must be completely local: i.e., it must depend only upon the current values of the input signal arriving at the processing element via impinging connections and upon values stored in the processing element's local memory".

We would like to subjoin the following important aspect:

> A central issue in the employment of a neural network is the way how information is encoded in and retrieved from the neural system.

We now look more accurately at the working of an individual neuron. In the mathematical approach, a neuron is assumed to receive input signals, to add them together, and to generate an output signal using a given 'transfer' (or 'activation') function, also termed input-output characteristic. More precisely, if O_i represents the output of neuron i, I_i an environmental (or external) input, w_{ij} the 'interconnection strength' from neuron j to i, U_i the total input, and g the transfer function, then the new output value of the neuron is calculated via

$$O_i^{\text{new}} = g(U_i^{\text{old}}) = g(\sum_j w_{ij} O_j^{\text{old}} + I_i). \tag{1.1}$$

The vector $O = (O_1, O_2, \ldots, O_n)$ is often called the *system state* of a neural network having n neurons. From (1.1) we see that the signals incoming from other neurons are weighted[4].

In the first model by McCulloch and Pitts [60], the transfer function is a binary threshold unit. The equation (1.1) can then be rewritten as

$$O_i^{\text{new}} = \Theta(\sum_j w_{ij} O_j^{\text{old}} + I_i \Leftrightarrow \mu_i), \tag{1.2}$$

books on ANNs are available, for example, [30, 32, 38, 54, 79, 82].

[4] In neurobiological terms, a weight w_{ij} represents the 'strength of the synapse' connecting neuron j to neuron i [44].

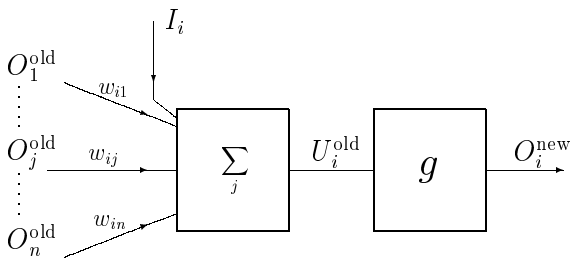


Figure 1.1: A scheme of an artificial neuron.

where μ_i is the local threshold value of neuron i and Θ is the unit step or Heaviside function defined conform

$$\Theta(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{1.3}$$

Other choices for the transfer function [44] include linear functions and non-linear functions like the sigmoid function (section 2.1.5). Even stochastic transfer rules are possible (section 2.3.3).

Having defined the transfer function, we must still choose a rule for the updating sequence of the neurons [44]:

- Asynchronous updating: one unit at a time is selected and its output value is adjusted according to equation (1.1).
- Synchronous updating: at each time step the output of all neurons is adjusted according to equation (1.1).
- Continuous updating: the output values of all units are continuously and simultaneously adjusted, while at the same time the inputs change continuously.

The last updating strategy will be discussed in section 2.3.2.

A taxonomy

Two basic criteria are often used to categorize ANNs. The first one concerns the way the signals propagate among the neurons [79, 44]. In a *feedforward* scheme, information is only allowed to flow in one direction without any back coupling. This implies that the output of the network is uniquely determined given the weights w_{ij}, the transfer function in the neurons, and the external inputs of the neural net. These networks are often structured in 'layers'. A one-layer feedforward network is called a perceptron. *Feedback* networks on the other hand, allow information to flow among neurons in either direction, implying that such a net needs not necessarily be in equilibrium nor that an equilibrium state is uniquely determined. It is even the case that these *recurrent* networks do not necessarily settle down to

a stable state. However, we shall confine ourselves to study those networks that find an equilibrium state via a so-called relaxation process.

The second fundamental criterium concerns the way the network learns. *Supervised* learning is a process that incorporates an external teacher and / or global information. A network is considered to learn if the weight matrix (w_{ij}) (sometimes called the networks 'memory') changes in time, mathematically expressed as

$$\exists i, \exists j : \dot{w}_{ij} \equiv \frac{\mathrm{d}w_{ij}}{\mathrm{d}t} \neq 0. \tag{1.4}$$

In *unsupervised* learning there is no teacher. The network must discover patterns, regularities and so on by itself. There should be a form of built-in self-organization.

Using the two criteria, four types of networks can be distinguished. We limit ourselves to comment on two of them (more details can be found in the aforementioned textbooks). The most popular network is probably the supervised, feedforward type. The mostly applied learning rule is called backpropagation: using a set of correct input-output pairs (called the training set), small changes in the connections w_{ij} are made in order to minimize the difference between the actual and the desired output value of every training example. In this way, the learned stuff is fixed in the weight values w_{ij} in a distributed way. All training examples have their contribution to all final weight values, but in the end, it is unclear what every individual weight precisely stands for: that is why we say the representation of knowledge in ANNs is 'unstructured'. Afterwards, it is hoped that the network can 'generalize' what it has learned: the network should also find the correct output for an input *not* belonging to the training set. Function approximation and pattern recognition are the common general applications[5], while the central points of theoretical study are learning, generalization and 'representation' [44, 84]: the representation problem concerns the question what type of function can be represented (and therefore might be learned) by a feedforward network of given architecture. Besides the afore-mentioned popular type, there exist many other supervised, feedforward models.

The second type we dwell on is the unsupervised, recurrent network type. Neglecting the many other examples of this type, the binary, the continuous, and the stochastic Hopfield models belong to this category. The models are called unsupervised since the matrix (w_{ij}) is fixed at the beginning (using global information in one way or another) and is never changed[6]. The Hopfield models are the main subject of our study. They will be introduced formally and discussed extensively in the forthcoming chapters. Application areas of these networks are (memory) association [44] and optimization, within a growing number of specialties (section 2.4). In the next subsection, we confine ourselves to present an intuitive idea of the working of these models.

[5] These types of applications can be considered subareas within the general class classification of section 1.1.3.

[6] Other recurrent models like the Boltzmann machine [44], do include learning besides relaxation. Our findings may also be applicable to the 'relaxation phase' of those networks.

1.1.5 A mental image of relaxation in neural networks

Let us put aside all mathematical notations and concentrate on the general idea behind the working of the Hopfield and allied models. We shall use a metaphor originating from the world of physics which, in fact, makes real sense as will be exposed later.

We imagine having a laboratory table with many magnets of various strengths on it, whose initial direction can be adjusted as desired. It is further supposed that the magnets can freely rotate after pulling over a lever. All magnets have their own magnetic dipole field around them. If the lever is pulled over after having initialized the magnets in a randomly chosen direction, they will start rotating under the influence of the mutual magnetic field forces. By the movement of the magnets, the structure of the magnetic field constantly changes. The result is a complex *deterministic dynamic system*. If we further suppose that energy dissipates in some way (e.g., by friction and-or air resistance forces), the system will spontaneously 'relax' to an equilibrium state after a certain lapse of time.

Since the strengths of the local magnetic forces vary and there are very many magnets, it is not unreasonable to suppose that there exist more than one different equilibrium states of the system. Depending on the initialization of the direction of all magnets, the system will find an equilibrium point, namely the 'nearest' one. Stating the relaxation dynamics in mathematical physical terms, we say that the system minimizes potential energy and settles down in that *local* minimum state, which can simply be reached via a route 'downhill', away from the random initial state. In figure 1.2, the process of energy minimization to a local minimum is vi-

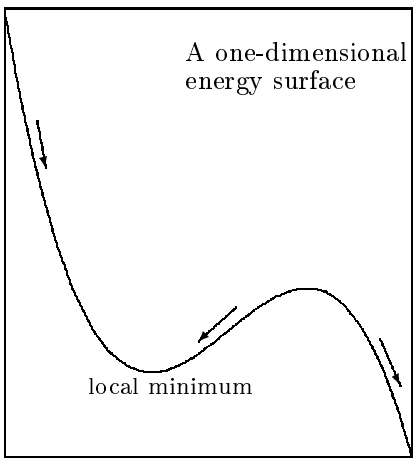


Figure 1.2: Energy minimization to a local minimum.

sualized. All system states are supposed to lie along the horizontal axis, while the arrows denote the direction of the minimization process along the energy surface.

It has been Hopfield's merit that he observed that the relaxation dynamics of his recurrent continuous neural network – itself a generalization of a certain binary model – can be described by a type of deterministic process as discussed above. It is probably also obvious that these ANNs can be useful in concrete optimization applications, where a certain cost function should be minimized[7]. One should merely choose a neural network whose energy function coincides with the given cost function, initialize the network in one way or another, and then allow it to relax: the final (equilibrium) state encountered is hoped to correspond to a (or the) solution sought. This is the idea in a nutshell. However, in practice things are generally much more complicated:

- In the first place, we are mostly interested in the *global* minimum of a cost function, not in some local one. One way to solve this problem is to introduce thermal fluctuations in the system by making the magnets *stochastic*. To illustrate, we suppose that any magnet has only two opposite positions, one with the magnetic north pole to, lets say, the right, and one with that pole to the left. The actual position of a magnet depends on two factors, namely on the current total magnetic field force (as caused by all other magnets) as well as on the value of the current temperature in the system. All magnets have a certain freedom in fluctuating randomly[8] controlled by the value of the temperature: the higher the temperature is, the more a magnet randomly fluctuates. Lowering the temperature has the effect that all magnets are more driven by the locally existing magnetic field forces. Looking at the dynamic relaxation process of this stochastic system after a randomly chosen initialization, we observe that at high temperatures, the system behaves randomly. However, at lower temperatures the system will relax to another so-called *dynamic equilibrium*: the magnets may still fluctuate but on average, they will prefer one direction over the other. Furthermore, owing to the random fluctuations, the system is more or less disposed to relax to the global minimum by kicking out encountered local minima. It should be clear that this stochastic magnetic field system is even more complex than the deterministic one described earlier.

- The second complicating factor is related to the first one. Practical problems are generally defined in a high-dimensional space where the minima lie widely scattered around. In fact, the picture on the cover slightly lifts the veil of this very complicated hilly world, although the image merely shows a two-dimensional landscape. Under high-dimensional circumstances, it is much more difficult to imagine how the addition of thermal fluctuations might achieve a relaxation to the global minimum.

- In the third place, practice may be unruly since solutions of problems are often submitted to a certain set of *constraints*. Among other things, this is often the case in the field of combinatorial optimization problems. There are several ways to deal with this phenomenon. The eldest approach applied in neural networks is the penalty method, but it did not turn out to be

[7] This approach has also been pioneered by Hopfield, in co-operation with Tank [48, 49].
[8] Random fluctuations correspond to so-called thermal noise: see section 2.1.2.

very successful (see chapter 2). Another way is trying to incorporate the constraints in the neural net, which appears to be possible in some cases. This approach is the subject of chapter 4. In chapter 5, we shall encounter still other techniques among which modifications of the well-known mathematical method using Lagrange multipliers.

- To conclude, we should note that the actual mapping of combinatorial (optimization) problems onto ANNs is far from trivial: in practice, there appear to be many ways to realize such a mapping, each one having its own benefits and drawbacks.

We conclude the metaphor as given in this section, by remarking that a stochastic Hopfield neural network turns out to behave like the sketched stochastic magnetic system. Moreover, it can be approximated by a certain, slightly adapted, continuous, and deterministic model. In both cases, the neurons in the Hopfield models correspond to the magnets in the magnetic counterpart models. The approximation of the stochastic system by such a deterministic system is an important topic of the chapters 3 and 4.

1.2 Research objectives

Contemplating the ways in which science is exercised, we can distinguish several approaches. Even within a specific area of science, one often encounters substantial differences concerning methodology: research can be either fundamental or applied, either explorative or mapped-out in advance, either inductive or deductive, etcetera. Additionally, the objectives of the research project at hand are often formulated *a posteriori,* that is, after having completed the actual work on it. Realizing these aspects, it seems appropriate to first touch upon the evolution of this research project before stating its objectives and justifying the methodology selected.

1.2.1 History of the research process

This study on the relaxation dynamics of recurrent neural networks started in the autumn of 1992. Actually, there was no explicit objective of study at that time. There was a paper [51] originated from a master's thesis which reported promising results with respect to a new way of tackling the travelling salesman problem using two Hopfield type neural networks. We further read the relevant parts of the textbook [44] on Hopfield networks which gave rise to certain questions, and we encountered the book of Takefuji [82] containing some theory and a lot of applications. Very soon, we hit upon certain inconsistencies which begged for a solution. At that time, we also found two articles [71, 86] concerning the use of Lagrange multipliers in combination with neural nets which seemed not to get the attention that they deserved after their publication. Eventually, the study and elaboration of all this led to several new results, among which the notion of a *dy-*

namic penalty method. Another consequence was the realization of our first publications[9].

During that initial period, which also included some studies on combinatorial optimization problems like the travelling salesman problem (TSP), the idea took form to study the general relationship between Hopfield neural networks and the so-called elastic net, the last one being a neural network especially set up to solve the TSP. We studied Simic's paper [77] (referred to in [44]) and other relevant ones, and were promptly engrossed in a process of profound investigation. Simic's article appeared to contain many hard results, although most of the proofs were only sketchy. To understand the details, we were forced to completely work out the derivations. Calculus was the general tool of analysis. In addition, dynamic systems theory and statistical physics appeared to be of high importance: the first one in order to study the stability of the relevant differential equations, the second in order to exploit existing knowledge on certain thermodynamic models, which have a close connection to the ANN models of our study[10]. This ultimately yielded many new theorems, including the ones relating to a quite general result of this thesis namely *the most general framework* of continuous Hopfield models. In addition to all efforts in the theoretical field, we performed several simulations whose computational results will be reported for a substantial part[11]. This part of investigation also led to several (international) presentations and publications, whose series does seem still not be exhausted.

Finally, a new master's thesis project was undertaken yielding a new analysis of the elastic net algorithm [35]. It turned to fit in precisely with our theoretical experiences. It also contained (and further inspired us to try) various alternative elastic net algorithms.

1.2.2 The aims of this study

From the sketch given above, it is clear that the precise subject of *what* to study and all reasons *why*[12], were not plain from the beginning. Instead, these insights evolved gradually. Initially, the driving force was above all to understand why the relevant models behave like they appear to do, particularly, when they are used to solve combinatorial optimization problems. Very soon, the wish emerged to solve certain inconsistencies we came across. Next, we wanted to extend existing theories, e.g., on the stability properties of the so-called Hopfield-Lagrange model. Finally, it turned out to be possible to generalize existing theories on Hopfield and allied networks, both on the set of equilibrium conditions and on the stability of the corresponding differential equations. Above all, the analysis has been mathematical and physical. During the whole period, we tested whether the models of study could be applied to solve combinatorial optimization problems in an ade-

[9] The references to our publications will be made more precise in the succeeding chapters.

[10] Simic [77] expresses this relationship in the following nice way: the ANN "algorithms are in a deeper sense an example of what one may call a 'physical computation' ".

[11] Besides a lot of encouraging results, the experimental outcomes indicate certain limitations concerning the general applicability of the framework.

[12] At least one reason was obvious from the beginning namely, getting a Ph.D., a not-unimportant by-product of all research efforts.

quate way. These considerations taking together, we can now, a posteriori, define the objectives of this research project as follows:

- The main objectives of this thesis are to *explain*[13] the relaxation dynamics of various recurrent (more precisely, Hopfield and allied) neural network models, and to *generalize* existing theories on them.
- The secondary objectives are more diverse:
  - the first one is to verify how the discovered theoretical results can be used to reveal the relationship between Hopfield neural networks and the elastic net;
  - the second one is to test whether the studied models can be applied to solve certain combinatorial (optimization) problems in an adequate way.

From the arguments given in this subsection, the choice of the title of this thesis should be obvious.

1.3 The chosen methodology: an apology

After having sketched the 'what and why' of this thesis, I[14] consider it proper to describe the 'how' of it too, or, stating this in other words, to illuminate the chosen methodology in the light of what is often referred to as the 'logic of science' [87]. As is often done in this philosophic domain, we shall differentiate between two aspects, namely, the so-called internalistic point of view and the broader externalistic one.

1.3.1 The externalistic view

In the externalistic view on how science originates, it is considered essential to include a sociological analysis of the scientific process: e.g., what are the motives, driving forces, beliefs, prejudices, and so on of the scientists involved. Furthermore, one should investigate the 'context of discovery', i.e., what is the scientific culture of the area of science at hand. Another fundamental aspect to look at is the 'context of justification', which concerns the ways how given theories are justified. This more restricted approach is often termed the internalistic view, and it will be discussed in a separate section.

The philosopher Kuhn, who is considered an externalist, distinguishes two types of periods during the growth of scientific knowledge [57]: after a 'preparadigmatic' period, revolutionary and non-revolutionary stages succeed each other. In a revolutionary period, inconsistencies lead to a rejection of the older

[13] The notion of 'scientific explanation' is far from simple. Hempel and Oppenheim [42, 87] have formulated four conditions to call an explanation 'adequate': (a) what is explained should follow on logical grounds, (b) the explanation should use other laws, (c) the explanation should have testable consequences, and (d) the explanation should be true.

[14] Throughout this dissertation, I use the word 'we' for reasons as explained in the preface. This subsection is an exception, because the chosen methodology is the one specifically selected by me.

theory, that is, the older 'paradigm', and to the formulation of a new one. In non-revolutionary periods, inconsistencies encountered are either simply ignored or, otherwise, adapted to the paradigm accepted everywhere.

Holding my exertions against the light of these considerations, it becomes clear that it is not easy to give an unprejudiced judgement of my own. I myself join in the neural network community, have been affected by it, and may even be indoctrinated, so perhaps, I am not aware of certain untenable starting points, motives, or ideas. However, in spite of these imperfections, some general and some personal externalistic aspects can be observed. Let me first consider some general sociological issues. The central premiss of all AI-research seems to be the belief that human intelligence can somehow be (partially) modelled, using scientific means. This matter is strongly related to the philosophic debate mentioned above in footnote 2. To illustrate, a group of researchers believes that human thinking is in fact *algorithmic*, implying that, in principle, it can be emulated by a machine like a computer. On the other hand, there are other groups of scientists who firmly oppose against this 'strong AI' point of view, arguing, in one way or another, that the human mind is more than 'just a collection of tiny wires and switches' [33].

Considering the actions of the ANN community, I observe that the natural sciences mathematics and physics are judged to be very helpful to model the capabilities of the human mind. This belief has certainly been enforced by the success of some ANN models (showing certain elementary brain abilities), which appeared to be analyzable by means of mathematical physical models. Yet another common belief in the community of AI is the idea that following the realized applications, the research efforts will ultimately yield a lot of new practical applications and thus, on that account, do make sense. In that manner, scientists may find a legitimation for their work. The afore-mentioned beliefs can be considered to belong to the context of discovery. Together with other ideas on ANN models, they are exchanged between researchers in the usual ways: papers, journals, and books are published, discussions and talks are held at conferences. Owing to this, the various paradigms of ANN theory have assumed a well-defined shape, each one having its advocates and followers.

Considering my personal motives, I have already given an important one in footnote 12. Another personal reason for choosing the subject of study of this thesis is that it precisely corresponds to a lot of foreknowledge I picked up during my life. In relation to the context of discovery, I think that I have adjusted myself considerably to the current research customs in the area of neural networks by reading much of the standard literature, by doing the same type of research, and by presenting my results in the normal ways, both orally and in writing, to the relevant research community.

Let me finish this section on the externalistic view with a personal opinion. In the light of Kuhn's philosophy, I feel that nowadays neural networks are in a non-revolutionary stage of research. The discipline has several well-established basic models and, in general, scientists are busy applying, refining, and understanding them. I hold that the statements I bring out in this thesis, are refinements, improvements, illuminations, corrections, and generalizations rather than paradigmatic revolutions. However, it will be other people who must decide this issue.

1.3.2 The internalistic view

The pure 'reconstruction' of what has happened – including the justification of the scientific results – is the central theme in the internalistic view on the growth of scientific knowledge. The context of discovery is not considered here. Instead, a scientist is thought to assume a more idealistic attitude: in the view of Popper [72], a theory is proposed, and thereupon, it is tried to 'falsify' it. Precisely falsifications increase scientific knowledge. It is impossible to establish absolute truth of any theory. Theories are 'conjectures' which should be refuted, if there are reasons for it. Let I consider my working-method in this view of Popper. I first try to describe the method itself.

1. Any research effort on a new subject started by the collection of 'the relevant' papers.
2. Depending on what was found, I tried to analyze, improve, correct, generalize, apply, etcetera, inspired by mainly mathematical and physical ideas as evoked in my head and as available in 'the relevant' literature. The endeavors took place in at least the following ways:
   - The technique I probably applied most was to make up an as simple as possible example corresponding to an encountered abstract mathematical expression. Analyzing this simple 'toy problem' by means of notions of calculus (sometimes supported by graphical software packages) and physics, I tried to understand the essence. In this inductive way, intuition grew and new ideas were born, in turn leading to the suggestion of new theorems and insights. Of course, these new discoveries had to be proven.
   - Sometimes, I had got already a new insight without being able to prove it[15]. Often, a lot of trial and error was necessary to find the explicit proof.
   - Another trick which I applied several times, was to switch between the mathematical and the physical point of view, especially at points where the one way seemed to come to nothing.
3. Finally, the acquired insights and stated theorems had to be tested. This has again been done in several ways:
   - By re-inspection of the derivations: I must admit to have found many self-made errors this way.
   - By making up several simulation studies: I wrote many computer programs to inspect (the consequences of) my suggestions[16]. Also

[15] Compare the pronouncement by Gauss: "Meine Resultate habe ich längst, ich such' nur noch den Weg dazu" (restated in [87], p. 57.) or the observations by Penrose [67], in a discussion on the non-algorithmic nature of mathematical insight: "Rigorous argument is usually the *last* step! Before that, one has to make many guesses, and for these, aesthetic convictions are enormously important – always constrained by logical argument and known facts".

[16] Some groups of computer scientists argue that the correctness of a computer program should always be proven using notions from mathematics (especially from logic). This is not a redundant luxury: in computer science, a famous phrase states: "There is always a last bug". Ironically, in this study I did the opposite, namely, testing my mathematical theories using computer programs.

in this way, I encountered many mistakes.

– Last but not least, I submitted papers and reports to colleagues, both on my own department and on scientific conferences and journals.

Looking back now, I see some limitations of my working-method. Whether I found all 'relevant' papers, is very doubtful: the quantity of published papers in proceedings, journals, books is overwhelming, even within a specialized discipline like recurrent neural networks. So, I possibly missed certain relevant papers and results.

The second point of the described working-method relates to the conjectures of Popper. Even the mathematical theorems are conjectures: I frequently made mistakes, some of which remained unnoticed for several weeks or so. Some probably still exist. However, I am not unique: as will be explained, I have met several confusing mistakes by others, which engaged me for a substantial amount of time and, simultaneously, inspired me to develop new conjectures. In fact, for a substantial part, the new conjecturing statements of this thesis have grown out of mistakes as made by others.

The last point, the testing phase of my working-method, consists of the efforts to falsify my theories. Of course, I tried to find every error, and to what extent I have been successful, is hoped to become clear in the near future. Actually, so far as my attempts have failed, my conjectures still stand up, or, stating this in other words, my as yet non-falsified conjectures are the body of this thesis[17].

1.4 The outline of the rest of the story

We finish this chapter by explaining the structure of the remainder of this thesis. In the next chapter, the theoretical starting points are given. It consists of a general sketch of theoretical results collected from the technical literature, that together are considered to form the necessary background and basis of the subsequent four chapters: the relevant ANN models will be introduced, preceded by an introduction on statistical mechanics and succeeded by an overview of example applications. The foundations as given in chapter 2 are related to a collection of mathematical techniques. These ones are described in the appendices, the last one of which consists of a list of applied lemmas including their proofs.

Chapter 3, 4, 5 and 6 constitute the kernel of this thesis. We start by analyzing the so-called unconstrained stochastic Hopfield neural network and relate it to the classical continuous Hopfield model. Since the mathematical functions involved are rather complex, we use a separate section to illuminate their properties by means of some simple examples. Next, an interesting part of chapter 3 opens up where for the first time a more general framework is presented. Chapter 4 deals with a certain constrained Hopfield model. Surprisingly, it can be analyzed in the same way and it can also be generalized. The apotheosis is the aforesaid most

[17] Another logical consequence of this way of thinking is that, as far as my statements are correct, the contents of this thesis can held to be trivial.

general framework, where Hopfield networks are generalized to networks which can model almost arbitrary energy expressions (instead of merely quadratic ones) and which provide means for incorporating new types of constraints in the network. However, experimental outcomes also show certain limitations of the general framework.

Chapter 5 treats the Hopfield-Lagrange model. Most of this chapter is devoted to an analysis of the stability properties of the model. First, a potential Lyapunov function is defined by means of which in certain cases stability of the unconstrained Hopfield-Lagrange model can be proven. Second, stability of the quadratic and allied constraints is demonstrated in a quite different way. In that case, the model generally 'degenerates' to a type of the afore-mentioned dynamic penalty model. Next, the theorem on the potential Lyapunov function for the unconstrained Hopfield-Lagrange model is widened to one for the generalized constrained model.

In chapter 6, the investigations concerning the elastic net are presented including its relation to the constrained Hopfield model of chapter 4: the surprising outcome is that also the elastic net algorithm can be considered as a special type of dynamic penalty method. A further analysis leads in a natural way to two alternative elastic net algorithms, which are investigated too.

Chapters 3, 4, 5, 6 all conclude with a few relevant computational results of certain toy problems and applications tested. If the outcomes did not turn out straightforward, corresponding comments are added. Finally, in chapter 7, we draw our conclusions, discuss them, and do suggestions for future research.

Chapter 2

Starting points

In this chapter, the relevant theory of Hopfield neural networks will be sketched. This theory constitutes the starting point of the explorations described in the rest of this thesis. Before coming to the heart of the matter, a review of statistical mechanics is given: the theory about this subject turns out to be of great importance for the understanding of stochastic Hopfield networks. We also present an introduction on combinatorial optimization, the challenging application area. In the succeeding chapters, we shall return to many aspects mentioned here.

2.1 Statistical mechanics

2.1.1 The basic assumption

The main goal of statistical mechanics [64] is to derive the macroscopic, i.e., physically measurable, properties of a system starting from a description of the interaction of the microscopic components like atoms, electrons, spins. If we would take the classical approach using a Hamiltonian system[1], this would normally be an impossible task: the huge number of microscopic components leads to a comparable huge number of motion equations which cannot be solved practically. What we need is a statistical approach yielding simpler models, which hopefully still include the essential physics and are tractable to analytic or numerical solutions [89]. To reach our goal this way, two subproblems can be distinguished: (a) Find the probability distribution of the microscopic components in thermal equilibrium. (b) Derive the macroscopic properties of the system using this probability distribution.

Limiting our discussion to a discrete configuration space (meaning that the space of all possible system states is countable), the basic assumption of statistical mechanics concerning subproblem (a) is that in thermal equilibrium – that is, after

[1] In certain circumstances, the quantum mechanical approach resolving the Schrödinger equation would be another, mostly non-adequate alternative.

a sufficient long time – any of the possible states α occurs with probability

$$P_\alpha^{\rm eq} = \frac{1}{Z} e^{-\beta H_\alpha}. \tag{2.1}$$

Here, H_α is the total energy, called the Hamiltonian, of the system and Z is a normalizing factor, called the partition function, which equals

$$Z = \sum_\alpha e^{-\beta H_\alpha}. \tag{2.2}$$

Equation (2.1) is called the Boltzmann formula or Boltzmann equilibrium probability distribution. The value of β is related to the absolute temperature T by

$$\beta = \frac{1}{kT}. \tag{2.3}$$

The constant k represents the Boltzmann coefficient, and, because it is only a scaling factor, we can set it equal to 1.

Knowing the energy H_α in every state, equation (2.1) can be used to calculate the 'thermal average' $\langle A \rangle$ of any observable quantity A by application of

$$\langle A \rangle = \sum_\alpha P_\alpha^{\rm eq} A_\alpha, \tag{2.4}$$

where A_α represents the particular value of A in state α.

Equation (2.1) will not be justified here. It can be made plausible from very general assumptions on the microscopic dynamics of the particles [64] or, in a discrete energy space, be derived from a calculation of the most likely distribution of the particles over the various energy levels [4].

2.1.2 The free energy

It turns out very fruitful to define the so-called free or effective energy F by

$$F = \Leftrightarrow \frac{1}{\beta} \ln Z. \tag{2.5}$$

Using most of the above given equations as well as the fact that $\sum_\alpha P_\alpha^{\rm eq} = 1$, an important relation can be obtained [44]:

$$\begin{aligned} F &= \Leftrightarrow \frac{1}{\beta} \ln Z = \Leftrightarrow \frac{1}{\beta} \sum_\alpha P_\alpha^{\rm eq} \ln Z \\ &= \Leftrightarrow \frac{1}{\beta} \sum_\alpha P_\alpha^{\rm eq} (\Leftrightarrow \beta H_\alpha + \beta H_\alpha + \ln Z) \\ &= \sum_\alpha P_\alpha^{\rm eq} H_\alpha + \frac{1}{\beta} \sum_\alpha P_\alpha^{\rm eq} \ln \frac{e^{-\beta H_\alpha}}{Z} \\ &= \langle H \rangle \Leftrightarrow \frac{1}{\beta} \mathcal{S}^{\rm eq}, \end{aligned} \tag{2.6}$$

where $\langle H \rangle$ equals the thermal average of the Hamiltonian and where

$$\mathcal{S}^{\mathrm{eq}} = \Leftrightarrow \sum_{\alpha} P_{\alpha}^{\mathrm{eq}} \ln P_{\alpha}^{\mathrm{eq}} \tag{2.7}$$

is the 'entropy' at thermal equilibrium, which appears to be a measure of the disorder of the system.

Equation (2.6) is derived under the assumption that the system is in thermal equilibrium described by the probability distribution (2.1). Instead, we now consider F as a function of an *arbitrary* probability distribution $P = (P_1, P_2, \dots)$ given by

$$\begin{aligned} F(P) &= E(P) \Leftrightarrow \frac{1}{\beta}\mathcal{S}(P) && (2.8) \\ &= \sum_{\alpha} P_{\alpha} H_{\alpha} + \frac{1}{\beta} \sum_{\alpha} P_{\alpha} \ln P_{\alpha}. && (2.9) \end{aligned}$$

From this equation, a variational principle[2], called the principle of minimal free energy, can be derived [64] which states that a minimum of $F(P)$ corresponds to the equilibrium probability distribution (2.1). The proof is based on the Lagrange multiplier method (appendix A) taking as the only constraint $\sum_{\alpha} P_{\alpha} \Leftrightarrow 1 = 0$. The principle of minimal free energy (which is strongly related to the famous principle of maximal entropy[3]) hands us a tool for calculating a stable equilibrium probability distribution at given temperature $T = 1/\beta$: we 'only' need to find the location of the minima of $F(P)$.

From (2.1) it follows that the equilibrium probability distribution is a function of both the energy levels H_{α} and the temperature T. It is sometimes said that the free energy 'knows about the (thermal) noise' of the system, i.e., it 'depends in a non-trivial way on the temperature' [77]. From (2.9) we conclude that at high enough temperatures, $F(P)$ is generally dominated by the second term of the right-hand side. This term appears to be a smooth and convex [25] function of P and will be called the thermal energy of the system. Thus, in circumstances of high temperature, $F(P)$ has only one minimum and the equilibrium probability distribution is (almost) uniform. On lowering the temperature, the thermal energy decreases and the free energy becomes more and more dominated by the first term of the right-hand side of (2.9) implying that, at thermal equilibrium, the system will have settled down in states of lowest energy H_{α}.

[2] The calculus of variations is concerned with maxima and minima of *functionals*, where a functional is defined as a function $J : \Omega \to \mathbb{R}$, Ω being a space of functions [7].

[3] The principle of maximal entropy, the second law of thermodynamics, holds for isolated systems, i.e., systems which have not any thermal interaction with their environment. Instead, the minimum of free energy holds for systems whose temperature is kept fixed via heat exchange between the system and its environment: the system is contained in a 'heath bath' of constant temperature. Both entropy and free energy are 'thermodynamic potentials'. The extreme values of these potentials are 'attractors' to which the corresponding thermodynamic systems spontaneously evolve [74].

2.1.3 Spin glasses

Statistical mechanics has been applied successfully to a large variety of systems. In the context of our future discussions on neural networks, the techniques as used for the study of so-called spin and other glasses – these are certain types of more or less disordered magnetic systems – appear to be extremely relevant: the analysis and understanding of Hopfield neural networks is strongly facilitated by the theory on these magnetic systems.

The microscopic elements of spin glasses are elementary atomic magnets, so-called spins, fixed in location but free to orient, interacting strongly but randomly with one another through pairwise forces [76]. In so-called Ising models, the magnetic orientation S_i of any spin i is supposed to be binary, where $S_i \in \{\Leftrightarrow 1, 1\}$. If n is the number of spins, the Hamiltonian of the magnetic system is defined as

$$H(S) = \Leftrightarrow \tfrac{1}{2} \sum_{i,j \neq i} w_{ij} S_i S_j \Leftrightarrow \sum_i I_i S_i, \tag{2.10}$$

where $S = (S_1, S_2, \ldots, S_n)$ is a global microstate. The w_{ij}'s correspond to contributions from pair-wise magnetic forces and the I_i's represent external magnetic field values. Adding up the magnetic force contributions from all the other spins together with the external magnetic force, the total local magnetic field h_i for spin i equals

$$h_i = \sum_{j \neq i} w_{ij} S_j + I_i. \tag{2.11}$$

Instead of taking $S_i \in \{\Leftrightarrow 1, 1\}$, we shall adopt $S_i \in \{0, 1\}$ throughout this thesis because this will facilitate the mapping of combinatorial optimization problems on Ising models[4].

Substitution of (2.10) in (2.2) yields as partition function of the spin glasses system

$$Z_{\mathrm{sg}} = \sum_S \exp(\beta(\tfrac{1}{2} \sum_{i,j \neq i} w_{ij} S_i S_j + \sum_i I_i S_i)). \tag{2.12}$$

The thermal average $\langle S_i \rangle$ can be stated as

$$S_i = \frac{1}{Z_{\mathrm{sg}}} \sum_S [S_i \exp(\beta(\tfrac{1}{2} \sum_{i,j \neq i} w_{ij} S_i S_j + \sum_i I_i S_i))]. \tag{2.13}$$

Knowing Z_{sg} as a function of the I_i's, $\langle S_i \rangle$ can directly be obtained by differentiation conform

$$\langle S_i \rangle = \frac{1}{\beta Z_{\mathrm{sg}}} \frac{\partial Z_{\mathrm{sg}}}{\partial I_i} = T \frac{\partial \ln Z_{\mathrm{sg}}}{\partial I_i}. \tag{2.14}$$

[4] Conversion from the one binary system to the other one, vice versa, is easy. Taking $S'_i \in \{-1, 1\}$ and $S_i \in \{0, 1\}$, it follows that $S'_i = 2S_i - 1$. The choice between the two types is a matter of mathematical convenience effecting the values of the quantities w_{ij} and I_i somewhat. Of course, this slightly modifies the physical meaning of these quantities too.

Writing $P(S_1 = s_1, S_2 = s_2, \ldots, S_n = s_n) = P(S)$ and $P(S_i = s_i) = P(S_i)$, where $s_i \in \{0, 1\}$, the free energy (2.9) becomes

$$F_{\mathrm{sg}}(P) = \sum_S P(S)(\Leftrightarrow\tfrac{1}{2} \sum_{i,j \neq i} w_{ij} S_i S_j \Leftrightarrow \sum_i I_i S_i) + \frac{1}{\beta} \sum_S P(S) \ln P(S). \quad (2.15)$$

If all values w_{ij} are positive, the system is called ferromagnetic and parallel spins are energetically favored. In thermal equilibrium, above a certain critical temperature T_{cr}, the thermal fluctuations will beat the magnetic interactions making $\forall i : \langle S_i \rangle \approx 0.5$, and the material loses nearly all its magnetization. Below T_{cr}, the magnetic interactions beat the thermal fluctuations in a certain degree making $\forall i : \langle S_i \rangle \neq 0.5$. Depending on the values of the w_{ij}'s, the I_i's, and T, the spins are found predominantly up or predominantly down. In the presence of an external magnetic field, the system will always be oriented in the direction of that field. In the absence of such a field, the system shows a time-independent 'spontaneous' magnetization [64], whose direction is not known in advance. It is also said that the ferromagnetic system exhibits a 'phase transition' at T_{cr}.

If instead, the values w_{ij} are negative – which often is the case when Hopfield networks are used to solve combinatorial optimization problems – the system is termed anti-ferromagnetic. Depending the values of the w_{ij}'s, the I_i's, and T, nearby spins now tend to become more or less anti-parallel [64], meaning that neighbouring S_i's will be found in 'opposite' states (i.e., 0 and 1). If the third possibility holds that certain w_{ij} are positive and other ones negative, the system has conflicts (also called frustration) with regard to the global orientations. The consequence is a system with several non-equivalent meta-stable global states [76].

2.1.4 Statistical dynamics and annealing

As mentioned in the beginning of this chapter, statistical mechanics especially deals with the equilibrium properties of a system. The driving mechanism by which the particles of the system – on account of their mutual interaction – are divided over the available energy levels resulting in dynamic equilibrium, is often ignored. However, applying numerical simulations (as is often done when analytic methods are inadequate), a dynamic rule has to be chosen in advance. It appeared to be possible to construct various (both deterministic and probabilistic) dynamics having the property of leading to thermal equilibrium. In the probabilistic case[5], the chosen algorithms frequently have the property that the probability of finding the system in state α_t only depends on the preceding state α_{t-1} (and not on the history prior to state α_{t-1}). These algorithms are completely described by the transition probabilities

$$P_{\mathrm{tr}}(\alpha, \alpha') = P(\alpha_t = \alpha' \mid \alpha_{t-1} = \alpha). \quad (2.16)$$

In practice, many of the transition probabilities corresponding to the selected dynamics are zero.

[5]The deterministic case we shall meet in section 2.3.

If the algorithm is 'ergodic' (meaning that any state is reachable from any other state by way of a finite number of intermediate states), and if the transition probability satisfies the so-called detailed balance condition

$$P_{\rm tr}(\alpha,\alpha')\; e^{-\beta H_\alpha} = P_{\rm tr}(\alpha',\alpha)\; e^{-\beta H_{\alpha'}}, \tag{2.17}$$

the system relaxes to equilibrium from an arbitrary starting state [64]. The condition (2.17) does not specify the transition probability uniquely. A very common choice in simulations is the Metropolis algorithm [44, 89], which applies the transition probability

$$P_{\rm tr}(\alpha,\alpha') = \begin{cases} 1 & \text{if } \Delta H < 0 \\ e^{-\beta \Delta H} & \text{otherwise,} \end{cases} \tag{2.18}$$

where $\Delta H = H_{\alpha'} \Leftrightarrow H_\alpha$. We see that a transition from state α to α' is accepted with probability 1 if the corresponding change of the Hamiltonian is negative. Depending on a probabilistic outcome, the transition may also be accepted if the corresponding energy change is positive. The underlying idea of this strategy is that the algorithm may escape from local minima. The Metropolis algorithm satisfies the detailed balance condition so that, at a fixed temperature, it leads to thermal equilibrium.

In condensed matter physics, 'annealing' is a technique for obtaining low energy states of a solid in a heat bath. The process consists of two steps:

- Increase the temperature of the heat bath to a value at which the solid melts.
- Carefully decrease the temperature of the heath bath until the particles arrange themselves in the ground state of the solid.

The physical annealing can be simulated using computer power yielding what is called simulated annealing. The most common approach is just to apply the Metropolis algorithm, where the temperature is decreased step by step. The temperature is called the control parameter. Using Markov chains, asymptotic convergence of the algorithm has been proven [2]. Furthermore, a lot of empirical performance analysis has been done in order to get practical, finite-time approximations. The algorithm has been used to solve, among other things, combinatorial optimization problems.

2.1.5 Mean field theory

Beside simulation, there exist various analytic techniques [64, 89] in order to understand statistical mechanical models like the 'power series expansions', the 'real normalization group', the 'field theoretical approach' and, the simplest one termed the 'mean field approximation'. The essential ingredient of the mean field theory is the neglect of thermal fluctuations of the individual neurons. Instead, one considers the *average* effect of these fluctuations. One starting point of mean field theory is the principle of minimal free energy. Instead of looking for the true minimum of the free energy (2.9), certain restrictions are imposed on the probability distribution.

An example consists of a mean field analysis of the spin glasses with Hamiltonian (2.10), where $(w_{ij}) = (w_{ji})$. Using the simplest approximation, the probability distribution is assumed to be factorized meaning that the spins are treated as independent described by

$$P(S) = P(S_1)P(S_2)\dots P(S_n). \tag{2.19}$$

Referring to the average magnetization $\langle S_i \rangle$ as V_i, it follows that $P(S_i = 1) = V_i$ and $P(S_i = 0) = 1 \Leftrightarrow V_i$. Using all this, we can write

$$\begin{aligned}
E(P) &= \sum_S P(S)H(S) \\
&= \sum_S P(S_1)P(S_2)\dots P(S_n)(\Leftrightarrow\tfrac{1}{2}\sum_{i,j\neq i} w_{ij}S_iS_j \Leftrightarrow \sum_i I_iS_i) \\
&= \Leftrightarrow\tfrac{1}{2}\sum_{S_iS_j} P(S_i)P(S_j)\sum_{i,j\neq i} w_{ij}S_iS_j \Leftrightarrow \sum_{S_i} P(S_i)\sum_i I_iS_i \\
&= \Leftrightarrow\tfrac{1}{2}\sum_{i,j\neq i} w_{ij}V_iV_j \Leftrightarrow \sum_i I_iV_i, \qquad (2.20) \\
\mathcal{S}(P) &= \Leftrightarrow\sum_S P(S_1)P(S_2)\dots P(S_n)(\ln P(S_1) + \ln P(S_2) + \dots + \ln P(S_n)) \\
&= \Leftrightarrow\sum_i (V_i \ln V_i + (1 \Leftrightarrow V_i)\ln(1 \Leftrightarrow V_i)). \qquad (2.21)
\end{aligned}$$

Under the conditions (2.19), the free energy of the spin glasses can thus be stated as

$$\begin{aligned}
F_{\mathrm{sg_{mf}}}(V) &= \Leftrightarrow\tfrac{1}{2}\sum_{i,j\neq i} w_{ij}V_iV_j \Leftrightarrow \sum_i I_iV_i + \\
&\qquad \tfrac{1}{\beta}\sum_i (V_i \ln V_i + (1 \Leftrightarrow V_i)\ln(1 \Leftrightarrow V_i)).
\end{aligned} \tag{2.22}$$

The necessary condition for finding a minimum of F_{mf} yields (using $w_{ij} = w_{ji}$)

$$\partial F_{\mathrm{sg_{mf}}}/\partial V_i = \Leftrightarrow\sum_{j\neq i} w_{ij}V_j \Leftrightarrow I_i + \frac{1}{\beta}\ln\frac{V_i}{1 \Leftrightarrow V_i} = 0. \tag{2.23}$$

Resolving this equation, we finally find that at thermal equilibrium

$$V_i = g_\beta(\hbar_i) = \frac{1}{1 + e^{(-\beta\hbar_i)}} \;\wedge\; \hbar_i = \sum_{j\neq i} w_{ij}V_j + I_i, \tag{2.24}$$

where $\hbar_i$ equals the effective magnetic field. The function g_β is the sigmoid or logistic function (see figure 2.1). For high values of the temperature $T = 1/\beta$, we see that $V_i \approx 0.5$, which corresponds to the outcome of the analysis of the exact free energy (2.15): the system is almost completely disordered. For low values of the temperature, the sigmoid function approximates the step or Heaviside

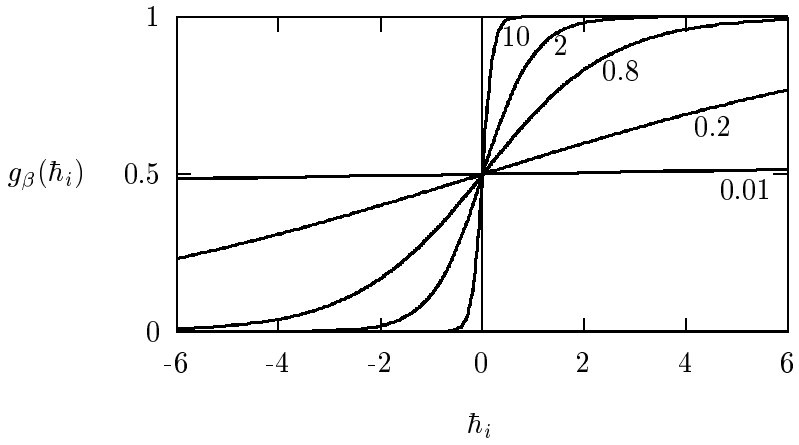


Figure 2.1: The logistic function for various values of β.

function implying that, on average, the spin i equals 0 or equals 1 depending on the value of the effective magnetic field: the system shows an ordered structure. Despite the simplicity of the expressions, the mean field approximation has a rich structure. Many physical phenomena like spontaneous magnetization, phase transitions, stability, metastability and unstability, can be described by the model [64, 89].

Comparing (2.11) and the second expression in (2.24), we see that the first, exact equation takes the spin fluctuations into account, while in the second one, the fluctuations S_i are replaced by their average value V_i. In other words, the stochastic magnetic field is replaced by an effective field as given by its mean field approximation.

2.2 Combinatorial optimization

2.2.1 Definition and complexity

Solving combinatorial optimization (i.e., either minimization or maximization) problems deals with the determination of the 'best' solution among a set of alternative solutions. In case of minimization, a combinatorial optimization problem can be defined [34] as a minimization problem consisting of a set of problem instances. For each instance, there is a finite set S_c of candidate solutions, where a cost function $f : S_c \to \mathbb{R}$ exists that assigns a real number (the solution value) to each candidate solution $c \in S_c$. An optimal solution is a candidate solution c^* such that

$$\forall c \in S_c : f(c^*) \leq f(c). \tag{2.25}$$

Candidate solutions can often be described by means of bond variables and the optimization problem as a whole is often described by a 'constrained' combinato-

rial minimization problem, stated as

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to :} & C_\alpha(x) = 0,\ \alpha = 1, \ldots, n, \end{array} \tag{2.26}$$

where $x = (x_1, x_2, \ldots, x_n)$. The $C_\alpha(x)$'s are the so-called constraints to which candidate solutions are subjected.

Over the years, it has been shown that many combinatorial optimization problems belong to the class of so-called $\mathcal{NP}$-hard problems. For several reasons [34], it is generally believed that all problems of this class are 'intractable' meaning that there exist no algorithms with running time polynomial in the input size. In practice this means that optimal solutions of 'large' instances of this type of problems cannot be obtained in 'reasonable' amounts of computation time.

2.2.2 Examples

Among all combinatorial optimization problems, the traveling salesman problem (TSP) – which has been proven to be $\mathcal{NP}$-hard – is probably the best known. A problem instance of the TSP consists of n cities and an $n \times n$-matrix (d_{pq}), whose elements denote the distance between each pair (p, q) of cities. A candidate solution is a 'tour', which is a closed path along all cities with the constraint that each city is visited exactly once. The goal is to find a tour of minimal length.

Another combinatorial optimization problem, which will also be tried, is the 'weighted matching problem' (WMP). An instance of the WMP consists of n (n being even) points again with known mutual distances (d_{pq}). A candidate solution is given by a state, where the points are linked together in pairs, with (the constraint of) each point being linked to exactly one other point. The goal is to find minimal total length of the links. Unlike for the TSP, for the solution of WMP exist fast polynomial algorithms [44].

2.2.3 Solving methods

Since many $\mathcal{NP}$-hard optimization problems are of practical interest, a lot of effort has gone into solving them one way or another. The solution strategies can be roughly divided in two categories [34]. Applying an algorithm of the one category, it is tried to obtain an improvement over the straightforward exhaustive search approach. Examples are methods based on 'branch-and-bound' or 'backtracking' consisting of a tree-structured search bounded by recognizing that some partial solutions can impossibly be extended to actual solutions or to solutions of better quality than the best one already found. Other approaches of this category apply alternative ways of organizing the search like 'divide and conquer' and 'dynamic programming' methods [52]. Applying the first of these two approaches, a problem is split into smaller ones, the smaller problems are resolved (by recursively applying the same technique), and their solutions are combined into the solution of the original problem. Applying dynamic programming on the other

hand, the solution of a problem is built stage-wise, where at each stage a new aspect of the problem is added until the solution to the original problem has been found.

The other category of algorithms apply a 'heuristic' approach, where it is attempted to find a 'good' solution within an acceptable amount of time. 'Local search' algorithms [1] are an example of this category. These algorithms take some solution and search over a set of neighbouring solutions, in this way trying to find solutions of lower cost.

Within the two classes, it is possible to distinguish between 'tailored' and 'general' algorithms [2]. Tailored algorithms use problem-specific information (domain knowledge) and their applicability is therefore often very limited. Instead, general algorithms are appropriate to a wider variety of problems and it is of high importance to discover general methods which – as a rule – perform well. In the last decade, several new general search algorithms have emerged, all inspired by optimization principles observed in nature. They are simulated annealing, 'genetic', and neural network algorithms. Solving combinatorial optimization problems using simulated annealing (section 2.1.4) is based on the assumptions [2] that (a) solutions in the optimization problem are equivalent to states of a physical system, and (b) the cost of a solution is equivalent to the energy of a state.

Genetic algorithms [36] try to solve problems based on the principles of natural evolution. The algorithm keeps up a population of candidate solutions. New generations of candidate solutions are successively created applying 'selection', 'mutation' and 'crossover' operations, where the 'fittest' solutions have the highest probability of being selected. It is hoped that the fitness of the population members gradually improves and, finally, a member among them is found with optimal fitness. The assumptions for applying genetic algorithms to combinatorial optimization problems are that (a) candidate solutions of the optimization problem can be represented as population members (and therefore can be selected, mutated, and crossed over), and (b) the cost of a solution is equivalent to the fitness of the corresponding population member.

This last class of algorithms refers to neural networks, whose relevant types are introduced now.

2.3 Classical Hopfield models

2.3.1 The asynchronous model

In 1982, Hopfield[6] showed how useful, computational properties can emerge as collective properties of neural systems [46]. The collective properties of his neural network produce a content-addressable memory. Each neuron S_i has two (output) states: $S_i = 0$ or $S_i = 1$, the other neural quantities are equivalent to those defined in section 1.1.4. The iterative algorithm for the time evolution of the sys-

[6] To be somewhat more exact historically, Hopfield's binary model is a stochastic reinterpretation of an earlier model by Amari (1977). The difference lies in the way the neurons are updated: in Amari's model this is done synchronously, in Hopfield's model this is assumed to occur asynchronously [54].

tem state[7] $S = (S_1, \ldots, S_n) \in \{0,1\}^n$ can be formulated as (compare the equations (1.2) and (1.3))

$$S_i^{\text{new}} = \begin{cases} 1 & \text{if } U_i^{\text{old}} = \sum_j w_{ij} S_j^{\text{old}} + I_i \geq \mu_i \\ 0 & \text{otherwise,} \end{cases} \tag{2.27}$$

where μ_i represents the threshold value of neuron i and where each neuron readjusts its state randomly in time but with equal mean rate. The importance of Hopfield's approach stems from his proof on stability considering the energy function

$$E_{\text{a}}(S) = \Leftrightarrow\tfrac{1}{2} \sum_{i,j} w_{ij} S_i S_j \Leftrightarrow \sum_i I_i S_i + \sum_i \mu_i S_i. \tag{2.28}$$

Theorem 2.1 (Hopfield). *If (w_{ij}) is a symmetric matrix and $\forall i : w_{ii} \geq 0$, then the energy function* (2.28) *is a Lyapunov*[8] *function for motion equations* (2.27)*.*

If all threshold values μ_i equal zero, then $E_{\text{a}}(S)$ nearly coincides with equation (2.10). In addition, U_i corresponds to the local magnetic field h_i in (2.11).

The asynchronous character makes the flow of the system not entirely deterministic, but in any case, the algorithm leads to a final attractor (like a memory) near the starting state. Stated in other words, the algorithm ends up in a *local* minimum. It explains the suitability of this neural network to model an associative memory.

2.3.2 The continuous model

In 1984, Hopfield generalized the asynchronous model to a deterministic one [47] using continuous-valued neurons with input values $U_i \in \mathbb{R}$ and output values $V_i \in [0,1]$. Instead of using an iterative updating rule like

$$V_i^{\text{new}} = g(U_i^{\text{old}}) = g(\sum_j w_{ij} V_j^{\text{old}} + I_i), \tag{2.29}$$

Hopfield introduced the updating rule (motion equation)

$$c_i \dot{U}_i = \Leftrightarrow \frac{\partial E_{\text{c}}(V)}{\partial V_i} = \sum_j w_{ij} V_j + I_i \Leftrightarrow U_i, \tag{2.30}$$

where *continuously* $V_i = g(U_i)$ holds and where c_i represents a suitable time constant. In our simulations, we shall approximate the time derivative of U_i by writing

$$\dot{U}_i \approx \frac{\Delta U_i}{\Delta t} \tag{2.31}$$

[7] Another, wider view on the notion of a 'system state' will be discussed in section 3.3.1.

[8] The notions of 'stability' and 'Lyapunov function' come from the theory on 'dynamic systems': see appendix B.

and then choose an appropriate value of Δt. If we confine ourselves to equal values for all c_i, no further restrictions are introduced if we simply take $\forall i : c_i = 1$. So, this will be done. The updating rule (2.30) can be derived using the circuit equations of an analogue electrical circuit implementing the continuous Hopfield model: it represents a resistance-capacitance charging equation that determines the rate of change of U_i. Mathematically, as denoted in equation (2.30), it can be derived applying the technique of gradient descent (appendix C) to the energy function $E_c(V)$ which is defined conform

$$E_{\mathrm{c}}(V) = \underbrace{\Leftrightarrow\tfrac{1}{2}\sum_{i,j} w_{ij}V_iV_j \Leftrightarrow\sum_i I_iV_i}_{E(V)} + \underbrace{\sum_i \int_0^{V_i} g^{-1}(v)dv}_{E_{\mathrm{h}}(V)} \quad (2.32)$$

$$= E(V) + E_{\mathrm{h}}(V) \quad . \quad (2.33)$$

Here, $E(V)$ is the energy function to be minimized. The second term, $E_{\mathrm{h}}(V)$, will be called the 'Hopfield term'. $V \in [0,1]^n$ is the state vector $(V_1, \ldots, V_n)$ of the continuous neural net. We further note that $U_i = \partial E_{\mathrm{h}}/\partial V_i$. In figure 2.2, a picture

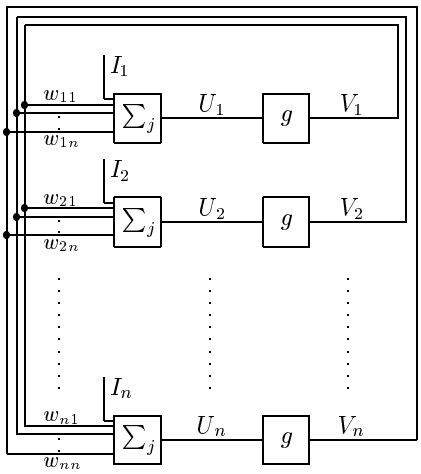


Figure 2.2: The continuous Hopfield network with equilibrium condition: $\forall i : U_i = \sum_j w_{ij}V_j + I_i$ and $V_i = g(U_i)$.

of the continuous Hopfield model is given. It can be used to explain the working of the motion equations (2.30). After initialization, the network is generally not in an equilibrium state. Then, while keeping the relations $V_i = g(U_i)$ valid, the input values U_i are adapted in agreement with (2.30). The following theorem, proven by Hopfield [47], gives conditions for which an equilibrium state will eventually be reached:

Theorem 2.2 (Hopfield). *If (w_{ij}) is a symmetric matrix and if $\forall i : V_i = g(U_i)$ is a monotone increasing, differentiable function, then E_{c} is a Lyapunov function for motion equations* (2.30).

Under the given conditions, the theorem guarantees convergence to an equilibrium state of the neural net where

$$\forall i : V_i = g(U_i) \ \wedge \ U_i = \sum_j w_{ij} V_j + I_i. \tag{2.34}$$

If the sigmoid function is chosen as the transfer function g, we see that expression (2.34) almost coincides with (2.24).

In his article, Hopfield dwells on the relation between the energy E_{a} of the asynchronous model and E_{c} of the continuous one. In order to understand the relationship, he introduces a scaling factor β (in the original paper denoted by λ) replacing $V_i = g(U_i)$ by $V_i = g(\beta U_i)$. He then argues that, in the high-gain limit $\beta \to \infty$, the Hopfield term E_{h} becomes negligible, making the locations of the extrema of E_{c} and E_{a} almost equal. Next, he remarks that for large but finite β-values, the Hopfield term begins to contribute, leading to an energy surface whose maxima are still at corners of the hypercube $[0,1]^n$, but whose minima are slightly displaced toward the interior. We will return to these aspects in section 3.2.2.

The conditions given in theorem 2.2 do not uniquely specify the transfer function g of a neuron. Commonly used functions include the tanh for the $[\Leftrightarrow 1, +1]$ range (used, e.g. in [68], in the iterative way given by equation (2.29)) and the sigmoid function (2.24) for the $[0,1]$ range. We shall meet other activation functions later on.

2.3.3 The stochastic model

It is possible to make the neurons of the binary asynchronous network behave stochastically[9] [44] applying a stochastic evolution rule like the transition probability (2.18) of the Metropolis algorithm. Instead, in the context of neural networks, another form is usually chosen that is suitable for parallel computation [3]: regardless of the previous state, the probability of setting $S_i = 1$ is taken

$$P(S_i = 1) = \frac{1}{1 + e^{-\beta U_i}}. \tag{2.35}$$

The units are selected in the same asynchronous way mentioned in section 2.3.1. It is not difficult to check [44] that the updating rule (2.35) leads to a transition probability which satisfies the detailed balanced condition (2.17). So, the stochastic Hopfield model is expected to reach thermal equilibrium conform the Boltzmann probability distribution. It is therefore sometimes called a Boltzmann machine with *a priori* chosen weights [44]. As the Metropolis algorithm makes it possible to escape from local minima, so the stochastic rule (2.35) does. This observation has suggested the idea trying to use stochastic Hopfield networks in order to find *global* minima of optimization problems. Besides, annealing can be applied by decreasing the temperature gradually during execution yielding a new form of simulated annealing.

[9] Still another possibility is to make continuous neurons behave stochastically [37].

The asynchronous Hopfield model can be considered a special case of the stochastic one: at very low temperatures, the noise level (i.e., the level of the thermal fluctuations) is negligible reducing the stochastic model to the asynchronous Hopfield model. This can be understood mathematically by observing that for $\beta \to \infty$ the sigmoid function (2.35) reduces to the step function as defined in (2.27). The continuous and the stochastic Hopfield network are also related. Because the energy expressions (2.10) of the Ising model and (2.28) of the stochastic Hopfield model almost coincide, a mean field analysis of the last one can be done precisely conform the analysis of section 2.1.5, yielding the equilibrium equations (2.34). This proves the following theorem:

Theorem 2.3. *The equilibrium states of the mean field approximation of the binary stochastic Hopfield model coincide with the equilibrium states of the continuous Hopfield model, if, in the last model, the sigmoid function is chosen as the transfer function of the neurons.*

In the literature [68, 44], several other proofs can be found which usually adopt the 'saddle point approximation' (see section 3.1). The theorem makes clear that the binary stochastic neural network can be approximated by the continuous one, or, stated more precisely [68]:

> "The hill-climbing property of the stochastic model at non-zero temperature can be cast into a deterministic procedure in a smoother energy landscape."

Consequently, if the networks are simulated on a sequential computer device, the problem of excessive computer time of the stochastic model is hoped to be circumvented applying the approximating, continuous model: the *deterministic* relaxation rule (2.30) is expected to converge much faster than the stochastic rule (2.35) [10]. If, in addition, annealing is applied, the simulated annealing approach of the stochastic model reduces to, what has been termed 'mean field annealing' [44, 50, 68, 69, 70]. Then, on lowering the temperature, fine details of the original cost function $E(V)$ gradually appear [77].

2.4 Hopfield networks and optimization

This section is meant to give a concise background on the application of Hopfield networks in the field of combinatorial optimization. We shall return to many aspects later on.

As for simulated annealing and genetic algorithms assumptions have been formulated to solve combinatorial optimization problems in a heuristic way (section 2.2.3), so this can be done for Hopfield neural networks. Here, the assumptions are that (a) candidate solutions of the optimization problem are equivalent to network states, and (b) the cost of a solution is equivalent to the energy value of the

[10] Alternatively, the deterministic iterative rule (2.29) can be chosen: experiments of this type have shown significant speedup factors, together with comparable and sometimes even better quality of solutions [68, 70, 43].

network. Since 1985 [48], researchers have tried to use both stochastic and continuous Hopfield networks in the field of combinatorial optimization. The general problem can be stated like (2.26), where the cost function $f(x)$ should be replaced by an energy function $E(x)$:

$$\begin{aligned} &\text{minimize} && E(x) \\ &\text{subject to :} && C_\alpha(x) = 0,\ \alpha = 1, \ldots, m, \end{aligned} \tag{2.36}$$

x being the state vector (S or V) of the neural net. There exist different ways in treating the constraints. The oldest approach consists of a so-called *penalty method*, sometimes called the 'soft' approach [77, 69]: extra 'penalty' terms are added to the original energy function, penalizing violation of constraints. We nowhere found a precise definition of the penalty method. Having collected many examples, we think the following characterization reflects the issue at stake: the penalty terms are weighted and chosen in such a way that

$$\begin{aligned} &\sum_{\alpha=1}^{m} c_\alpha C_\alpha(x) \text{ has a minimum value zero} \Leftrightarrow \\ &\qquad\qquad x \text{ represents a valid solution.} \end{aligned} \tag{2.37}$$

A valid (admissible, or feasible) solution is defined as a candidate solution which complies with all submitted constraints. Usually, the chosen penalty terms are *quadratic* expressions. Applying a continuous Hopfield network, the original problem (2.36) is converted into

$$\text{minimize } E_\text{p}(V) = E(V) + \sum_{\alpha=1}^{m} c_\alpha C_\alpha(V) + E_\text{h}(V), \tag{2.38}$$

$E(V)$ and $E_\text{h}(V)$ being given by (2.33). The corresponding updating rule is:

$$\dot{U}_i = \Leftrightarrow \frac{\partial E_\text{p}}{\partial V_i} = \Leftrightarrow \frac{\partial E}{\partial V_i} \Leftrightarrow \sum_\alpha c_\alpha \frac{\partial C_\alpha}{\partial V_i} \Leftrightarrow U_i, \tag{2.39}$$

where, in case of $(w_{ij}) = (w_{ji})$, $\Leftrightarrow \partial E/\partial V_i = \sum_j w_{ij} V_j + I_i$. We already know from Hopfield's analysis (section 2.3.2), that the influence of the Hopfield term $E_\text{h}(V)$ may be small. Ignoring this term for the moment, the energy function E_p is a *weighted* sum of $m+1$ terms and hence a difficulty arises in determining correct weights c_α. The minimum of E_p is a *compromise* between fulfilling the constraints and minimizing the original cost function $E(V)$. Applying this penalty approach to the TSP [20, 48, 49, 88], the weights had to be determined by trial and error. For only a small low-dimensional region of the parameter space valid tours were found, especially when larger problem instances were tried[11].

In a second approach, the features of the neural net are changed. The alteration is usually done in such a way, that some or all constraints are *automatically*

[11] Aside we mention that for 'purely combinatorial problems' (by which we mean combinatorial problems without a cost function to be minimized like the n-queen problem and the 4-coloring problem), the penalty method has proven to be useful [82]. See also section 2.7.

fulfilled. This way of dealing with the constraints is sometimes called the 'strong' one [77]. As an example, observe the following constraint

$$\sum_{i}^{n} S_i \Leftrightarrow 1 = 0. \tag{2.40}$$

A consequence of (2.40) is that precisely one of the binary S_i's equals one, all the other ones being 0. Physically, this model is related to Potts glasses. Trying to solve the TSP, condition (2.40) can be used several times in order to guarantee that all cities are visited once. The other condition – that two cities are never visited at the same time – can be fulfilled in the soft way using penalty terms. A mean field annealing approach using an iterative updating role of the form

$$V_i^{\text{new}} = \frac{\exp(\Leftrightarrow\!\!\!\!\beta U_i^{\text{old}})}{\sum_k \exp(\Leftrightarrow\!\!\!\!\beta U_k^{\text{old}})}, \tag{2.41}$$

has shown 'encouraging' results: experiments for problem sizes up to 200 cities yielded solutions with a quality comparable to and sometimes even better than the simulated annealing one. Besides, stability analyses including an estimation of the critical temperature (at which a phase transition takes place corresponding to a rapid drop of the energy in the system) have been reported [85, 69, 70, 78]. Another way of implementing the constraint (2.40) is to use so-called 'maximum neurons' defined by

$$S_i = \begin{cases} 1 & \text{if } U_i = \max\{U_1, \ldots, U_n\} \\ 0 & \text{otherwise.} \end{cases} \tag{2.42}$$

They have been applied for, among other things, finding near-optimum solutions of 'channel routing problems' [82]. Another way of changing the features of the neural net has been the introduction of an extra layer. In an attempt to solve the TSP [51], a first layer was chosen conform a continuous Hopfield network where the penalty term is based on city adjacency in the tour. The second layer of the network had to detect, in parallel, closed sub-tours of intermediate solutions. Unfortunately, the implementation of the second layer is more complicated than was indicated.

A third way of treating the constraints was introduced in 1988 [71]. Here, the starting point is the multiplier method of Lagrange (appendix A), where a constrained optimization problem is converted into an unconstrained *extremization* one: a solution of the general problem (2.36) is also a *critical* point of

$$E_{\text{pb}}(V, \lambda) = E(V) + \sum_{\alpha=1}^{m} \lambda_\alpha C_\alpha(V), \tag{2.43}$$

where λ is a vector of multipliers $(\lambda_1, \cdots, \lambda_m)$. Contrary to the requirement (2.37) used in the penalty approach, the constraints should now be formulated such that

$$\forall \alpha : C_\alpha(x) = 0 \Leftrightarrow x \text{ represents a feasible solution.} \tag{2.44}$$

Moreover, the multiplier values are not supplied by the user, but, after having been initialized, are determined by the system itself: conform the so-called basic differential multiplier method (BDMM), the values of the Lagrange multipliers can be estimated applying a gradient *ascent*[12]. The complete system of motion equations of the model equals

$$\dot{V}_i = \Leftrightarrow \frac{\partial E_{\mathrm{pb}}}{\partial V_i} = \Leftrightarrow \frac{\partial E}{\partial V_i} \Leftrightarrow \sum_{\alpha} \lambda_\alpha \frac{\partial C_\alpha}{\partial V_i}, \tag{2.45}$$

$$\dot{\lambda}_\alpha = + \frac{\partial E_{\mathrm{pb}}}{\partial \lambda_\alpha} = C_\alpha(V). \tag{2.46}$$

Stability can be analyzed by combining both of these differential equations into one second-order differential equation, which describes a damped harmonic motion. The total energy of the mass system consists of the sum of kinetic and potential energy given by

$$E_{\mathrm{kin+pot}} = \sum_i \tfrac{1}{2} \dot{V}_i^2 + \sum_\alpha \tfrac{1}{2} C_\alpha^2(V). \tag{2.47}$$

Theorem 2.4 (Platt & Barr). *If the damping matrix* (a_{ij}) *defined by*

$$a_{ij} = \frac{\partial^2 E}{\partial V_i \partial V_j} + \sum_\alpha \lambda_\alpha \frac{\partial^2 C_\alpha}{\partial V_i \partial V_j} \tag{2.48}$$

is positive definite, then the energy (2.47) *is a Lyapunov function for motion equations* (2.45) *and* (2.46)*.*

Using the definition of $E(V)$ as given by (2.33), it is clear that if (w_{ij}) is symmetric then

$$\frac{\partial^2 E}{\partial V_i \partial V_j} = \Leftrightarrow w_{ij}. \tag{2.49}$$

We further note that in formula (2.45) the gradient descent on E_{pb} is equated to the time derivative of V_i and not of U_i, as is done in the continuous Hopfield model. Moreover, the term $\Leftrightarrow U_i$ is lacking and, corresponding to this, the Hopfield term $E_{\mathrm{h}}(V)$ in (2.43) is missing. The necessary steps to bring these things into line with one another were made in 1989 and are explained in the next section.

2.5 The Hopfield-Lagrange model

By adding the Hopfield term $E_{\mathrm{h}}(V)$ to the energy $E_{\mathrm{pb}}(V, \lambda)$, the continuous Hopfield model and the Lagrange multiplier method were integrated [86] in what we shall call the *Hopfield-Lagrange* model. The model was used to solve the Multiple TSP (MTSP). The MTSP is an extension of the TSP, where a set of minimal closed

[12] The background of this sleight will be illuminated extensively in chapter 5.

routes should be found for a given number of salesmen. The constraints are similar to those of the original TSP. In general terms, the energy of the model is given by

$$E_{\mathrm{hl}}(V,\lambda) = E(V) + \sum_{\alpha} \lambda_\alpha C_\alpha(V) + E_{\mathrm{h}}(V) \tag{2.50}$$

$$= \Leftrightarrow\tfrac{1}{2} \sum_{i,j} w_{ij} V_i V_j \Leftrightarrow \sum_i I_i V_i + \sum_\alpha \lambda_\alpha C_\alpha(V) + E_{\mathrm{h}}(V) \tag{2.51}$$

with the corresponding set of differential equations

$$\dot{U}_i = \Leftrightarrow \frac{\partial E_{\mathrm{hl}}}{\partial V_i} = \sum_j w_{ij} V_j + I_i \Leftrightarrow \sum_\alpha \lambda_\alpha \frac{\partial C_\alpha}{\partial V_i} \Leftrightarrow U_i, \tag{2.52}$$

$$\dot{\lambda}_\alpha = + \frac{\partial E_{\mathrm{hl}}}{\partial \lambda_\alpha} = C_\alpha(V). \tag{2.53}$$

In the literature, little attention has been paid to this model. We did not find an analysis of the stability of the differential equations (2.52) and (2.53) anywhere. In case of the Multiple TSP, six coupled differential equations had to be resolved, whose stability properties were 'in the process of investigating' based on the Lyapunov function (2.47). By numerical simulation using a first order Euler method, good solutions have been found for certain small problem instances up to 20 cities and 4 salesmen.

2.6 Elastic nets

The 'elastic net' [28] deals with a specific type of neural network, namely one for solving the TSP. The elastic net algorithm (ENA) was derived from a hypothetical 'tea trade model' [59] for the establishment of topographically ordered, neighbor-preserving projections[13]. The energy function to be minimized of the elastic net equals:

$$E_{\mathrm{en}}(x) = \tfrac{\alpha_2}{2} \sum_{i=1}^{m} \mid x^{i+1} \Leftrightarrow x^i \mid^2 \Leftrightarrow \tfrac{\alpha_1}{\beta} \sum_{p=1}^{n} \ln \sum_{j=1}^{m} \exp(\tfrac{-\beta^2}{2} \mid x_p \Leftrightarrow x^j \mid^2). \tag{2.54}$$

Here, x^i represents the i-th elastic net point or 'bead' and x_p represents the location of city p. The succeeding m elastic net points form a sort of elastic rubber ring, that should be dragged along all n cities. The first term of E_{en} equals the sum of distance squares between succeeding net points (which, in a sufficient degree, corresponds to the tour length), while the second term enforces a match between each city and one of the elastic net points. Application of gradient descent to equation (2.54) yields, after a discretization step, the updating rule

$$\Delta x^i = \tfrac{\alpha_2}{\beta}(x^{i+1} \Leftrightarrow 2x^i + x^{i-1}) + \alpha_1 \sum_p \Lambda^p(i)(x_p \Leftrightarrow x^i), \tag{2.55}$$

[13] Making topology preserving maps is part of the 'unsupervised learning' approach of neural networks [44].

where the time-step $\Delta t = 1/\beta$ equals the current temperature T and where $\Lambda^p(i)$ is defined conform

$$\Lambda^p(i) = \frac{\exp(\Leftrightarrow\frac{\beta^2}{2} \mid x_p \Leftrightarrow x^i \mid^2)}{\sum_l \exp(\Leftrightarrow\frac{\beta^2}{2} \mid x_p \Leftrightarrow x^l \mid^2)}. \tag{2.56}$$

The ENA has an important scaling property: the number of variables (i.e., the two-dimensional net points) required is linear relative to the number of cities, while in case of the Hopfield model the number of neurons needed is usually quadratic relative to that number.

In practice, all x_p are normalized to points in the unit square. The elastic network is usually initialized in a small ring in the middle of that square. Taking $m = 2.5n$, the following parameter values appear to be efficient [28]: $\alpha_1 = 2.0$ and $\alpha_2 = 0.2$. The initial value of the temperature $T = 1/\beta$ is set to 0.2, and is reduced by 1% every n iterations to a final value in the range 0.01-0.02. The general effect of this lowering is that large-scale, global adjustments occur early on, resulting in a general stretching out of the elastic net. This initial stretching out is strongest to regions in the unit square having the highest concentration of cities. Later on, smaller refinements occur corresponding to an increasingly local adaptation of the elastic net towards city points. Eventually, every city must be 'visited' by one bead. In [28, 44], a picture can be found of the gradual stretching out of the elastic net. Up to several hundred cities, the ENA yields sub-optimal solutions where the final tour-lengths exceed the optimal lengths by approximately 6% [78]. The results strongly depend on the chosen parameters and the algorithm may end up in a non-valid state.

In 1990, two papers have been published on the relationship between elastic and Hopfield neural nets. One paper [77] suggested statistical mechanics as the common underlying framework, to which (in our view incorrect) analysis we shall return extensively in chapter 6. There, we shall also take stock of the other proposed common framework [90], namely that of 'generalized deformable templates'. The ENA has also been modified in many ways in other to improve the performance quality with respect to both the shortest tour found and the percentage of valid solutions encountered: see, e.g., [78, 31, 5, 22].

2.7 Computational results from the literature

Besides the afore-mentioned applications, many other achievements have been gained in the field of (combinatorial) optimization and of association using Hopfield type neural networks. We here give a notable but not exhaustive list of examples.

- The book of Takefuji [82] contains several practical problems which have been tackled and resolved quite successfully using Hopfield networks of various types, e.g., networks with alternative transfer functions. Besides solutions of the n-queen and the k-colourability problem, near-optimal solutions of 'graph planarization' and 'channel

routing' problems (both important topics in designing printed circuit boards) are presented. Furthermore, 'RNA secondary structure prediction', 'tiling', 'sorting and searching', 'broadcast scheduling', and various other problems are discussed including their computational results.

- Neural computational results of the TSP and the WMP (section 2.2.2) as well as solutions to the 'graph bipartioning' and to the 'reconstruction of an image' (from noisy or blurred data) can be found in the textbook [44], with a lot of references belonging to them. In fact, in the proceedings of any large recent international conference on neural networks, one often encounters an article containing a new attempt to tackle the TSP. A recently encountered example is [24], which applies so-called 'higher order' neural networks and which appears to be quite related to the analysis as given in chapter 4 of this thesis (see the discussion at the end of section 4.3.3).
- Similarly, higher order neural networks were applied in the context of process scheduling in flexible manufacturing systems [80]. Other examples of scheduling problems resolved by using Hopfield neural networks, can be found in [70, 81]. The first of these references describes, among other things, neural solutions to the determination of a timetable for teachers and classes in a high school, the second discusses a neural solution to an assortment problem as found in the iron and steel industry.
- In [55], two applications of Hopfield neural networks in the field of vision are given, the first one on 'texture segmentation' of images (where the segmentation problem is formulated as an optimization problem), the second one on 'image restauration' (from a recording which is degraded in one way or another). Comparisons to other methods are given. Image restoration by Hopfield networks has become a popular area of research as it is, see e.g. , the proceedings of ICNN'95.
- Between other neural network applications in the area of high-speed communication networks (where the 'asynchronous transfer mode' technology is the standard), 'optimized routing' and 'optimal packet scheduling in input queues' by means of recurrent neural networks are discussed in [65], including their hardware implementations.

Chapter 3

Unconstrained Hopfield networks

In this chapter, we start trying to attain the first object of study as mentioned in section 1.2.2. Most part of it is devoted to the study of the continuous Hopfield model as introduced in section 2.3.2. We start offering an alternative derivation of theorem 2.3 (on the mean field approximation of the stochastic model) yielding some old and several new approximations of the free energy of the system. Next, we analyze the properties of these approximations and their relation to the corresponding continuous model. In a third section, we generalize this continuous model in two steps, eventually culminating in a very general framework. Finally, we report the results of simulations that were set up in order to test some of the theoretical conjectures.

Some parts of this chapter have been published earlier in [9, 14, 15, 17] or will be published soon [11]. A large part has also been recorded in the technical reports [13, 16].

3.1 The mean field approximation revisited

A mean field analysis of binary stochastic Hopfield networks was described in chapter 2. Here, we shall deal with an alternative mean field analysis yielding various approximations of the true free energy. These expressions will turn out to be very useful later on. To reach our goal, we adopt (a slightly modified version of) an approach given by Simic [77]. One difference between his and our approach concerns the way the external fields I_i are treated: he includes *small* 'generating fields' [76] in the expression of the partition function (2.2), which are set to zero during the derivation. We use *real* external fields I_i, conform equation (2.28), which remain in the formulas. Unlike Simic, we start analyzing the original (*un*constrained) binary Hopfield model.

Theorem 3.1. *If (w_{ij}) is a symmetric matrix, then a mean field approximation of the free energy of stochastic binary Hopfield networks can be stated as*

$$F_{\mathrm{u1}}(V) = \tfrac{1}{2}\sum_{i,j} w_{ij}V_iV_j \Leftrightarrow \tfrac{1}{\beta}\sum_i \ln\left[1+\exp\left(\beta(\sum_j w_{ij}V_j + I_i)\right)\right], \tag{3.1}$$

where the stationary points of F_{u1} are found at points of the state space for which

$$\forall i : V_i = \frac{1}{1+e^{-\beta(\sum_j w_{ij}V_j + I_i)}}. \tag{3.2}$$

Proof. The proof applies certain lemmas, whose precise formulation and demonstration can be found in the appendix D. As usual, the starting point of the statistical mechanical analysis is the partition function (2.2), where, in this case, the Hamiltonian equals the energy of the binary Hopfield model as defined in (2.28). Thus, we have

$$Z_{\mathrm{hu}} = \sum_S \exp(\beta(\tfrac{1}{2}\sum_{i,j} w_{ij}S_iS_j + \sum_i I_iS_i)). \tag{3.3}$$

To be able to perform the summation in the partition function, the exponentials in the quadratic terms S_iS_j are turned into exponentials that are linear in the S_i's by using lemma 1 with the plus sign[1]. This yields

$$Z_{\mathrm{hu}} = \sum_S \frac{\int \exp\left[\Leftrightarrow\frac{\beta}{2}\sum_{i,j}\phi_i w_{ij}^{-1}\phi_j + \beta\sum_i S_i(\phi_i + I_i)\right]\prod_i \mathrm{d}\phi_i}{\int \exp\left[\Leftrightarrow\frac{\beta}{2}\sum_{i,j}\phi_i w_{ij}^{-1}\phi_j\right]\prod_i \mathrm{d}\phi_i}, \tag{3.4}$$

where the w_{ij}^{-1}'s represent the elements of the matrix inverse of (w_{ij}) and where the domain of integration of both (improper) integrals equals $\mathbb{R}^n$. In analyses of this kind, the integrals are often expanded around the point which maximizes the integrand. The point is called the *saddle point* [44]. We shall apply this saddle point approach in two ways. First, we calculate the saddle point for every state S and then perform the summation over all states yielding the average $\langle\hat{\phi}\rangle$ of saddle points. This calculation can be done exactly. Second, we change the order of these actions by starting with the summation and, after that, calculating the (one and only) saddle point $\tilde{\phi}$ of the summed quotients of integrals. This time, for mathematical complications, a first-order approximation is applied.

By expanding, for every state, the integrand in the numerator and the integrand in the denumerator of (3.4) around their respective saddle points, it is possible to recover formula (3.3). This follows in a straightforward way by the application of lemma 2 (see also the note after the proof of that lemma). The saddle point equation of the numerator of (3.4) leads to the formula

$$\hat{\phi}_i = \sum_j w_{ij}S_j \ \mathrm{implying\ that}\ \langle\hat{\phi}_i\rangle = \sum_j w_{ij}\langle S_j\rangle = \sum_j w_{ij}V_j, \tag{3.5}$$

[1] In an aside, we note that the condition of symmetry of the matrix (w_{ij}) of lemma 1 coincides with one of the conditions for theorem 2.1.

where $\langle\hat{\phi}_i\rangle$ is the i-th component of the average of the saddle point values of (3.4)[2].

Now we change the order. Summation over all 2^n states S in (3.4) yields, using lemma 3,

$$Z_{\rm hu} = \frac{\int \exp\left[\Leftrightarrow\frac{\beta}{2}\sum_{ij}\phi_i w_{ij}^{-1}\phi_j + \sum_i \ln\left(1+\exp(\beta(\phi_i+I_i))\right)\right]\prod_i {\rm d}\phi_i}{\int \exp\left[\Leftrightarrow\frac{\beta}{2}\sum_{ij}\phi_i w_{ij}^{-1}\phi_j\right]\prod_i {\rm d}\phi_i}. \tag{3.6}$$

Writing

$$E_{\rm hu}(\phi, I) = \tfrac{1}{2}\sum_{ij}\phi_i w_{ij}^{-1}\phi_j \Leftrightarrow \tfrac{1}{\beta}\sum_i \ln\left[1+\exp\left(\beta(\phi_i+I_i)\right)\right], \tag{3.7}$$

the saddle point $\tilde{\phi}$ of the numerator in (3.6) is found by partial differentiation of $E_{\rm hu}(\phi, I)$ to the ϕ_i's, giving

$$\tilde{\phi}_i = \sum_j \frac{w_{ij}}{1+e^{-\beta(\tilde{\phi}_j+I_j)}}. \tag{3.8}$$

Up till now, the calculations have been exact. The question arises how $\langle\hat{\phi}\rangle$ and $\tilde{\phi}$ are related. Here, the (first order) saddle point approximation as applied in lemma 4 turns out useful. Using this lemma, we find

$$V_i \approx \Leftrightarrow\frac{\partial E_{\rm hu}(\tilde{\phi}, I)}{\partial I_i} = \frac{1}{1+e^{-\beta(\tilde{\phi}_i+I_i)}}. \tag{3.9}$$

If we now substitute the approximation (3.9) in the exact formula (3.5), we obtain

$$\langle\hat{\phi}_i\rangle \approx \sum_j \frac{w_{ij}}{1+e^{-\beta(\tilde{\phi}_j+I_j)}}. \tag{3.10}$$

Comparing (3.8) and (3.10), we conclude that

$$\langle\hat{\phi}\rangle \approx \tilde{\phi}. \tag{3.11}$$

In the saddle point approximation of lemma 4, the partition function (3.6) has been approximated according to

$$Z_{\rm hu} \approx \exp(\Leftrightarrow\beta E_{\rm hu}(\tilde{\phi}, I)). \tag{3.12}$$

Using this, we can derive a saddle point approximation of the free energy of the binary stochastic Hopfield model. The derivation goes like

$$F_{\rm hu} = \Leftrightarrow\frac{1}{\beta}\ln Z_{\rm hu} \approx E_{\rm hu}(\tilde{\phi}, I) \approx E_{\rm hu}(\langle\hat{\phi}\rangle, I) = F_{\rm u1}(V), \tag{3.13}$$

[2]Apparently, $\langle\hat{\phi}_i\rangle$ represents the average *internal* input of neuron i.

where the last equality is found by substitution of (3.5). The stationary points of $F_{\mathrm{u}1}$ are found by resolving the equations $\partial F_{\mathrm{u}1}/\partial V_i = 0$. Again using the symmetry of (w_{ij}), we precisely obtain (3.2) via

$$\begin{aligned}\frac{\partial F_{\mathrm{u}1}}{\partial V_i} &= \sum_j w_{ij}V_j \Leftrightarrow \tfrac{1}{\beta}\sum_k \frac{\beta\exp(\beta(\sum_j w_{kj}V_j + I_k))w_{ki}}{1+\exp(\beta(\sum_j w_{kj}V_j + I_k))} \\ &= \sum_k w_{ik}\,(V_k \Leftrightarrow \frac{1}{1+\exp(\Leftrightarrow\beta(\sum_j w_{kj}V_j + I_k))}) = 0. \end{aligned} \tag{3.14}$$

In fact, the equations (3.2) are mean field equations (see theorem 2.3 and equations (2.24)). Apparently, the first order saddle point approximation and the mean field approximation such as derived in section 2.1.5 are similar approaches yielding the same results[3]. This observation completes the proof. □

We may realize in another way that the first order saddle point and the mean field approximation are approaches of the same kind. By combining (3.9), (3.11), and (3.5), the saddle point approximation results in the mean field equations by recognizing that

$$V_i \approx \frac{1}{1+e^{-\beta(\hat{\phi}_i + I_i)}} \approx \frac{1}{1+e^{-\beta(\langle\hat{\phi}_i\rangle + I_i)}} = \frac{1}{1+e^{-\beta(\sum_j w_{ij}V_j + I_i)}}. \tag{3.15}$$

Besides, we note that in the final result (3.1), the free energy $F_{\mathrm{u}1}$ is written as a function over V, that is, (just like in (2.9)) over an arbitrary probability distribution[4]. Comparing the original Hamiltonian (2.28) and the free energy approximation (3.1), it is remarkable that a *sign flip* in the quadratic expression of the S_i's has occurred. Even more curious is the observation, that the sign flip can be undone producing the mean field free energy expression (2.22):

Theorem 3.2. *If (w_{ij}) is a symmetric matrix, then a mean field approximation of the free energy of stochastic binary Hopfield networks can also be stated as*

$$F_{\mathrm{u}2}(V) = \Leftrightarrow\tfrac{1}{2}\sum_{i,j} w_{ij}V_iV_j \Leftrightarrow \sum_i I_iV_i + \tfrac{1}{\beta}\sum_i \big(V_i\ln V_i + (1\Leftrightarrow V_i)\ln(1\Leftrightarrow V_i)\big), \tag{3.16}$$

where the stationary points of $F_{\mathrm{u}2}$ coincide with those of $F_{\mathrm{u}1}$.

Proof. Taking $U_i = \beta\big(\sum_j w_{ij}V_j + I_i\big)$, lemma 5 states:

$$\begin{aligned}&\ln\big[1+\exp\big(\beta(\sum_j w_{ij}V_j + I_i)\big)\big] = \\ &\qquad \Leftrightarrow V_i\ln V_i \Leftrightarrow (1\Leftrightarrow V_i)\ln(1\Leftrightarrow V_i) + \beta\big(\sum_j w_{ij}V_iV_j + I_iV_i\big).\end{aligned} \tag{3.17}$$

[3] We notice that the usual argument for the admissibility of the saddle point approximation is that in the thermodynamic limit (that is for $n\to\infty$), the integrals are extremely dominated by the contributions which maximize the integrand [44, 76, 70]. We shall not further explore this here.

[4] Remember from section 2.1.5 that V_i can be interpreted as $P(S_i = 1)$.

By combining this result and equation (3.1), the expression (3.16) is found. The stationary points are found by resolving

$$\begin{aligned}\frac{\partial F_{\mathrm{u}2}}{\partial V_i} &= \Leftrightarrow\sum_j w_{ij}V_j \Leftrightarrow I_i + \tfrac{1}{\beta}(\ln V_i + 1 \Leftrightarrow \ln(1 \Leftrightarrow V_i) \Leftrightarrow 1) \\ &= \tfrac{1}{\beta}(\Leftrightarrow\beta(\sum_j w_{ij}V_j + I_i) + \ln\frac{V_i}{1 \Leftrightarrow V_i}) = 0. \end{aligned} \tag{3.18}$$

This yields the mean field equations (3.2). □

3.2 Properties

3.2.1 The relation between $F_{\mathrm{u}1}$ and $F_{\mathrm{u}2}$

We have found two approximations of the free energy, namely $F_{\mathrm{u}1}$ and $F_{\mathrm{u}2}$. This raises the question of how they are related. Let us start analyzing two simple examples. We take two binary stochastic Hopfield networks having the Hamiltonian

$$H_1(S) = S_1^2 \Leftrightarrow S_1 \quad \text{and} \quad H_2(S) = \Leftrightarrow S_1^2 + S_1. \tag{3.19}$$

The first one is the most simple model of an anti-ferromagnetic system, the second one of a ferromagnetic system (section 2.1.3). The corresponding free energy functions are

$$F_{\mathrm{u}1,H1}(V) = \Leftrightarrow V_1^2 \Leftrightarrow \tfrac{1}{\beta}\ln(1+\exp(\beta(\Leftrightarrow 2V_1+1))) \tag{3.20}$$

$$F_{\mathrm{u}2,H1}(V) = V_1^2 \Leftrightarrow V_1 + \tfrac{1}{\beta}(V_1\ln V_1 + (1\Leftrightarrow V_1)\ln(1\Leftrightarrow V_1)) \tag{3.21}$$

$$F_{\mathrm{u}1,H2}(V) = V_1^2 \Leftrightarrow \tfrac{1}{\beta}\ln(1+\exp(\beta(2V_1\Leftrightarrow 1))) \tag{3.22}$$

$$F_{\mathrm{u}2,H2}(V) = \Leftrightarrow V_1^2 + V_1 + \tfrac{1}{\beta}(V_1\ln V_1 + (1\Leftrightarrow V_1)\ln(1\Leftrightarrow V_1)). \tag{3.23}$$

The figures 3.1 and 3.2 show the free energies $F_{\mathrm{u}1}$ and $F_{\mathrm{u}2}$ of H_1, respectively H_2, for various values of β. In all cases, the stationary points of $F_{\mathrm{u}1}$ and $F_{\mathrm{u}2}$ coincide.

In the left-hand figure, the minima of $F_{\mathrm{u}2,H1}$ coincide with maxima of $F_{\mathrm{u}1,H1}$, all at $V_1 = 0.5$. Away from the stationary points, the free energy approximations differ substantially, where the approximation $F_{\mathrm{u}2,H1}$ looks the better one: H_1 is a convex function, so a free energy approximation is expected to be convex too since the energy contribution of noise is convex (section 2.1.2). Moreover, $\beta \to \infty$ (disappearing noise) implies that $\forall V_1 \in [0,1] : F_{\mathrm{u}2,H_1} \to H_1$, while this limiting property certainly does not hold for $F_{\mathrm{u}1,H1}$.

In the right-hand figure, the free energy approximations are more similar while the extrema of $F_{\mathrm{u}1,H2}$ and $F_{\mathrm{u}2,H2}$ have the same character. We also recognize a phase transition: for high values of $T = 1/\beta$, there exists one minimum at $V_1 = 0.5$, while at lower temperatures, we see one (metastable) maximum and

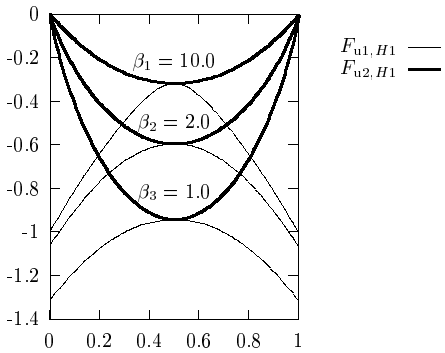


Figure 3.1: Free energies of H_1.

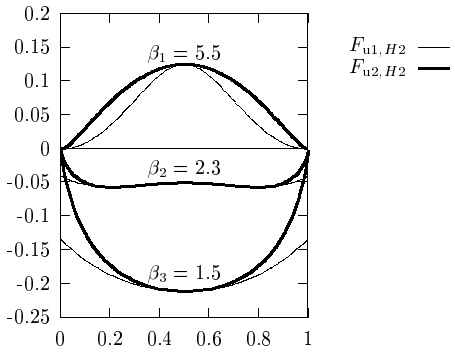


Figure 3.2: Free energies of H_2.

two (stable) minima, allowing the occurrence of a spontaneous magnetization. Although F_{u1} performs better now, the approximation by $F_{\mathrm{u2},H2}$ is still better: for low values of β, the approximation $F_{\mathrm{u2},H2}$ near $V_1 = 0$ or $V_1 = 1$ is superior.

The indicated attributes concerning the type of the extrema can further be underpinned by inspection of the second derivatives of F_{u1} and F_{u2}. Denoting the solutions of the mean field equations (3.2) by $\tilde{V}_i$, we find for the elements of the respective Hessians at stationary points:

$$\begin{aligned} h_{\mathrm{u1},ij} = \frac{\partial^2 F_{\mathrm{u1}}}{\partial V_i \partial V_j} &= w_{ij} \Leftrightarrow \beta \sum_k w_{ik} w_{kj} \frac{\exp(\beta(\sum_j w_{ij}\tilde{V}_j + I_i))}{(1+\exp(\Leftrightarrow\beta(\sum_j w_{kj}\tilde{V}_j + I_k)))^2} \\ &= w_{ij} \Leftrightarrow \beta \sum_k w_{ik} w_{kj} \tilde{V}_k (1 \Leftrightarrow \tilde{V}_k), \end{aligned} \tag{3.24}$$

$$h_{\mathrm{u2},ij} = \frac{\partial^2 F_{\mathrm{u2}}}{\partial V_i \partial V_j} = \begin{cases} \Leftrightarrow w_{ij} & \text{if } j \neq i \\ \Leftrightarrow w_{ii} + 1/(\beta \tilde{V}_i (1 \Leftrightarrow \tilde{V}_i)) & \text{if } j = i. \end{cases} \tag{3.25}$$

In the present example, in case of H_1 (where $w_{11} < 0$), we find

$$\forall \beta : h_{\mathrm{u1}} < 0 \ \wedge \ h_{\mathrm{u2}} > 0. \tag{3.26}$$

This confirms the (opposite) character of the extrema in the left figure. In case of H_2 (where $w_{11} > 0$), we find

$$h_{\mathrm{u1}} > 0 \ \wedge \ h_{\mathrm{u2}} > 0 \quad \text{if} \quad \beta < 1/(2\tilde{V}_1(1 \Leftrightarrow \tilde{V}_1)) = 2, \tag{3.27}$$

$$h_{\mathrm{u1}} < 0 \ \wedge \ h_{\mathrm{u2}} < 0 \quad \text{if} \quad \beta > 1/(2\tilde{V}_1(1 \Leftrightarrow \tilde{V}_1)) = 2. \tag{3.28}$$

This confirms the (same) character of the extrema in the right figure. In the mean time, we have calculated the critical temperature[5] being $T_{\mathrm{cr}} = 1/\beta_{\mathrm{cr}} = 0.5$. We

[5] In this case, the critical temperature can also be calculated by considering the equilibrium equations 2.34 [44]. They can be written as $V_1 = 1/(1+\exp(-\beta U_1)) \wedge V_1 = \frac{1}{2}U_1 + \frac{1}{2}$. For $T > T_{\mathrm{cr}} = 0.5$, the equations have only one solution $V_1 = 0.5$. For $T < T_{\mathrm{cr}} = 0.5$, there are 3 solutions.

further notice that inspection of (3.24) and (3.25) reveals that the noted phenomena concerning the character of the extrema of $F_{\mathrm{u}1}$ and $F_{\mathrm{u}2}$ may also occur in other cases.

Concluding this section, we notice that in general the use of the superior mean field approximation $F_{\mathrm{u}2}$ is preferred. However, the approximation $F_{\mathrm{u}1}$ will turn out to be of great theoretical importance in section 3.3.1.

3.2.2 The effect of noise

There is still another way to understand the relationship between the mean field approximation of the binary stochastic and the continuous Hopfield model. Here, the starting point is Hopfield's energy expression (2.32). Taking the sigmoid as the transfer function, we can elaborate the Hopfield term E_{h}, i.e., the sum of integrals $\sum_i \int_0^{V_i} g^{-1}(v)\mathrm{d}v$. Since $V_i = g(U_i) = (1 + e^{-\beta U_i})^{-1}$, we can write

$$U_i = \Leftrightarrow\frac{1}{\beta}\ln(\frac{1 \Leftrightarrow V_i}{V_i}) = g^{-1}(V_i), \tag{3.29}$$

and therefore

$$\int_0^{V_i} g^{-1}(v)\mathrm{d}v = \tfrac{1}{\beta}[(1 \Leftrightarrow V_i)\ln(1 \Leftrightarrow V_i) + V_i \ln V_i] = \Leftrightarrow\tfrac{1}{\beta}\mathcal{S}(V_i). \tag{3.30}$$

Thus, we have proven the following theorem:

Theorem 3.3. *If the sigmoid function is chosen as the transfer function in the continuous Hopfield model, then the energy E_{c} equals the free energy approximation $F_{\mathrm{u}2}$ of the stochastic binary Hopfield model. The Hopfield term E_{h} of the continuous model can physically be interpreted as the (approximation of the) thermal noise term $\Leftrightarrow\frac{1}{\beta}\mathcal{S}$ of* (2.9).

For the specific choice of the sigmoid as the transfer function, we can examine the effect of the temperature more thoroughly (compare Hopfield's discussion as mentioned at the end of section 2.3.2). In figure 3.3, the term (3.30) is visualized at various temperatures. The term is always non-positive and for $\beta \to \infty$, $\Leftrightarrow\frac{1}{\beta}\mathcal{S}(V_i) \to 0$, so in the limit of an annealing process, the noise term does not influence the extrema of the original cost function $E(V)$ of (2.32). For finite values of β, minima of $E(V)$ situated in a corner of the hypercube, are displaced toward the interior (see also figure 3.4). This is true for any finite value of β since

$$\frac{\partial E_{\mathrm{h}}}{\partial V_i}(V_i = 0) = \Leftrightarrow\infty \quad \text{and} \quad \frac{\partial E_{\mathrm{h}}}{\partial V_i}(V_i = 1) = \infty, \tag{3.31}$$

whereas the partial derivatives of $E(V)$ are always finite. The smaller β is, the larger is the displacement toward the interior.

The displacement noted should be considered a pretty feature of the model. First, it makes mean field annealing (section 2.1.5) possible, since the shift goes hand in hand with a smoothing effect on the energy landscape of $E(V)$ and gradually disappears if T is lowered. Second, by keeping the final temperature small but

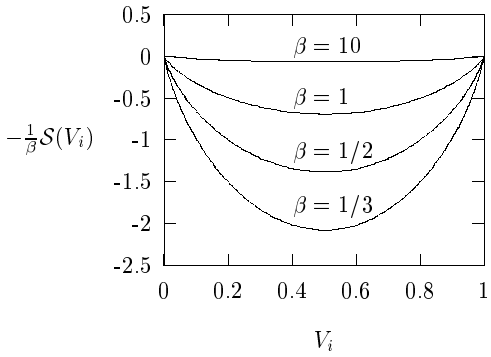


Figure 3.3: The term $\Leftrightarrow\frac{1}{\beta}\mathcal{S}(V_i)$ for various values of β.

positive, solutions are dragged away from corners of the hypercube $[0,1]^n$ causing the corresponding U-values of the neurons to be *finite*. We further notice that minima of $E(V)$ situated in the interior of the hypercube are also displaced by the effect of noise. The magnitude of the displacement strongly depends on the parameter value β.

In the literature, we occasionally encountered a slight confusion concerning the Hopfield term $E_{\rm h}(V)$. As we have seen, the term directly relates to the U_i-terms in the corresponding updating rules (2.30): $U_i = \partial E_{\rm h}/\partial V_i$. Takefuji considers the 'decay term' U_i 'harmful' and concludes (quote from pp. 6 and 7 in [82]):

> "Hopfield gives the motion equation of the i-th neuron (Hopfield and Tank 1985):
>
> $$\frac{dU_i}{dt} = -\frac{U_i}{\tau} - \frac{\partial E}{\partial V_i} \tag{3.32}$$
>
> (...) . Wilson and Pawley strongly criticized the Hopfield and Tank neural network through the Travelling Salesman Problem. Wilson and Pawley did not know what causes the problem. The use of the decay term $(-U_i/\tau)$ in Eq. 3.32 increases the computational energy function E under some conditions instead of decreasing it."

So Takefuji suggests, but does not prove that the problems which Wilson and Pawley [88] encountered, are caused by the decay term U_i/τ (in our formulations $\tau = 1$). We think this suggestion is not correct for two reasons. First, in his analysis, Takefuji does not add the Hopfield term $E_{\rm h}(V)$ to the energy function, but at the same time, he does take up the decay term U_i in equation (3.32). He then concludes, that the decay term is responsible for incrementing the cost function $E(V)$ under some conditions, making it thereby harmful. In fact, this conclusion on the increase of the cost function is correct, but it should not be considered harmful: we shall prove in the next section that the encountered energy increase precisely corresponds to the aforesaid displacement of solutions.

Second, analyzing the TSP, Wilson and Pawley applied the penalty method with many competing (sometimes called mutually 'frustrating') penalty terms: this soft approach should be considered the crucial factor for the poor results in their approach.

3.2.3 Why the decay term is *not* harmful

We already know from theorem 2.2, that under some general conditions, equation (3.32) continuously decreases $E_{\rm c}(V) = E(V) + E_{\rm h}(V)$ until an equilibrium point is reached. Takefuji argues in the following way that the cost function $E(V)$ *alone* may increase: using equation (3.32) with $\tau = 1$, it is seen that

$$\begin{aligned} \dot{E} &= \sum_i \frac{\partial E}{\partial V_i}\dot{V}_i = \sum_i (\Leftrightarrow\!\dot{U}_i \Leftrightarrow U_i)\dot{V}_i \\ &= \Leftrightarrow\!\sum_i (\dot{U}_i^2 + U_i\dot{U}_i)\frac{dV_i}{dU_i}. \end{aligned} \tag{3.33}$$

Because $dV_i/dU_i > 0$, a necessary condition for an increase of $E(V)$ can be stated as follows: there should be at least one i such, that

$$\dot{U}_i^2 + U_i\dot{U}_i < 0, \tag{3.34}$$

which is equivalent to

$$\Leftrightarrow\!U_i < \dot{U}_i < 0 \quad \text{or} \quad 0 < \dot{U}_i < \Leftrightarrow\!U_i. \tag{3.35}$$

These two conditions correspond precisely to a displacement of a solution toward the interior of the state space. We shall prove that the first condition results in a displaced minimum with a *lower* value of V_i (the second corresponds to a displacement with a *higher* value). The left inequality of $\Leftrightarrow\!U_i < \dot{U}_i < 0$ implies that

$$\Leftrightarrow\!U_i \Leftrightarrow\!\dot{U}_i < 0. \tag{3.36}$$

Using again (3.32) with $\tau = 1$, one finds:

$$\frac{\partial E}{\partial V_i} = \Leftrightarrow\!U_i \Leftrightarrow\!\dot{U}_i < 0, \tag{3.37}$$

so that E as function of V_i is decreasing.

The right-hand inequality of $\Leftrightarrow\!U_i < \dot{U}_i < 0$ implies that $\Leftrightarrow\!\dot{U}_i > 0$. Using once again (3.32) and the equation $U_i = \partial E_{\rm h}/\partial V_i$, one finds:

$$\frac{\partial E}{\partial V_i} + \frac{\partial E_{\rm h}}{\partial V_i} = \frac{\partial E}{\partial V_i} + U_i = \Leftrightarrow\!\dot{U}_i > 0, \tag{3.38}$$

so that the sum of E and $E_{\rm h}$ is increasing. The inequalities (3.37) and (3.38) together imply

$$\frac{\partial E_{\rm h}}{\partial V_i} > 0, \tag{3.39}$$

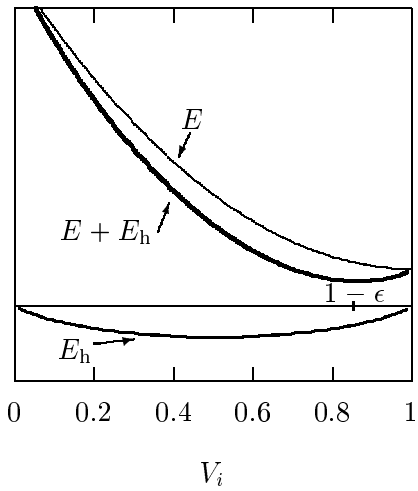


Figure 3.4: E, $E_{\rm h}$ and $E + E_{\rm h}$ as function of V_i.

so that $E_{\rm h}$ as a function of V_i is increasing. Therefore, $V_i > 0.5$. We have put this altogether in figure 3.4 (for the case that E has a minimum for $V_i = 1$). It should be clear now that the conditions (3.37) to (3.39) imply a displacement of the minimum of $E(V)$ to the interior, caused by the contribution of $E_{\rm h}(V)$.

It is easy to prove that the converse also holds: a displacement of a solution to a smaller value of V_i, caused by the Hopfield term, implies $\Leftrightarrow U_i < \dot{U}_i < 0$. Summarizing, we conclude that the conditions (3.35) which may cause an increase of the cost function $E(V)$, precisely correspond to a displacement of a solution to the interior of the state space. Since we argued in the previous subsection that such a displacement is a pretty feature of the model, the decay term is not at all a harmful one.

3.3 Generalizing the model

3.3.1 A first generalization step

In this subsection, we introduce a more general view on Hopfield neural networks which puts the analysis of section 3.1 in a wider context.

Theorem 3.4. *If (w_{ij}) is a symmetric matrix, then a mean field approximation of the free energy of stochastic binary Hopfield networks can also be stated as*

$$F_{\rm u3}(U,V) = \Leftrightarrow\tfrac{1}{2}\sum_{i,j} w_{ij}V_iV_j \Leftrightarrow\sum_i I_iV_i + \sum_i U_iV_i \Leftrightarrow\tfrac{1}{\beta}\sum_i \ln(1+\exp(\beta U_i)), \quad (3.40)$$

where the stationary points of $F_{\mathrm{u}3}$ are found at points of the state space for which

$$\forall i : V_i = \frac{1}{1 + e^{-\beta U_i}} \ \wedge \ U_i = \sum_j w_{ij} V_j + I_i. \tag{3.41}$$

Proof. Substitution of lemma 5 in the energy function $F_{\mathrm{u}2}$ of theorem 3.2 immediately yields the free energy expression (3.40). Resolving the system of equations $\forall i : \partial F_{\mathrm{u}3}/\partial U_i = 0, \partial F_{\mathrm{u}3}/\partial V_i = 0$ (compare 3.14) yields the equations (3.41) as solutions. □

The most interesting point of theorem 3.4 is the fact that the stationary points of $F_{\mathrm{u}3}$ coincide with the *complete* set of equilibrium conditions (2.34), provided that the sigmoid is the chosen transfer function. In fact, this clarifies (what may be clear intuitively) that, for a full description of the (continuous) Hopfield network, one should know both all input values (U_i) and all output values (V_i). Thus, it is actually better to call the set of vectors $\{U, V\}$ the system state of the neural net (instead of merely the vector V).

Just like $F_{\mathrm{u}2}$ is a Lyapunov function, so $F_{\mathrm{u}3}$ appears to be a Lyapunov function of the motion equations (2.30):

Theorem 3.5. *If (w_{ij}) is a symmetric matrix and if $\forall i : V_i = 1/(1 + \exp(\Leftrightarrow \beta U_i))$ is the transfer function, then the energy $F_{\mathrm{u}3}$ is a Lyapunov function for the motion equations* (2.30).

Proof. Knowing that the sigmoid function is a monotone increasing and differentiable function and that w_{ij} is a symmetric matrix, it follows that

$$\begin{aligned}
\dot{F}_{\mathrm{u}3} &= \sum_i \frac{\partial F_{\mathrm{u}3}}{\partial V_i}\dot{V}_i + \sum_i \frac{\partial F_{\mathrm{u}3}}{\partial U_i}\dot{U}_i && (3.42)\\
&= \sum_i \big(\Leftrightarrow \sum_j w_{ij} V_j \Leftrightarrow I_i + U_i \big)\dot{V}_i + \sum_i \big(V_i \Leftrightarrow \frac{1}{1 + e^{-\beta U_i}} \big)\dot{U}_i && (3.43)\\
&= \Leftrightarrow \sum_i \dot{U}_i \dot{V}_i = \Leftrightarrow \sum_i (\dot{U}_i)^2 \frac{\mathrm{d}V_i}{\mathrm{d}U_i} \leq 0. && (3.44)
\end{aligned}$$

In section 3.2.2, it is shown that the solution values of U_i are finite for finite values of β. Then, $F_{\mathrm{u}3}$ is bounded below. Therefore, execution of the motion equations (2.30) constantly decreases the value of $F_{\mathrm{u}3}$ until $\forall i : \dot{U}_i = 0$ and a (local) minimum has been reached. □

Inspection of the proof immediately yields a well-known [44], complementary set of motion equations for which $F_{\mathrm{u}3}$ or $\Leftrightarrow F_{\mathrm{u}3}$ might be a Lyapunov function:

Theorem 3.6. *If the matrix (w_{ij}) is symmetric and positive definite, then $F_{\mathrm{u}3}$ or alternatively, if the matrix (w_{ij}) is symmetric and negative definite, then $\Leftrightarrow F_{\mathrm{u}3}$ is a Lyapunov function for the motion equations*

$$\dot{V}_i = \frac{1}{1 + e^{-\beta U_i}} \Leftrightarrow V_i, \tag{3.45}$$

where

$$U_i = \sum_j w_{ij} V_j + I_i. \tag{3.46}$$

Proof. The proof again considers the time derivative of F_{u3}. If (w_{ij}) is positive definite, then

$$\dot{F}_{\mathrm{u3}} = \Leftrightarrow \sum_i \dot{V}_i \dot{U}_i = \Leftrightarrow \sum_i \dot{V}_i \sum_j \frac{\partial U_i}{\partial V_j} \dot{V}_j = \Leftrightarrow \sum_i \dot{V}_i \sum_j w_{ij} \dot{V}_j \leq 0. \tag{3.47}$$

If (w_{ij}) is negative definite, then $\Leftrightarrow \dot{F}_{\mathrm{u3}} \leq 0$. In both cases, updating conform (3.45) decreases the corresponding Lyapunov function until, finally, $\forall i : \dot{V}_i = 0$. □

3.3.2 A more general framework

Since equations (3.41) are a special case of (2.34) and similarly, equation (3.16) is a special case of (2.32), the question arises whether theorem 3.4 can be generalized to an energy expression of a continuous Hopfield network having neurons with an *arbitrary*[6] transfer function of the form $V_i = g(U_i)$. The following two theorems answer this, and other questions, affirmatively.

Theorem 3.7. *If (w_{ij}) is a symmetric matrix, then any stationary point of the energy F_{gf} defined by*

$$F_{\mathrm{gf}}(U,V) = \Leftrightarrow \tfrac{1}{2} \sum_{i,j} w_{ij} V_i V_j \Leftrightarrow \sum_i I_i V_i + \sum_i U_i V_i \Leftrightarrow \sum_i \int_0^{U_i} g(u)\mathrm{d}u \tag{3.48}$$

coincides with an equilibrium state of the continuous Hopfield neural network.

Proof. Resolving

$$\forall i : \partial F_{\mathrm{gf}} / \partial U_i = 0 \ \wedge \ \partial F_{\mathrm{gf}} / \partial V_i = 0, \tag{3.49}$$

the set of equilibrium conditions (2.34) is immediately found. □

In fact, the energy expression (3.48) can simply be derived from Hopfield's original expression (2.32) using partial integration. Having $V_i = g(U_i)$, we can write

$$\begin{aligned} \sum_i \int_0^{V_i} g^{-1}(v)\mathrm{d}v &= \sum_i \left[g^{-1}(v) v \right]_0^{V_i} \Leftrightarrow \sum_i \int_{g^{-1}(0)}^{U_i} v\mathrm{d}u \\ &= \sum_i U_i V_i \Leftrightarrow \sum_i \int_0^{U_i} g(u)\mathrm{d}u + c, \end{aligned} \tag{3.50}$$

[6] In fact, $V_i = g(U_i)$ is not completely arbitrary, since, for mathematical reasons, one should impose one or more general restrictions. E.g., $g(U_i)$ may have to be continuous, differentiable and-or integrable. To keep things simple, we mention these restrictions explicitly so far as they are of special interest. In other cases, the precise conditions are omitted and supposed to hold implicitly.

where $c = \Leftrightarrow\sum_i \int_{g^{-1}(0)}^{0} g(u)\mathrm{d}u$ is an unimportant constant which may be neglected[7]. Substitution of the result in (2.32) yields (3.48).

Theorem 3.8. *If (w_{ij}) is a symmetric matrix and if $\forall i : V_i = g(U_i)$ is a differentiable and monotone increasing function, then the energy function F_{gf} is a Lyapunov function for the motion equations*

$$\dot{U}_i = \sum_j w_{ij} V_j + I_i \Leftrightarrow U_i, \text{ where } V_i = g(U_i). \tag{3.51}$$

Proof. The proof is a direct generalization of the proof of theorem 3.5. □

Theorem 3.9. *If the matrix (w_{ij}) is symmetric and positive definite, then F_{gf} or alternatively, if the matrix (w_{ij}) is symmetric and negative definite, then $\Leftrightarrow F_{\mathrm{gf}}$ is a Lyapunov function for the motion equations*

$$\dot{V}_i = g(U_i) \Leftrightarrow V_i, \text{ where } U_i = \sum_j w_{ij} V_j + I_i. \tag{3.52}$$

Proof. The proof is a direct generalization of the proof of theorem 3.6. □

It is interesting to see that the conditions for which the updating rules (3.51) and (3.52) guarantee stability are so different. In the first case, stability only depends on the transfer function chosen. The corresponding condition that $V_i = g(U_i)$ should be differentiable and monotone increasing is generally easy to check. In the second case, stability depends on the structure of the optimization problem involved. The corresponding condition that the matrix (w_{ij}) should be positive or negative definite, may be difficult to check. The motion equations (3.51) are therefore in practice the preferable choice.

3.4 Computational results

We already discussed the fact that, in principle, the unconstrained continuous Hopfield model can be used to solve combinatorial optimization problems. The approach required is the soft one applying penalty terms. However, the computational results as known from literature are often very poor (section 2.4). On the other hand, we noticed in footnote 11 of the previous chapter that the penalty method may be useful for solving purely combinatorial problems. For this reason, we first confine ourselves to report certain experimental results involving one of such problems, namely the n-rook problem[8]. By doing this, we can simultaneously check some of the theoretical statements of this chapter, especially concerning the role of the temperature parameter. Secondly, we shall dwell upon the outcomes of a simple problem which is resolved using mean field annealing.

[7] It is not difficult to see that $g(0) = 0 \Rightarrow c = 0$.

[8] In addition, this problem acts as an introduction to the TSP, the experimental outcomes of which – together with those of other problems – will be reported in the next chapters.

3.4.1 The n-rook problem

The n-rook problem (NRP), which is strongly related to the famous n-queen problem, can be stated as follows: given an $n \times n$ chess-board the goal is to place n non-attacking rooks on the board. The problem is the same as the 'crossbar switch scheduling' problem, where the throughput of packets should be controlled in such a way that at any time, no two inputs may be connected to the same output and, vice versa, no output may be connected to more than one input simultaneously[9]. We may map the problem on the continuous Hopfield network as follows: if V_{ij} represents whether a rook is placed on the square of the chess-board with row number i and column number j, we search for a combination of V_{ij}-values such that the following constraints are fulfilled:

$$C_1 = \sum_{i,j} \sum_{k>j} V_{ij} V_{ik} = 0, \tag{3.53}$$

$$C_2 = \sum_{j,i} \sum_{k>i} V_{ij} V_{kj} = 0, \tag{3.54}$$

$$C_3 = \tfrac{1}{2} (\sum_{i,j} V_{ij} \Leftrightarrow n)^2 = 0. \tag{3.55}$$

$C_1 = 0$ implies that in any row at most one $V_{ik} \neq 0$, $C_2 = 0$ implies that in any column at most one $V_{kj} \neq 0$. $C_3 = 0$ in combination with $C_1 = C_2 = 0$ implies that precisely n rooks are placed on the board. The constraints fulfill the condition (2.37). C_1, C_2, C_3 can thus be used as penalty terms. The cost function to be minimized becomes

$$F_{\mathrm{u,nr}}(V) = \sum_{\alpha=1}^{3} c_\alpha C_\alpha(V) + E_{\mathrm{h}}(V). \tag{3.56}$$

The corresponding motion equation of this problem is

$$\dot{U}_{ij} = \Leftrightarrow \frac{\partial F_{\mathrm{u,nr}}}{\partial V_{ij}} = \Leftrightarrow c_1 \sum_{k \neq j} V_{ik} \Leftrightarrow c_2 \sum_{k \neq i} V_{kj} \Leftrightarrow c_3 (\sum_{i,j} V_{ij} \Leftrightarrow n) \Leftrightarrow U_{ij}, \tag{3.57}$$

where $V_{ij} = 1/(1 + \exp(\Leftrightarrow \beta U_{ij}))$. We note that in this problem the matrix $(w_{ij,kl})$ is symmetric so that (3.57) is expected to be stable. In the numerical simulation, we apply the approximation

$$\dot{U}_{ij} \approx \Delta U_{ij} / \Delta t. \tag{3.58}$$

Using random initializations of V_{ij} around 0.5, and choosing $\forall \alpha : c_\alpha = 1$, convergence is always present, provided Δt is chosen to be small enough. In case of $n = 4$, $\Delta t = 0.01$ is a good choice. At low temperatures, most of the neurons approach zero, while four of them become approximately one. In fact, one of the 24 possible solutions is ever found. The four neuron values which have become approximately one, are all equal and depend on β:

[9] As we shall see later on, the problem is also strongly related to the TSP. It has been resolved by Takefuji [82] too, although he applied another neural network. See further also [65].

β	$V_{ij} \approx 1$
1000	0.998392
200	0.993678
20	0.960207

Table 3.1: Solution values $V_{ij} \approx 1$ as function of β, in case $n = 4$.

At high temperatures however, all 16 V_{ij}'s become equal. E.g., for $\beta = 0.0002$ keeping the other parameters the same, we found $\forall i, j : V_{ij} = 0.499650$. The effect of a high noise level is present now.

In case of $n = 25$, $\beta = 1000$, $\Delta t = 0.0001$, we again found convergence, namely to one of the 25!, i.e. to one of the approximately 1.55×10^{25} solutions. In case of $n = 50$ and other parameters as before, we found convergence to one of the approximately 3.04×10^{64} solutions. 'Even' taking $n = 100$ with $\Delta t = 0.00005$, the system turns out to be stable. However, the calculation time now becomes an issue (several hours), since the neural network involved consists of 10 000 neurons, which have to be *sequentially* updated in the simulation for several thousand of times.

3.4.2 Mean field annealing

We finish this chapter by showing how the addition of noise can help to find the global minimum of a function. We look for the minimum of the Hamiltonian

$$E_{\mathrm{mf}}(V) = \Leftrightarrow V_1^2 + 1.5V_1, \tag{3.59}$$

where $V_1 \in [0, 1]$. The global boundary minimum of E_{mf} is the point (0,0), while (1,0.5) is the other (local) boundary minimum. Direct application of gradient descent on $E_{\mathrm{mf}}(V)$ with random initialization of V_1 on the interval (0.0,1.0) yields the global minimum in 75% of the cases, namely, if $V_1 \in (0.0, 0.75)$. However, in 25% of the cases, namely, if $V_1 \in (0.75, 1.0)$, the local boundary minimum is found.

If we apply mean field annealing by adding a sufficient amount of noise in the beginning, the global solution is *always* found. Figure (3.5) demonstrates how this can happen: at high temperatures (low values of β), the minimum of the free energy

$$F_{\mathrm{mf}}(V) = E_{\mathrm{mf}}(V) + E_{\mathrm{h}}(V) \tag{3.60}$$

occurs slightly left of $V_1 = 0.5$. On lowering the temperature, this minimum is gradually displaced and finally appears in the state $V_1 = 0$, while at the same time, the free energy F_{mf} more and more approximates the original E_{mf}. Even if the initial value of V_1 is in the interval (0.75,1.0), the right solution will still be found. A simulation using the corresponding motion equation

$$\dot{U}_1 = \Leftrightarrow \frac{\partial F_{\mathrm{mf}}}{\partial V_1} = 2V_1 \Leftrightarrow 1.5 \Leftrightarrow U_1, \tag{3.61}$$

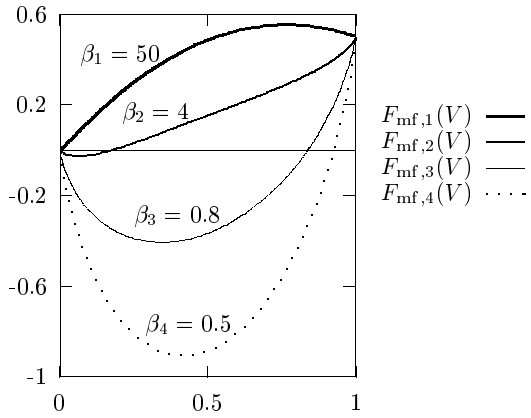


Figure 3.5: F_{mf} for various values of β.

(where as usual, $V_1 = 1/(1 + \exp(\Leftrightarrow \beta U_1))$), confirms this conjecture: e.g., having $\beta = 0.5$ and $V = 0.977$ initially, the network immediately relaxes to the equilibrium point at that temperature: we found (0.418,-0.907). On lowering the temperature step by step, the network continually relaxes to the new equilibrium point, which is gradually displaced towards the final limit point (0.0,0.0).

As has been pointed out in chapter 1, matters are usually much more complicated when real practical problems are tackled. E.g., problem instances of practical interest generally have energy functions in a high-dimensional space with many local minima widely scattered around, which gradually appear after each other on lowering the temperature. Thus, in those cases, the precise effect of the temperature is not quite clear and it is strongly connected to the actual structure of the energy surface of the problem.

Chapter 4

Constrained Hopfield networks

We take up the strong approach of dealing with the constraints as mentioned in section 2.4: the constraints are built-in in the neural network. Surprisingly, the selected constrained binary stochastic Hopfield neural network can be analyzed in a similar way as the unconstrained network of the previous chapter[1]. It leads to the insight that this constrained model also coincides, in mean field approximation, with an (adapted) continuous Hopfield net. Having elucidated this, we generalize the encountered free energy expressions: in three steps, the most general framework of continuous Hopfield models will emerge. As usual, we conclude by reporting some experimental results.

This chapter is largely structured like the previous one. Parts of this chapter have been published earlier in [9, 15, 17] or will be published soon [11]. A considerable part has been recorded in the technical reports [13, 16].

4.1 Once again, the mean field approximation

We restrict the space of allowed states of the neural net by imposing the constraint (2.40), that is, we impose

$$\sum_i S_i \Leftrightarrow 1 = 0. \tag{4.1}$$

Thus, only such states are admitted where exactly one of the neurons is on, all the others being off. The original state space $\{0,1\}^n$ is reduced to a much smaller one having the admissible n states $(1,0,0,\ldots,0), (0,1,0,\ldots,0), \ldots, (0,0,0,\ldots,1)$. In order to analyze this constrained neural network, we again adopt the modified version of Simic's approach [77].

[1] In fact, the research efforts which induced the (more difficult) analysis given here, for the greater part *preceded* those concerning the unconstrained networks.

Theorem 4.1. *If (w_{ij}) is a symmetric matrix, then a mean field approximation of the free energy of stochastic binary Hopfield networks submitted to the constraint* (4.1) *can be stated as*

$$F_{\mathrm{c1}}(V) = \tfrac{1}{2}\sum_{i,j} w_{ij}V_iV_j \Leftrightarrow \tfrac{1}{\beta}\ln\big[\sum_i \exp(\beta(\sum_j w_{ij}V_j + I_i))\big], \tag{4.2}$$

where the stationary points of F_{c1} are found at points of the state space for which

$$\forall i : V_i = \frac{\exp(\beta(\sum_j w_{ij}V_j + I_i))}{\sum_l \exp(\beta(\sum_j w_{lj}V_j + I_l))}. \tag{4.3}$$

Proof. The proof follows the scheme of the proof of theorem 3.1. This time, we shall indicate the partition function by Z_{hc}. Up to and including the exact equation (3.5), the proof is precisely the same. Thereupon, summation over the n states of the constrained space using lemma 6 yields,

$$Z_{\mathrm{hc}} = \frac{\int \exp\left[\Leftrightarrow\frac{\beta}{2}\sum_{i,j}\phi_i w_{ij}^{-1}\phi_j + \ln\sum_i \exp(\beta(\phi_i + I_i))\right]\prod_i \mathrm{d}\phi_i}{\int \exp\left[\Leftrightarrow\frac{\beta}{2}\sum_{i,j}\phi_i w_{ij}^{-1}\phi_j\right]\prod_i \mathrm{d}\phi_i}. \tag{4.4}$$

Writing

$$E_{\mathrm{hc}}(\phi, I) = \tfrac{1}{2}\sum_{ij}\phi_i w_{ij}^{-1}\phi_j \Leftrightarrow \tfrac{1}{\beta}\ln\sum_i \exp(\beta(\phi_i + I_i)), \tag{4.5}$$

partial differentiation of $E_{\mathrm{hc}}(\phi, I)$ this time leads to the saddle point equation

$$\tilde{\phi}_i = \sum_j w_{ij}\frac{\exp(\beta(\tilde{\phi}_i + I_i))}{\sum_l \exp(\beta(\tilde{\phi}_l + I_l))}. \tag{4.6}$$

Up till now, the calculations have been exact. The question arises, whether $\langle\hat{\phi}\rangle$ and $\tilde{\phi}$ are again related conform a saddle point approximation. Applying a modified, but very similar version of lemma 4, we arrive at the following saddle point approximation:

$$V_i \approx \Leftrightarrow\frac{\partial E_{\mathrm{hc}}(\tilde{\phi}, I)}{\partial I_i} = \frac{\exp(\beta(\tilde{\phi}_i + I_i))}{\sum_l \exp(\beta(\tilde{\phi}_l + I_l))}. \tag{4.7}$$

If we now substitute the approximation (4.7) in the exact formula (3.5), we indeed obtain (3.11), which states that in a saddle point approximation $\langle\hat{\phi}\rangle \approx \tilde{\phi}$. We further realize that again equation (3.12) holds and that it leads to (4.2) conform

$$F_{\mathrm{hc}} = \Leftrightarrow\frac{1}{\beta}\ln Z_{\mathrm{hc}} \approx E_{\mathrm{hc}}(\tilde{\phi}, I) \approx E_{\mathrm{hc}}(\langle\hat{\phi}\rangle, I) = F_{\mathrm{c1}}(V), \tag{4.8}$$

where the last equality is obtained by substitution of (3.5). Using the symmetry of w_{ij}, we finally find the equations (4.3) via

$$\begin{aligned}\frac{\partial F_{\mathrm{c1}}}{\partial V_i} &= \sum_j w_{ij}V_j \Leftrightarrow \tfrac{1}{\beta}\sum_k \frac{\beta\exp(\beta(\sum_j w_{kj}V_j + I_k))w_{ki}}{\sum_l \exp(\beta(\sum_j w_{lj}V_j + I_l))} \\ &= \sum_k w_{ik}(V_k \Leftrightarrow \frac{\exp(\beta(\sum_j w_{kj}V_j + I_k))}{\sum_l \exp(\beta(\sum_j w_{lj}V_j + I_l))}) = 0. \end{aligned} \tag{4.9}$$

These equations are the mean field equations of the constrained neural network [69, 70, 78]. Apparently, the first order saddle point approximation and the mean field analysis again yield the same results. This completes the proof. □

We may realize in another way that the first order saddle point and the mean field approximation are approaches of the same kind. By combining (4.7), (3.11), and (3.5), the saddle point approximation results into the mean field equations by realizing that

$$\begin{aligned} V_i &\approx \frac{\exp(\beta(\tilde{\phi}_i + I_i))}{\sum_l \exp(\beta(\tilde{\phi}_l + I_l))} \\ &\approx \frac{\exp(\beta(\langle\hat{\phi}_i\rangle + I_i))}{\sum_l \exp(\beta(\langle\hat{\phi}_l\rangle + I_l))} = \frac{\exp(\beta(\sum_j w_{ij}V_j + I_i))}{\sum_l \exp(\beta(\sum_j w_{lj}V_j + I_l))}. \end{aligned} \tag{4.10}$$

The *sign flip* (in the quadratic expression of the S_i's) we mentioned in the previous chapter is present again. Likewise, it can be undone producing a new free energy expression. This is stated more precisely in the following theorem, where the constrained subspace $\mathcal{C}$ is defined as the subspace of the state space $[0,1]^n$ for which $\sum_i V_i = 1$.

Theorem 4.2. *If (w_{ij}) is a symmetric matrix, then a mean field approximation of the free energy of stochastic binary Hopfield networks submitted to the constraint* (4.1) *can also be stated as*

$$F_{\mathrm{c2}}(V) = \Leftrightarrow\tfrac{1}{2}\sum_{i,j} w_{ij}V_iV_j \Leftrightarrow\sum_i I_iV_i + \tfrac{1}{\beta}\sum_i V_i\ln V_i, \tag{4.11}$$

where the stationary points of F_{c2}, considered as a function over the constrained space *$\mathcal{C}$, coincide with the (global) stationary points of F_{c1}.*

Proof. Taking $U_i = \sum_j w_{ij}V_j + I_i$, lemma 7 states:

$$\begin{aligned}\ln\sum_i \exp(\beta(\sum_j w_{ij}V_j + I_i)) = \\ \Leftrightarrow\sum_i V_i\ln V_i + \beta(\sum_{ij} w_{ij}V_iV_j + \sum_i I_iV_i).\end{aligned} \tag{4.12}$$

By combining this result and equation (4.2), the expression (4.11) for $F_{c2}(V)$ is found. In order to find the *constrained* stationary points of F_{c2}, a Lagrange multiplier term is added to (4.11) giving

$$L_{c2}(V,\lambda) = \Leftrightarrow\tfrac{1}{2}\sum_{i,j} w_{ij}V_iV_j \Leftrightarrow\sum_i I_iV_i + \tfrac{1}{\beta}\sum_i V_i \ln V_i + \lambda(\sum_i V_i \Leftrightarrow 1). \quad (4.13)$$

The stationary points of L_{c2} are found by resolving the following set of equations (4.14) and (4.15):

$$\frac{\partial L_{c2}}{\partial V_i} = \Leftrightarrow\sum_j w_{ij}V_j \Leftrightarrow I_i + \tfrac{1}{\beta}(1+\ln V_i) + \lambda = 0,\ i = 1,\ldots,n, \quad (4.14)$$

$$\frac{\partial L_{c2}}{\partial \lambda} = \sum_i V_i \Leftrightarrow 1 = 0. \quad (4.15)$$

From (4.14) it follows that

$$\frac{V_k}{V_i} = \frac{\exp(\sum_j w_{kj}V_j + I_k)}{\exp(\sum_j w_{ij}V_k + I_i)}. \quad (4.16)$$

Combining this result with (4.15), we obtain

$$1 = \sum_k V_k = V_i\frac{\sum_k \exp(\sum_j w_{kj}V_j + I_k)}{\exp(\sum_j w_{ij}V_k + I_i)}. \quad (4.17)$$

This equation implies the mean field equations (4.3). The solutions of these equations are stationary points of L_{c2} and constrained stationary points of F_{c2} as well. This completes the proof. □

It should be clear that a replacement of w_{ij} by $\Leftrightarrow w_{ij}$ and of I_i by $\Leftrightarrow I_i$ slightly modifies the above given theorems yielding mean field equations of the type

$$V_i = \frac{\exp(\Leftrightarrow\beta(\sum_j w_{ij}V_j + I_i))}{\sum_l \exp(\Leftrightarrow\beta(\sum_j w_{lj}V_j + I_l))}. \quad (4.18)$$

4.2 Properties

4.2.1 The relation between F_{c1} and F_{c2}

As in the unconstrained case, we have found two approximations of the free energy, namely F_{c1} and F_{c2}. We again want to understand how they are related. We start with an example. Suppose the function to be minimized is

$$H_3(S) = \tfrac{1}{2}(S_1^2 + 2S_2^2) \quad \text{subject to} \quad S_1 + S_2 = 1, \quad (4.19)$$

then the corresponding free energy expressions (from theorems 4.1 and 4.2) equal

$$F_{c1,H3}(V_1,V_2) = \Leftrightarrow\tfrac{1}{2}(V_1^2 + 2V_2^2) \Leftrightarrow \tfrac{1}{\beta}\ln[\exp(\Leftrightarrow\beta V_1) + \exp(\Leftrightarrow 2\beta V_2)], \quad (4.20)$$

$$F_{c2,H3}(V_1,V_2) = \tfrac{1}{2}(V_1^2 + 2V_2^2) + \tfrac{1}{\beta}(V_1\ln V_1 + V_2 \ln V_2). \quad (4.21)$$

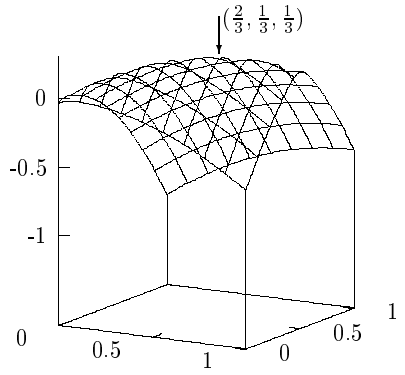


Figure 4.1: The free energy $F_{c1,H3}$.

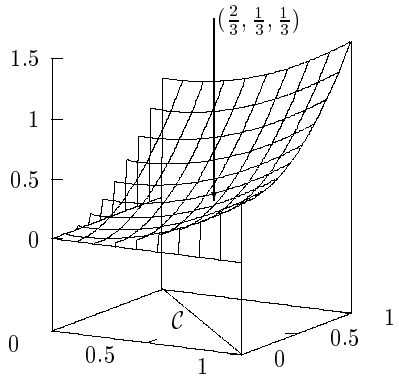


Figure 4.2: The free energy $F_{c2,H3}$.

A diagram of these functions is shown in the figures 4.1 and 4.2, with $\beta = 20$, which corresponds to a low noise level. The arrow denotes the point $(\frac{2}{3}, \frac{1}{3}, \frac{1}{3})$ which is the *global* maximum of $F_{c1,H3}$, respectively the *constrained* minimum of $F_{c2,H3}$, if noise is neglected. In this example, the constrained subspace $\mathcal{C}$ consists of the subspace of $[0,1]^2$ for which $V_1 + V_2 = 1$. In figure 4.3, $F_{c1,H3}$ and $F_{c2,H3}$ are shown over this constrained subspace. We notice the same phenomenon like in section 3.2 concerning the Hamiltonian E_1: $F_{c1,H3}$ and $F_{c2,H3}$ have coinciding stationary points with an opposite character of the extrema.
Likewise, analyzing the Hamiltonian

$$H_4(S) = \Leftrightarrow\frac{1}{2}(S_1^2 + 2S_2^2) \quad \text{subject to} \quad S_1 + S_2 = 1, \tag{4.22}$$

we found that $F_{c1,H4}$ and $F_{c2,H4}$ have extrema of the same kind. This is not further elaborated here.

Concluding this subsection, we observe that, within the constrained space $\mathcal{C}$, F_{c1} and F_{c2} seem to behave in the same way as F_{u1} and F_{u2} in the unconstrained case.

4.2.2 The effect of noise

The resemblance of the constrained Hopfield network to the unconstrained one reaches even further. The free energy F_{c2} can be interpreted as a function over a probability distribution V, where in this case

$$V_i = \langle S_i \rangle = \mathrm{P}(S_i = 1 \wedge \forall j \neq i : S_j = 0). \tag{4.23}$$

A closer investigation reveals that F_{c2}, like F_{u2}, is structured conform the general free expression (2.9). However, contrary to what we concluded in the unconstrained case, the neurons now have a mutually *dependent* contribution (of $\frac{1}{\beta} V_i \ln V_i$) to the entropy term. At high temperatures, the thermal noise energy

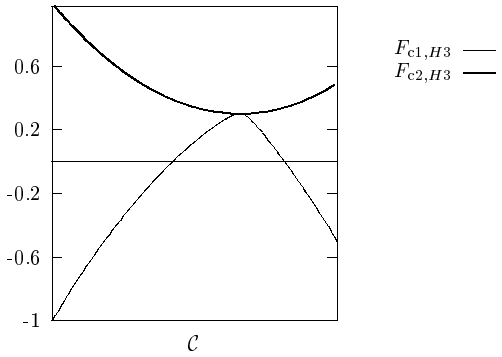


Figure 4.3: The energy expressions $F_{c1,H3}$ and $F_{c2,H3}$ in the constrained space $\mathcal{C}$.

dominates, this time yielding the constrained equilibrium solution $\forall i : V_i = 1/n$. This is easily recognized by resolving (using Lagrange's multiplier method)

$$\begin{aligned} &\text{minimize} && \tfrac{1}{\beta} \textstyle\sum_i V_i \ln V_i, \\ &\text{subject to :} && \textstyle\sum_i V_i \Leftrightarrow 1 = 0. \end{aligned} \tag{4.24}$$

Lowering the temperature corresponds to a decrease of thermal noise in the system and the details of the original cost function become visible. Therefore, mean field annealing can be applied.

4.3 Generalizing the model

4.3.1 A first generalization step

In this subsection, we introduce a general view on the binary constrained Hopfield model which puts the analysis of section 4.1 in a wider context. It will also enable us to analyze the stability properties of the constrained model.

Comparing the unconstrained and the constrained binary stochastic Hopfield model, the question may be posed whether the free energy approximation F_{c2} coincides with the energy of the continuous Hopfield model with the transfer function

$$V_i = g_i(U) = \frac{\exp(\beta U_i)}{\sum_l \exp(\beta U_l)}. \tag{4.25}$$

This transfer function is (of course) induced by the mean field equations (4.3). The corresponding continuous Hopfield network is visualized in figure 4.4. It is important to notice that the expression (4.11) for F_{c2} is *not* a special case of the general energy expression E_c (2.32) of the original continuous Hopfield model. This

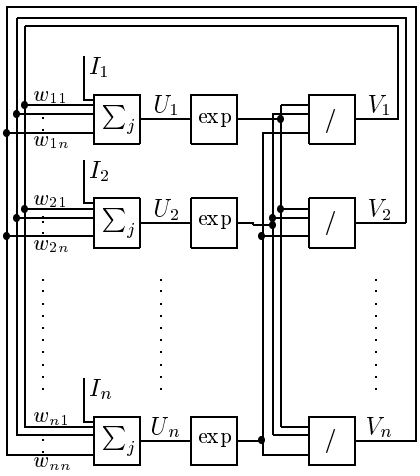


Figure 4.4: The constrained Hopfield network with equilibrium condition: $\forall i : U_i = \sum_j w_{ij} V_j + I_i$ and $V_i = \exp(U_i) / \sum_l \exp(U_l)$.

follows from the observation that

$$\sum_i \int_0^{V_i} g_i^{-1}(v) \mathrm{d}v \tag{4.26}$$

is not properly defined here, since $V_i = g_i(U)$ is now a function of $U_1, U_2, \ldots, U_n$ and not of U_i alone. Apparently, we have introduced a new, *adapted* continuous Hopfield network. The relation between this network and its stochastic counterpart is given by the following theorem.

Theorem 4.3. *If (w_{ij}) is a symmetric matrix, then a mean field approximation of the free energy of stochastic binary Hopfield networks submitted to the constraint* (4.1) *can also be stated as*

$$F_{\mathrm{c}3}(U,V) = \Leftrightarrow\tfrac{1}{2} \sum_{i,j} w_{ij} V_i V_j \Leftrightarrow \sum_i I_i V_i + \sum_i V_i U_i \Leftrightarrow \tfrac{1}{\beta} \ln(\sum_i \exp(\beta U_i)), \tag{4.27}$$

where the stationary points of $F_{\mathrm{c}3}$ are found at points of the state space for which

$$\forall i : V_i = \frac{\exp(\beta U_i)}{\sum_l \exp(\beta U_l)} \ \wedge \ U_i = \sum_j w_{ij} V_j + I_i. \tag{4.28}$$

Proof. Substitution of lemma 7 (in its original form) in the energy function $F_{\mathrm{c}2}$ of theorem 4.2 immediately yields expression $F_{\mathrm{c}3}$. Resolving the system of equations $\forall i : \partial F_{\mathrm{c}3}/\partial U_i = 0, \partial F_{\mathrm{c}3}/\partial V_i = 0$ yields the equations (4.28) as solutions. □

Again, we encounter the interesting phenomenon that the stationary points of a free energy approximation of a stochastic model coincide with the conditions

of equilibrium of a continuous Hopfield network. From the analysis presented above it also follows that, in the constrained case, Hopfield's theorem 2.2 does not hold. This induces the question whether, and if so, under which conditions, the adapted continuous Hopfield model converges. The following theorem answers this question.

Theorem 4.4. *If* (w_{ij}) *is a symmetric matrix, if* (4.25) *is used as the transfer function, and if, during updating, the Jacobian matrix* $J_{\rm g} = (\partial V_i/\partial U_j)$ *first is or becomes and then remains positive definite, then the energy* $F_{\rm c3}$ *is a Lyapunov function for the motion equations* (2.30)*.*

Proof. Assuming that the conditions of the theorem hold we may say that in the long run

$$\begin{aligned}
\dot{F}_{\rm c3} &= \sum_i \frac{\partial F_{\rm c3}}{\partial V_i}\dot{V}_i + \sum_i \frac{\partial F_{\rm c3}}{\partial U_i}\dot{U}_i \\
&= \sum_i (\Leftrightarrow \sum_j w_{ij}V_j \Leftrightarrow I_i + U_i)\dot{V}_i + \sum_i (V_i \Leftrightarrow \frac{\exp(\beta U_i)}{\sum_l \exp(\beta U_l)})\dot{U}_i \\
&= \Leftrightarrow \sum_i \dot{U}_i \sum_j \frac{\partial V_i}{\partial U_j}\dot{U}_j = \Leftrightarrow \dot{U}^T J_{\rm g}\dot{U} \leq 0. \qquad (4.29)
\end{aligned}$$

Since $F_{\rm c3}$ is bounded below at finite temperatures (for similar reasons as explained in the unconstrained case), its value decreases constantly until $\forall i : \dot{U}_i = 0$ and a local minimum is reached. □

Whether the general condition holds that the matrix $J_{\rm g}$ will become and remain positive definite, is not easy to say. Applying lemma 8, the symmetric matrix $J_{\rm g}$ is given by

$$\beta \begin{pmatrix} V_1(1 \Leftrightarrow V_1) & \Leftrightarrow V_1V_2 & \cdots & \Leftrightarrow V_1V_n \\ \Leftrightarrow V_2V_1 & V_2(1 \Leftrightarrow V_2) & \cdots & \Leftrightarrow V_2V_n \\ \vdots & \vdots & & \vdots \\ \Leftrightarrow V_nV_1 & \Leftrightarrow V_nV_2 & \cdots & V_n(1 \Leftrightarrow V_n) \end{pmatrix}. \qquad (4.30)$$

So we see that all diagonal elements of $J_{\rm g}$ are positive, while all non-diagonal elements are negative. Knowing that $\sum_i V_i = 1$, we argue that for large n in general

$$\forall i, \forall j, \forall k : V_iV_j << V_k(1 \Leftrightarrow V_k), \qquad (4.31)$$

although this statement is certainly not always true. Nevertheless, it is not unreasonable to expect that in many cases, the matrix $J_{\rm g}$ is dominated by the (positive) diagonal elements, making it positive definite[2]. For these reasons, *it is conjectured* that the motion equations (2.30) turn out to be stable in many practical applications.

[2] Under the given conditions, the symmetric matrix $J_{\rm g}$ has only positive eigenvalues, implying the definite positiveness of it [66].

As in the unconstrained case, inspection of the proof of the previous theorem immediately yields a complementary set of motion equations for which F_{c3} may be a Lyapunov function:

Theorem 4.5. *If the matrix (w_{ij}) is symmetric and positive definite, then F_{c3} or alternatively, if the matrix (w_{ij}) is symmetric and negative definite, then $\Leftrightarrow F_{c3}$ is a Lyapunov function for the motion equations*

$$\dot{V}_i = \frac{\exp(\beta U_i)}{\sum_l \exp(\beta U_l)} \Leftrightarrow V_i, \tag{4.32}$$

where

$$U_i = \sum_j w_{ij} V_j + I_i. \tag{4.33}$$

Proof. The proof is the same as the proof of theorem 3.6. □

4.3.2 A very general framework

It is remarkable, that the motion equations (2.30) of the continuous unconstrained model may still be applied using the constrained model, where the concrete transfer function (4.25) is a function of all inputs U_i. This poses the question whether those motion equations can still be applied if an *arbitrary*[3] function of the form $V_i = g_i(U) = g_i(U_1, U_2, \ldots, U_n)$ is used. This would yield a further generalization of (2.32), of section 3.3.2, and of the previous section. The following theorems answer this question.

Theorem 4.6. *Let $G(U) = G(U_1, U_2, \ldots, U_n)$ be a function for which*

$$\forall i : \frac{\partial G(U)}{\partial U_i} = g_i(U). \tag{4.34}$$

If (w_{ij}) is a symmetric matrix, then any stationary point of the energy

$$F_{\mathrm{vgf}}(U, V) = \Leftrightarrow \tfrac{1}{2} \sum_{i,j} w_{ij} V_i V_j \Leftrightarrow \sum_i I_i V_i + \sum_i U_i V_i \Leftrightarrow G(U) \tag{4.35}$$

coincides with an equilibrium state of the continuous Hopfield neural network defined by[4]

$$\forall i : V_i = g_i(U) \ \wedge \ U_i = \sum_j w_{ij} V_j + I_i. \tag{4.36}$$

Proof. Resolving

$$\forall i : \partial F_{\mathrm{vgf}} / \partial U_i = 0 \ \wedge \ \partial F_{\mathrm{vgf}} / \partial V_i = 0, \tag{4.37}$$

the set of equilibrium conditions (4.36) is found. □

[3] Again, certain general restrictions should be imposed on the transfer function: see footnote 6 of the previous chapter.

[4] Note, that the set of equilibrium conditions (4.36) is indeed a generalization of the set (2.34).

Theorem 4.7. *If the matrix (w_{ij}) is symmetric and if, during updating, the Jacobian matrix J_{g} first is or becomes and then remains positive definite, then the energy function F_{vgf} is a Lyapunov function for the motion equations*

$$\dot{U}_i = \sum_j w_{ij} V_j + I_i \Leftrightarrow U_i, \text{ where } V_i = g_i(U). \tag{4.38}$$

Proof. The proof is a direct generalization of the proof of theorem 4.4. □

Theorem 4.8. *If the matrix (w_{ij}) is symmetric and positive definite, then F_{vgf} or alternatively, if the matrix (w_{ij}) is symmetric and negative definite, then $\Leftrightarrow F_{\mathrm{vgf}}$ is a Lyapunov function for the motion equations*

$$\dot{V}_i = g_i(U) \Leftrightarrow V_i, \text{ where } U_i = \sum_j w_{ij} V_j + I_i. \tag{4.39}$$

Proof. The proof is a direct generalization of the proof of theorem 4.5. □

The conditions for which the updating rules (4.36) and (4.39) guarantee stability are quite different. Compared to the general framework of the unconstrained network (section 3.3.2), the condition on the transfer function (theorems 3.8 and 4.7) has become more difficult to check. On the other hand, the condition on the matrix (w_{ij}) (theorems 3.9 and 4.8) has remained the same and, often unfortunately hard to check.

4.3.3 The most general framework

We now ask ourselves whether the expression $U_i = \sum_j w_{ij} V_j + I_i$ can also be generalized, namely, to an arbitrary 'summation function' of type $U_i = h_i(V)$ (where an external input I_i is still admitted), and whether we can still give conditions that guarantee stability. Since we have done all the preparatory work, the affirmative answers to these questions are surprisingly simple. The result is what we have termed the 'most general framework of continuous Hopfield models'.

Theorem 4.9. *Let $G(U)$ be function defined like in theorem* 4.6 *and let in the same way $H(V) = H(V_1, V_2, \ldots, V_n)$ be a function for which*

$$\forall i : \frac{\partial H(V)}{\partial V_i} = h_i(V). \tag{4.40}$$

Then any stationary point of the energy

$$F_{\mathrm{mgf}}(U, V) = \Leftrightarrow H(V) + \sum_i U_i V_i \Leftrightarrow G(U) \tag{4.41}$$

coincides with an equilibrium state of the continuous Hopfield neural network defined by

$$\forall i : V_i = g_i(U) \ \wedge \ U_i = h_i(V) \tag{4.42}$$

Proof. Resolving

$$\forall i : \partial F_{\mathrm{mgf}}/\partial U_i = 0 \ \wedge \ \partial F_{\mathrm{mgf}}/\partial V_i = 0, \tag{4.43}$$

the set of equilibrium conditions (4.42) is found. □

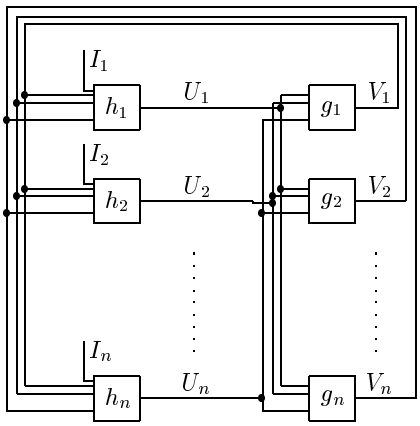


Figure 4.5: The most general continuous Hopfield network with equilibrium condition: $\forall i : U_i = h_i(V)$ and $V_i = g_i(U)$.

Theorem 4.10. *Suppose that $F_{\mathrm{mgf}}(U, V)$ is bounded below. Then the following statements hold:*
(a) If, during updating, the Jacobian matrix $J_{\mathrm{g}} = (\partial V_i/\partial U_j)$ first is or becomes and then remains positive definite, then the energy function F_{mgf} is a Lyapunov function for the motion equations

$$\dot{U}_i = h_i(V) \Leftrightarrow U_i, \ \text{where } V_i = g_i(U). \tag{4.44}$$

(b) If, during updating, the Jacobian matrix $J_{\mathrm{h}} = (\partial U_i/\partial V_j)$ first is or becomes and then remains positive definite, then the energy function F_{mgf} is a Lyapunov function for the motion equations

$$\dot{V}_i = g_i(U) \Leftrightarrow V_i, \ \text{where } U_i = h_i(V). \tag{4.45}$$

(c) If, during updating, the Jacobian matrices J_{g} and J_{h} first are or become and then remain positive definite, then the energy function F_{mgf} is a Lyapunov function for the motion equations

$$\dot{U}_i = h_i(V) \Leftrightarrow U_i \ \wedge \ \dot{V}_i = g_i(U) \Leftrightarrow V_i. \tag{4.46}$$

Proof. Assuming that the conditions as mentioned in (c) hold, we obtain

$$\begin{aligned}
\dot{F}_{\mathrm{mgf}} &= \sum_i \frac{\partial F_{\mathrm{mgf}}}{\partial V_i}\dot{V}_i + \sum_i \frac{\partial F_{\mathrm{mgf}}}{\partial U_i}\dot{U}_i \\
&= \sum_i (\Leftrightarrow h_i(V) + U_i) \sum_j \frac{\partial V_i}{\partial U_j}\dot{U}_j + \sum_i (V_i \Leftrightarrow g_i(U)) \sum_j \frac{\partial U_i}{\partial V_j}\dot{V}_j \\
&= \Leftrightarrow \sum_i \dot{U}_i \sum_j \frac{\partial V_i}{\partial U_j}\dot{U}_j \Leftrightarrow \sum_i \dot{V}_i \sum_j \frac{\partial U_i}{\partial V_j}\dot{V}_j \\
&= \Leftrightarrow \dot{U}^T J_{\mathrm{g}} \dot{U} \Leftrightarrow \dot{V}^T J_{\mathrm{h}} \dot{V} \leq 0. \qquad (4.47)
\end{aligned}$$

Then, the boundedness of F_{mgf} is sufficient to guarantee stability where at equilibrium $\forall i : \dot{U}_i = \dot{V}_i = 0$ implying the general equilibrium condition

$$\forall i : U_i = h_i(V) \quad \wedge \quad V_i = g_i(U). \qquad (4.48)$$

Using (4.47), the proofs of (a) and (b) can be done in the same way as the proof of theorem 4.4. □

Contemplating the results of this section, several striking observations emerge:

- By choosing appropriate transfer functions $g_i(U)$, several different types of constraints $C_\alpha(V)$ can be incorporated in continuous Hopfield networks. If they are chosen in such a way that the Jacobian matrix J_g first is or becomes and then remains positive definite, stability of the differential equations (4.44) is generally guaranteed. Alternatively, stability can be forced by choosing appropriate summation functions $h_i(V)$ while at the same time applying motion equations (4.45).
- By choosing appropriate summation functions $h_i(V)$, 'arbitrary' energy expressions $H(V)$ (not merely quadratic ones!) can be modelled by generalized continuous Hopfield networks. If they are chosen in such a way that the Jacobian matrix J_h first is or becomes and then remains positive definite, stability of the differential equations (4.45) is generally guaranteed. Alternatively, stability can be forced by choosing appropriate transfer functions $g_i(U)$ while at the same time applying motion equations (4.44).
- Taking the purely mathematical point of view, it is clear that the transfer functions $g_i(U)$ and the summation functions $h_i(V)$ are completely interchangeable.

An important consequence of these observations is the fact that within the introduced generalization, much more freedom exists for configuring continuous Hopfield neural networks. On the one hand, modelling an energy expression $H(V)$ is rather simple, since the corresponding summation functions (which should implement the desired energy expression $H(V)$) can be found by simply taking the corresponding partial derivatives $h_i(V) = \partial H(V)/\partial V_i$. It is interesting to note that, very recently, we came across two examples of this approach. In [24], 'higher

order neural networks' are introduced and appear to represent a much stronger heuristic to solving the Ising spin (checkerboard pattern) problem than that which is implemented by the Hopfield network. In [80], again higher order couplings between the neurons are admitted, just as well, to solve a combinatorial optimization problem (namely, a certain scheduling problem in behalf of 'cellular robotic systems'). It is argued that this approach avoids the spurious states [44] which are usual in neural networks without higher couplings.

On the other hand, building-in constraints may be more difficult: the transfer functions g_i should be chosen in such a way that the output values *always* fulfill the constraints, that is, for *any* set of input values U_i. The type of the built-in constraints effects the way the state space is walked through. E.g., having

$$\sum_i^n V_i = 1, \tag{4.49}$$

the constrained space consists of an (n-1)-dimensional flat hyperplane, while choosing

$$\prod_i^n V_i = 1, \tag{4.50}$$

this space is composed of an (n-1)-dimensional curved surface. But whatever the choice of the constraints may be, stability should be investigated whether in an analytical or in an experimental way. As we shall see in the next section on the results of certain simulations, the choice of right transfer functions even turns out quite complicated. The difficulties encountered there, are strongly related to the following question: *which conditions should the built-in constraints fulfill in order to guarantee that the continuous Hopfield network can be considered a mean field approximation of a corresponding stochastic network* (submitted to the same set of constraints)?

We conclude this theoretical section by observing that the original continuous Hopfield model, as introduced in section 2.3.2, beautifully fits into the most general framework presented here: having monotone increasing, differentiable transfer functions $V_i = g(U_i)$, the Jacobian matrix J_g is positive definite since all its diagonal elements are positive while all its non-diagonal elements equal zero. Using the motion equations (4.44) with $h_i(V) = \sum_j w_{ij}V_j + I_i$, stability is guaranteed conform theorem 4.10.

4.4 Computational results

The object of presenting the computational results of some experiments here, is not to give an exhaustive list of all possible ways the given theory of this chapter can be applied. Instead, the more modest objective is to show that the derived general theories are not falsified by the elementary tests we performed, and that, at

the same time, these tests yielded certain encouraging, informative, and startling results which invite to do more practical research in times to come[5].

4.4.1 A first toy problem

Let us start with a very simple experiment concerning constrained optimization:

$$\text{minimize } V_1^2 + 2V_2^2 + 3V_3^2 + 4V_4^2 \quad \text{subject to : } V_1 + V_2 + V_3 + V_4 = 1. \tag{4.51}$$

We apply the motion equations (2.30) with transfer function (4.25), which implies that the constraints are enforced in the strong sense. As has been mentioned in section 4.3.1, stability can be hoped for, but can not be guaranteed. Taking random initializations, we found the correct solution in all cases. Choosing $\beta = 20$ (low temperature), the solution $V_1 = 0.471,\ V_2 = 0.244,\ V_3 = 0.163,\ V_4 = 0.122$ is obtained. This corresponds precisely to the location of the constrained minimum. On the other hand, taking $\beta = 0.0001$, the equilibrium solution $V_1 = 0.250,\ V_2 = 0.250,\ V_3 = 0.250,\ V_4 = 0.250$ is found, showing the expected effect of a high thermal noise level.

4.4.2 A second toy problem

A second simple problem concerns a test whether *non-quadratic* cost functions can be tackled using the most general framework of continuous Hopfield networks having certain built-in constraints (section 4.3.3). We consider the following problem:

$$\text{minimize } \Leftrightarrow V_1^2 V_2^3 + V_2^5 \quad \text{subject to : } V_1 + V_2 = 1. \tag{4.52}$$

The corresponding motion equations are

$$\dot{U}_1 = 2V_1V_2^3 \Leftrightarrow U_1, \tag{4.53}$$

$$\dot{U}_2 = 3V_1^2V_2^2 \Leftrightarrow 5V_2^4 \Leftrightarrow U_2, \tag{4.54}$$

where

$$V_i = \frac{\exp(\beta U_i)}{\exp(\beta U_1) + \exp(\beta U_2)}. \tag{4.55}$$

Applying random initializations, we always found a monotone decreasing function $F_{\text{mgf}}(V)$ and the correct solutions. Taking $\beta = 0.001$, the encountered solution values are $V_1 = 0.5001$ and $V_2 = 0.4999$. Choosing $\beta = 50$, $V_1 = 0.617$ and $V_2 = 0.383$ are found, which approach the exact solution values (without any noise) in the interval $[0, 1]$, being $V_1 = 0.625$ and $V_2 = 0.375$.

[5] Actually, some of the experimental results presented here, have been obtained quite recently. They were induced by the most general framework, whose final formulation dates from only a couple of months ago. There is still much work to do in order to understand all capabilities of this framework.

4.4.3 An informative third toy problem

This third toy problem is set up in order to test whether cost functions submitted to *asymmetric* linear constraints can be resolved successfully. We consider the following problem:

$$\text{minimize } 2V_1^2 + V_2^2 \quad \text{subject to : } V_1 + 2V_2 = 1. \tag{4.56}$$

The corresponding motion equations are

$$\dot{U}_1 = \Leftrightarrow 4V_1 \Leftrightarrow U_1, \tag{4.57}$$
$$\dot{U}_2 = \Leftrightarrow 2V_2 \Leftrightarrow U_2. \tag{4.58}$$

Now the problem is how to define the transfer functions. In fact, there are several possibilities, e.g.,

$$V_1 = \frac{\exp(\beta U_1)}{\exp(\beta U_1) + \exp(\beta U_2)} \quad \text{and} \quad V_2 = \frac{\exp(\beta U_2)/2}{\exp(\beta U_1) + \exp(\beta U_2)} \tag{4.59}$$

or

$$V_1 = \frac{\exp(\beta U_1)}{\exp(\beta U_1) + 2\exp(\beta U_2)} \quad \text{and} \quad V_2 = \frac{\exp(\beta U_2)}{\exp(\beta U_1) + 2\exp(\beta U_2)}. \tag{4.60}$$

Applying random initializations, we always found convergence. However, the solutions found did *not* approximate the exact solution $V_1 = 1/9$ and $V_2 = 4/9$.

Inspection of equations (4.59) and (4.60) reveals that in both cases, $V_1 \in [0, 1]$ and $V_2 \in [0, 0.5]$. Thus, we have lost the usual property that

$$\forall i : V_i \in [0, 1]. \tag{4.61}$$

This observation inspired us to look for a modification of the original problem such that it can be mapped onto a network having constraints that yet fulfill condition (4.61). Eventually, we tested the following formulation of the problem:

$$\text{minimize } 2V_1^2 + \tfrac{1}{2}V_2^2 + \tfrac{1}{2}V_3^2 \quad \text{subject to : } V_1 + V_2 + V_3 = 1, V_2 = V_3. \tag{4.62}$$

The corresponding motion equations are

$$\dot{U}_1 = \Leftrightarrow 4V_1 \Leftrightarrow U_1, \tag{4.63}$$
$$\dot{U}_2 = \Leftrightarrow V_2 \Leftrightarrow U_2, \tag{4.64}$$
$$\dot{U}_3 = \Leftrightarrow V_3 \Leftrightarrow U_3, \tag{4.65}$$

where the transfer function of all neurons equals

$$V_i = \frac{\exp(\beta U_i)}{\sum_l \exp(\beta U_l)}. \tag{4.66}$$

Having $V_2 = V_3$ (after a correct initialization), equation (4.65) exactly coincides with (4.64). It therefore suffices in practice to merely apply motion equations (4.63)

and (4.64), where V_1 and V_2 are defined conform (4.60). The difference between this model and the previous one, comes from the difference between equations (4.58) and (4.64).

Applying random initializations, we always found convergence and this time, also the correct solution! Taking $\beta = 50.0$, the solution $V_1 = 0.117$, $V_2 = 0.441$ is found, which approximates the afore-mentioned exact solution of the original problem. Taking $\beta = 0.0001$, the expected solution values at high temperature are encountered, namely $V_1 = 0.333$ and $V_2 = 0.333$.

An important conclusion

The last example shows that the general framework can not be used groundless. The results also set us conjecture that a property like (4.61), expressing that all neurons should belong to the same interval, may be essential. Furthermore, it should be clear that the approach of this section to tackle asymmetric linear constraints can easily be generalized, that is, constraints of the type

$$\sum_j a_j V_j = c, \quad a_j, c \in \mathbb{R}, \tag{4.67}$$

can normally be grappled with in the way shown. This is not further elaborated here.

4.4.4 A startling fourth toy problem

Still another experiment has been performed in order to test whether an alternative type of constraints can be built-in successfully. Moreover, it is tried to solve the problem using two different sets of motion equations. We consider the following problem:

$$\text{minimize } 2V_1^2 + V_2^2 \quad \text{subject to : } V_1 * V_2 = 1. \tag{4.68}$$

It is easy to check that the exact solutions of this problem are $V_1 = \sqrt[4]{0.5} \approx 0.841$ and $V_2 = \sqrt[4]{2} \approx 1.189$. Using the differential equations (4.44), the concrete motion equations are

$$\dot{U}_1 = \Leftrightarrow 4V_1 \Leftrightarrow U_1, \tag{4.69}$$
$$\dot{U}_2 = \Leftrightarrow 2V_2 \Leftrightarrow U_2, \tag{4.70}$$

where we take

$$V_i = \frac{\exp(\beta U_i)}{\sqrt{\exp(\beta(U_1 + U_2))}}. \tag{4.71}$$

The last equation (which has been found after some tries and guesses) indeed implies that $V_1 * V_2 = 1$. Alternatively, using the type of differential equations (4.45),

the concrete motion equations are

$$\dot{V}_1 = \frac{\exp(\beta U_1)}{\sqrt{\exp(\beta(U_1+U_2))}} \Leftrightarrow V_1, \tag{4.72}$$

$$\dot{V}_2 = \frac{\exp(\beta U_2)}{\sqrt{\exp(\beta(U_1+U_2))}} \Leftrightarrow V_2, \tag{4.73}$$

where $U_1 = \Leftrightarrow 4V_1$ and $U_2 = \Leftrightarrow 2V_2$.

Applying random initializations and $\Delta t = 0.001$, we found proper convergence for all values of $\beta \in [\Leftrightarrow 0.19, 20]$, while for values outside this interval the motion equations were (nearly always) divergent. Both models behaved the same, and some solutions are given in table 4.1. Actually, these solution values do not

β	V_1	V_2
- 0.1	1.1555	0.8654
0.1	0.9258	1.0802
0.5	0.8160	1.2254
1.0	0.7740	1.2919
2.0	0.7449	1.3425
10.0	0.7155	1.3976
20.0	0.7114	1.4057

Table 4.1: Solutions values of V_1 and V_2 as function of β

falsify the theoretical conjectures of section 4.3.3. However, again we meet the phenomenon that we did not solve our original optimization problem. Likewise, the values of V_1 and V_2 do not fulfill condition (4.61). The effect of the controlling parameter β has been changed too: neither solution values are dragged towards the center of the solution space for low values of β (high temperatures), nor the solutions found approximate the solution of the original problem at low temperatures. Apparently, the free energy F_{mgf} does not approximate the original cost function for low values of β! [6]

A second important conclusion

The aforesaid computational outcomes show that one should be very careful in interpreting the results of the most general framework in case of building-in new types of constraints. The quite fundamental issue at stake is that the usual statistical mechanical interpretation of the continuous Hopfield model (where $1/\beta$ corresponds to a pseudo-temperature) does not hold for every set of built-in constraints. This issue raises the question as mentioned in the end of section 4.3.3, which, alternatively, can be stated as: which conditions relating to the built-in constraints can guarantee that the free energy F_{mgf} (as defined in (4.41)), can be writ-

[6] Perhaps, this observation does not come as a surprise. The complete statistical mechanical interpretation has shut down: because of definition (4.71), V_i can impossibly be associated with a probability.

ten in the standard form (2.8) as known from statistical mechanics? This question still begs for an answer.

4.4.5 The n-rook problem revisited

We here return to the constrained model that was analyzed extensively at the beginning of this chapter. Since part of the constraints of the NRP can be built-in in the neural network, whereby at the same time the space of admissible states is considerably limited, this partially strong approach is expected to work better than the purely soft approach applied in section 3.4. Here, the V_{ij} are chosen in such a way that

$$\forall i : \sum_j V_{ij} = 1, \tag{4.74}$$

implying that in every row, the sum of occupied squares of the chess-board equals one. It now suffices to minimize the cost function

$$F_{\mathrm{c,nr}}(V) = c_1 C_2(V) + E_h(V), \tag{4.75}$$

since C_2 enforces that in any column j at most one $V_{k,j} \neq 0$. The corresponding motion equation is simply

$$\dot{U}_{ij} = \Leftrightarrow \frac{\partial F_{\mathrm{c,nr}}}{\partial V_{ij}} = \Leftrightarrow c_2 \sum_{k \neq i} V_{kj} \Leftrightarrow U_{ij}, \quad \text{where} \quad V_{ij} = \frac{\exp(\beta U_{ij})}{\sum_l \exp(\beta U_{il})}. \tag{4.76}$$

We notice that the matrix $(w_{ij,kl})$ is still a symmetric one. A little analysis may clarify how the state space is limited. For that purpose, we consider the binary model with neurons S_{ij} (remember that $V_{ij} = \langle S_{ij} \rangle$). In the soft approach, all n^2 neurons may independently have value 0 or 1, so then there are $2^{n \times n}$ different neural net states. In the strong approach, every row has n states, so in that case, there are n^n different states. The following table shows both quantities as function of n:

n	$2^{n \times n}$	n^n
1	2	1
2	16	4
3	524	27
4	65536	256
p	$2^{p \times p}$	$2^{p \log_2 p}$

Table 4.2: $2^{n \times n}$ and n^n as function of n.

So for large values of n, the number of admissible states differ substantially.
The experimental outcomes confirm the conjecture that the constrained network behaves much better. Using the numerical approximation (3.58), again with random initializations and taking $\Delta t = 0.01$, convergence is always present provided

the penalty weight is set large enough. At low temperatures, the effect of noise is small as can be seen from table 4.3, where the neural outputs that are close to 1 are shown. If the temperature is increased slightly more, a rapid phase transition occurs: for $\beta = 0.3$, the solution values become almost equal conform $V_{ij} \approx 0.2500$.

β	$V_{ij} \approx 1$
10	1.0000
1	0.9999
0.5	0.9767

Table 4.3: Solution values $V_{ij} \approx 1$ as function of β, in case $n = 4$.

The larger n is the chosen, the larger the penalty weight c_2 should be taken in order to arrive at equilibrium. This contributes to speed up the convergence process. The convergence time is invariably only *a small fraction* of the convergence time of the pure penalty method. E.g., taking $c_2 = 50$, only a few minutes are needed in order to find a solution for the 150-rook problem while many hours would be needed if the soft approach was applied!

It is interesting to note that the values of the neurons initially seem to change in a chaotic way: the value of the $F_{\mathrm{c,nr}}$ strongly oscillates in an unclear way. However, after a certain period, the network suddenly finds its way to a stable minimum, at the same time rapidly minimizing the value of the cost function.

Chapter 5

The Hopfield-Lagrange model

As mentioned in section 2.4, a third way of coping with constraints is the use of Lagrange multipliers. In order to better understand the behavior of the corresponding Hopfield-Lagrange model (introduced in section 2.5), we here start by analyzing its stability properties by means of a new Lyapunov function. Next, we prove that, under certain conditions, the model degenerates into a so-called *dynamic penalty method* and we dwell on the effect of so-termed hard constraints. We thereafter scrutinize the stability of the 'constrained Hopfield-Lagrange model', which is a combination of the constrained Hopfield model of the previous chapter with the multiplier approach of this chapter. In this case, an 'arbitrary' (see, again, footnote 6 of chapter 3) cost function is admitted as well as 'arbitrary' transfer functions can be chosen.

We finish by presenting the computational results of various experiments both with unconstrained and constrained Hopfield-Lagrange networks. Parts of this chapter have been published earlier in [12, 14], much has also been recorded in technical report [13].

5.1 Stability analysis, the unconstrained model

5.1.1 Some reconnoitrings

For convenience, we again state the equations of the Hopfield-Lagrange model, which is based on the use of Lagrange multipliers in combination with the original unconstrained continuous Hopfield model. The energy of this model[1] is given by

$$E_{\rm hl}(V,\lambda) = E(V) + \sum_{\alpha} \lambda_\alpha C_\alpha(V) + E_{\rm h}(V) \tag{5.1}$$

$$= \Leftrightarrow\tfrac{1}{2}\sum_{i,j} w_{ij}V_iV_j \Leftrightarrow\sum_{i} I_iV_i + \sum_{\alpha} \lambda_\alpha C_\alpha(V) + E_{\rm h}(V) \tag{5.2}$$

[1] Although we have shown in theorem 3.3 that $E_{\rm h}(V)$ is a thermal noise term, $E_{\rm hl}(V,\lambda)$ does not turn out to be a properly bounded free energy (see below). This is why we do *not* replace $E_{\rm hl}$ by $F_{\rm hl}$.

having the corresponding set of differential equations

$$\dot{U}_i = \Leftrightarrow\frac{\partial E_{\rm hl}}{\partial V_i} = \sum_j w_{ij}V_j + I_i \Leftrightarrow \sum_\alpha \lambda_\alpha \frac{\partial C_\alpha}{\partial V_i} \Leftrightarrow U_i, \tag{5.3}$$

$$\dot{\lambda}_\alpha = +\frac{\partial E_{\rm hl}}{\partial \lambda_\alpha} = C_\alpha(V), \tag{5.4}$$

where $V_i = g(U_i)$. Let us first take a simple toy problem in order to try to understand why the gradient ascent or sign flip as referred to in section 2.4 is needed in (5.4). The problem is stated as follows:

$$\begin{aligned} &\text{minimize } E(V) = V_1^2, \\ &\text{subject to: } V_1 \Leftrightarrow 1 = 0. \end{aligned} \tag{5.5}$$

Using the Hopfield-Lagrange model with the sigmoid as the transfer function, the energy function (5.2) equals

$$E_{\rm hl,t}(V,\lambda) = V_1^2 + \lambda_1(V_1 \Leftrightarrow 1) + \frac{1}{\beta}((1 \Leftrightarrow V_1)\ln(1 \Leftrightarrow V_1) + V_1 \ln V_1). \tag{5.6}$$

At low temperatures, this energy expression simply reduces to an expression of the form (2.43)

$$E_{\rm pb,t}(V,\lambda) = V_1^2 + \lambda_1(V_1 \Leftrightarrow 1), \tag{5.7}$$

which is visualized in figure 5.1. To find the critical point $(V_1, \lambda_1) = (1, \Leftrightarrow 2)$ using a

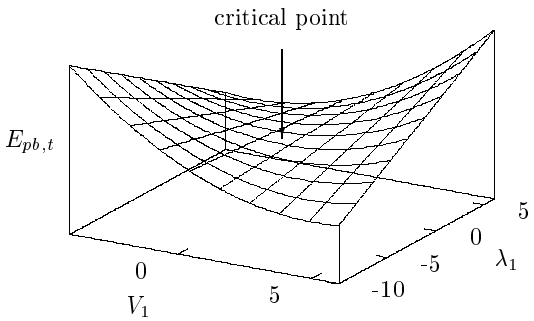


Figure 5.1: The energy landscape of $V_1^2 + \lambda_1(V_1 \Leftrightarrow 1)$.

direct gradient method, we should apply a gradient *des*cent with respect to V_1 and, at the same time, a gradient *as*cent with respect to λ_1: the result is a spiral motion towards the critical point. We shall see that the gradient ascent is also needed if the Hopfield-Lagrange network is applied.

Let us pose the question under which circumstances the set of differential equations (5.3) and (5.4) converge. A natural approach is to try the energy (5.2) as Lyapunov function. Taking the time derivative, we obtain

$$\begin{aligned}\dot{E}_{\mathrm{hl}}(V,\lambda) &= \sum_i (\Leftrightarrow \sum_j w_{ij}V_j \Leftrightarrow I_i + \sum_\alpha \lambda_\alpha \frac{\partial C_\alpha}{\partial V_i} + U_i)\dot{V}_i + \sum_\alpha \dot{\lambda}_\alpha C_\alpha \\ &= \Leftrightarrow \sum_i \dot{U}_i^2 \frac{\mathrm{d}V_i}{\mathrm{d}U_i} + \sum_\alpha C_\alpha^2. \end{aligned} \tag{5.8}$$

This reveals that *if* the constraints are (and remain) fulfilled, stability is guaranteed by using a transfer function whose derivative is always positive. However, if the constraints are not fulfilled, $\dot{E}_{\mathrm{hl}}$ is not necessarily monotone decreasing. Thus, we realize that stability is not guaranteed if we apply a random initialization of the neural network. On the other hand, if we would apply a gradient descent in equation (5.4), then $\dot{E}_{\mathrm{hl}}(V,\lambda) \leq 0$. Nevertheless, this does not work since $E_{\mathrm{hl}}(V,\lambda)$ is generally not bounded below (see also figure 5.1). The corresponding differential equations may be unstable and in practice, they appear to be so.

Therefore, we adhere to the original set of differential equations (5.3), (5.4) and adopt the approach of Platt and Barr from section 2.5 as our guiding principle for analyzing them.

5.1.2 A potential Lyapunov function

In the afore-stated approach, physics is the source of inspiration. We want to set up an expression of the sum of kinetic and potential energy. For that purpose, the differential equations (5.3) and (5.4) are taken together, yielding one second-order differential equation:

$$\ddot{U}_i = \Leftrightarrow \sum_j a_{ij} \frac{\mathrm{d}V_j}{\mathrm{d}U_j}\dot{U}_j \Leftrightarrow \dot{U}_i \Leftrightarrow \sum_\alpha C_\alpha \frac{\partial C_\alpha}{\partial V_i}, \tag{5.9}$$

where (a_{ij}) equals (2.48), that is,

$$a_{ij} = \Leftrightarrow w_{ij} + \sum_\alpha \lambda_\alpha \frac{\partial^2 C_\alpha}{\partial V_i \partial V_j}. \tag{5.10}$$

Equation (5.9) coincides with the equation for a damped harmonic motion of a mass system, where the mass equals 1, the spring constant equals 0, and where the external force of the system equals $\Leftrightarrow \sum_\alpha C_\alpha \partial C_\alpha / \partial V_i$.

Theorem 5.1. *If the matrix (b_{ij}) defined by*

$$b_{ij} = a_{ij}\frac{\mathrm{d}V_j}{\mathrm{d}U_j} + \delta_{ij} \tag{5.11}$$

(δ_{ij} being the Kronecker delta) first is or becomes and then remains positive definite, then the energy function

$$E_{\mathrm{kin+pot}} = \sum_i \tfrac{1}{2}\dot{U}_i^2 + \sum_{i,\alpha} \int_0^{U_i} C_\alpha \frac{\partial C_\alpha}{\partial V_i}\mathrm{d}u \tag{5.12}$$

is a Lyapunov function[2] *for the set of motion equations* (5.3) *and* (5.4)*.*

Proof. Taking the time derivative of $E_{\mathrm{kin+pot}}$ and using (5.9) as well as the positive definiteness of (b_{ij}), we obtain

$$\begin{aligned}
\dot{E}_{\mathrm{kin+pot}} &= \sum_i \dot{U}_i \ddot{U}_i + \sum_{i,\alpha} C_\alpha \frac{\partial C_\alpha}{\partial V_i} \dot{U}_i \\
&= \sum_i \dot{U}_i \left(\Leftrightarrow \sum_j a_{ij} \frac{\mathrm{d}V_j}{\mathrm{d}U_j} \dot{U}_j \Leftrightarrow \dot{U}_i \Leftrightarrow \sum_\alpha C_\alpha \frac{\partial C_\alpha}{\partial V_i} \right) + \sum_{i,\alpha} C_\alpha \frac{\partial C_\alpha}{\partial V_i} \dot{U}_i \\
&= \Leftrightarrow \sum_{i,j} \dot{U}_i a_{ij} \frac{\mathrm{d}V_j}{\mathrm{d}U_j} \dot{U}_j \Leftrightarrow \sum_i \dot{U}_i^2 \\
&= \Leftrightarrow \sum_{i,j} \dot{U}_i b_{ij} \dot{U}_j \leq 0. \qquad (5.13)
\end{aligned}$$

Provided $E_{\mathrm{kin+pot}}$ is bounded below (which is expected to hold in view of its definition), its value constantly decreases until finally $\forall i : \dot{U}_i = 0$. From (5.3) we see that this normally implies that $\forall \alpha : \dot{\lambda}_\alpha = 0$ too. We then conclude from equations (5.3) and (5.4) that a stationary point of the Langrangian function $E_{\mathrm{hl}}(V, \lambda)$ must have been reached under those circumstances. Or, in other words, a constrained equilibrium point of the neural network is attained. □

Inspection of the derivation reveals why the gradient ascent is helpful in (5.4): only when the sign flip is applied do the two terms $\sum_i \dot{U}_i \sum_\alpha C_\alpha \partial C_\alpha / \partial V_i$ cancel each other. In order to prove stability, we should analyze the complicated matrix (b_{ij}) which in full equals

$$b_{ij} = \left(\Leftrightarrow w_{ij} + \sum_\alpha \lambda_\alpha \frac{\partial^2 C_\alpha}{\partial V_i \partial V_j} \right) \frac{\mathrm{d}V_j}{\mathrm{d}U_j} + \delta_{ij}. \qquad (5.14)$$

Application of the Hopfield-Lagrange model to combinatorial optimization problems yields non-positive values for w_{ij}, so then $w'_{ij} \equiv \Leftrightarrow w_{ij} \geq 0$. If we confine ourselves to expressions C_α which are linear functions in V, then equation (5.14) reduces to

$$b_{ij} = w'_{ij} \frac{\mathrm{d}V_j}{\mathrm{d}U_j} + \delta_{ij}. \qquad (5.15)$$

If the δ_{ij}-terms dominate, then (b_{ij}) is positive definite and stability is sure. However, it seems impossible to formulate general conditions which guarantee stability, since the matrix elements b_{ij} are a function of $\mathrm{d}V_j/\mathrm{d}U_j$ and thus change dynamically during the update of the differential equations. This observation explains why we called this subsection 'A *potential* Lyapunov function'.

[2] Since $E_{\mathrm{kin+pot}}$ is the sum of kinetic and potential energy of the damped mass system, this function is a generalization of the Lyapunov function introduced by Platt and Barr [71]. They used equation (2.47) which has a simple quadratic potential energy term. Here, this term cannot be used because of the non-linear relationship $V_i = g(U_i)$. The quadratic term has to be modified in the integral as shown, while $\dot{V}_i$ is replaced by $\dot{U}_i$.

In practical applications, we can try to analyze matrix (b_{ij}). If this does not turn out successful, we may rely on experimental results. However, there is a way of escape, namely, by applying quadratic constraints. Under certain general conditions, they appear to guarantee stability in the long run at the cost of a degeneration of the Hopfield-Lagrange model to a type of penalty model.

5.2 Degeneration to a dynamic penalty model

5.2.1 Non-unique multipliers

We consider the Hopfield-Lagrange model as defined in the beginning of section 5.1.1.

Theorem 5.2. *Let W be the subspace of $[0,1]^n$ such that $V \in W \Rightarrow \forall\alpha : C_\alpha(V) = 0$ and let $V^0 \in W$. If the condition*

$$\forall\alpha, \forall i : C_\alpha = 0 \Rightarrow \frac{\partial C_\alpha}{\partial V_i} = 0 \tag{5.16}$$

holds, then there do not *exist unique numbers $\lambda_1^0, \ldots, \lambda_m^0$ such that $E_{\mathrm{hl}}(V,\lambda)$ has a critical point in (V^0, λ^0).*

Proof. The condition (5.16) implies that all $m \times m$ submatrices of the Jacobian (A.3) are singular. Conform the 'Lagrange Multiplier Theorem' of appendix A, uniqueness of the numbers $\lambda_1^0, \ldots, \lambda_m^0$ is not guaranteed. Moreover, in the critical point of E_{hl} the following equations hold:

$$\sum_j w_{ij} V_j^0 + I_i \Leftrightarrow \sum_\alpha \lambda_\alpha \frac{\partial C_\alpha}{\partial V_i}(V^0) \Leftrightarrow U_i = 0. \tag{5.17}$$

Since $\forall\alpha : \partial C_\alpha / \partial V_i(V^0) = 0$, the multipliers λ_α may have *arbitrary* values in a critical point of $E_{\mathrm{hl}}(V,\lambda)$. □

In the literature (e.g. in [85, 44, 82, 86, 88]) and in section 5.5 and 5.6 of this thesis, quadratic constraints are frequently encountered, often having the form

$$C_\alpha(V) = \tfrac{1}{2}(\sum_{i_\alpha} V_{i_\alpha} \Leftrightarrow n_\alpha)^2 = 0, \quad \alpha = 1 \cdots m, \tag{5.18}$$

where any n_α equals some constant. Commonly, the constraints relate to only a subset of all V_i. So, for a constraint C_α, the index i_α passes through some subset N_α of $\{1, 2, \ldots, n\}$. We conclude that

$$\frac{\partial C_\alpha}{\partial V_i} = \begin{cases} \sum_{i_\alpha} V_{i_\alpha} \Leftrightarrow n_\alpha & \text{if } i \in N_\alpha \\ 0 & \text{otherwise.} \end{cases} \tag{5.19}$$

It follows that condition (5.16) holds for the quadratic constraints (5.18). This implies that multipliers associated with those constraints are not uniquely determined in equilibrium points of the corresponding Hopfield-Lagrange model.

5.2.2 Stability yet

The question may arise how the Hopfield-Lagrange model deals with the non-determinacy of the multipliers[3]. To answer that question, we again consider (5.2), (5.3) and (5.4) and substitute the quadratic constraints (5.18). This yields

$$E_{\mathrm{hl,q}}(V,\lambda) = E(V) + \sum_{\alpha} \frac{\lambda_\alpha}{2} (\sum_{i_\alpha} V_{i_\alpha} \Leftrightarrow n_\alpha)^2 + E_{\mathrm{h}}(V), \tag{5.20}$$

$$\dot{U}_i = \Leftrightarrow \frac{\partial E}{\partial V_i} \Leftrightarrow \sum_{\alpha : i \in S_\alpha} \lambda_\alpha (\sum_{i_\alpha} V_{i_\alpha} \Leftrightarrow n_\alpha) \Leftrightarrow U_i, \tag{5.21}$$

$$\dot{\lambda}_\alpha = \tfrac{1}{2} (\sum_{i_\alpha} V_{i_\alpha} \Leftrightarrow n_\alpha)^2. \tag{5.22}$$

Theorem 5.3. *If* $\forall i : V_i = g(U_i)$ *is a differentiable and monotone increasing function, then the set of differential equations* (5.21) *and* (5.22) *is stable.*

Proof. We start by making the following crucial observations:

1. As long as a constraint is not fulfilled, it follows from (5.22) that the corresponding multiplier increases:

$$\dot{\lambda}_\alpha > 0. \tag{5.23}$$

2. If, at a certain moment, all constraint are fulfilled, then the set of motion equations (5.21) and (5.22) reduces to

$$\dot{U}_i = \Leftrightarrow \frac{\partial E}{\partial V_i} \Leftrightarrow U_i. \tag{5.24}$$

   Since we are dealing with the unconstrained Hopfield model, this system is stable provided the transfer function is differentiable and monotone increasing (chapter 3). This implies that instability of the system can *only* be caused by violation of one or more of the quadratic constraints.

We now consider the total energy $E_{\mathrm{hl,q}}$ of (5.20). Suppose that the system is initially unstable (if it would be stable, the set of differential equations would converge rapidly). One or more constraints must then be violated and the values of the corresponding multipliers will increase. If the instability endures, the multipliers will eventually become positive. It follows from (5.20) that the contribution

$$\sum_{\alpha} \frac{\lambda_\alpha}{2} (\sum_{i_\alpha} V_{i_\alpha} \Leftrightarrow n_\alpha)^2 \tag{5.25}$$

to $E_{\mathrm{hl,q}}$ then consists of only convex quadratic forms, which correspond to various parabolic 'pits' or 'troughs'[4] in the energy landscape of $E_{\mathrm{hl,q}}$. As long as the

[3] This must be in a certain positive way, since the aforementioned experiments from the literature were at least partially successful.

[4] If i_α passes through the whole set $\{1, 2, \ldots, n\}$, $(\sum_{i_\alpha} V_{i_\alpha} - n_\alpha)^2$ represents a n-dimensional parabolic pit. If, instead, i_α passes through a proper subset of $\{1, 2, \ldots, n\}$, this quadratic expression represents a trough in the energy landscape of $E_{\mathrm{hl,q}}$. However, in both cases, we shall speak of pits.

multipliers grow, the pits become steeper and steeper. Eventually, the quadratic terms will dominate and the system settles down in one of the created energy pits (whose location, we realize, is more or less influenced by E and E_{h}). In this way, the system will ultimately fulfill all constraints and will have become stable. □

Actually, for positive values of λ_α, the multiplier terms (5.25) fulfill the penalty term condition (2.37) and therefore act as penalty terms. Furthermore we notice that in case of applying a continuous neural network (where $\forall i : V_i \in [0,1]$), the minima of (5.25) might be boundary extrema.

As was sketched in the proof, the *system itself* always finds a feasible solution. This contrasts strongly with the traditional penalty approach, where the experimenter may need a lot of trials to determine appropriate penalty weights. Moreover, as sketched, the penalty terms might be 'as small as possible', having the additional advantage that the original cost function can be minimally distorted. Since the penalty weights change dynamically on their journey to equilibrium, we have met with what we shall term a *dynamic penalty method*.

5.2.3 A more general view on the degeneration

In the previous two subsections, we analyzed the degeneration of the Hopfield-Lagrange model under the specific condition (5.16) concerning the constraints. Here, a more general analysis of this deterioration to a dynamic penalty method is sketched, where the proof whether the multipliers are unique or not, does notbother us.

We consider the unconstrained Hopfield-Lagrange model as it was re-stated at the beginning of section 5.1.1. We already observed in that section that instability must be caused by the violation of one or more of the constraints provided the correct transfer function has been selected. We realize that if

1. $\forall\alpha, \forall V : C_\alpha(V) \geq 0$, and
2. increasing multiplier values correspond to a changing energy landscape with ever deeper pits whose minima represent valid solutions,

then the set of differential equations (5.3), (5.4) will generally be stable. The evidence for this phenomenon is based on the crucial observations (5.23) and (5.24), and is further discussed below.

It is interesting to note that the origin of ever deeper pits in the energy landscape resembles the phenomenon of a *phase transition* in a certain sense. If, in the unconstrained Hopfield model the temperature is increased, one steep pit is created by the entropy term (3.30). Above the critical temperature, the entropy term dominates and the corresponding solution equals $\forall i : V_i \approx 0.5$. In case of the degenerated Hopfield-Lagrange model, the pits originate by increasing multipliers (which behave like penalty weights). Above a certain set of critical values, the multiplier terms dominate and the various minima correspond to approximately feasible solutions.

In addition, mean field annealing can be applied. In that case, two transformations of the energy landscape occur simultaneously, one being caused by increasing multipliers, the other by a lowering of the temperature. We must take the adventitious consequence of a tuning problem concerning the absolute and relative speed of the two transformations.

5.3 Hard constraints

Let us return to our toy problem (5.5) and see how it works in practice. Using the Hopfield-Lagrange model, the differential equations corresponding to (5.6) are

$$\dot{U}_1 = \Leftrightarrow(2V_1 + \lambda_1) \Leftrightarrow U_1, \tag{5.26}$$

$$\dot{\lambda}_1 = V_1 \Leftrightarrow 1, \tag{5.27}$$

where $V_1 = g(U_1) = 1/(1 + \exp(\Leftrightarrow\beta U_1))$. We note that V_1 is now bounded to the interval $[0, 1]$. We can easily prove stability, since in this case

$$\dot{E}_{\mathrm{kin+pot,t}} = \Leftrightarrow 2\dot{U}_1^2 \frac{\mathrm{d}V_1}{\mathrm{d}U_1} \Leftrightarrow \dot{U}_1^2 \leq 0. \tag{5.28}$$

Consequently, $E_{\mathrm{kin+pot}}$ is monotone decreasing until $\dot{U}_1 = 0$ and thus, normally, until $\dot{\lambda}_1 = 0$, which in turn implies $V_1 = 1$ and $U_1 = \infty$. Inspection of (5.26) now reveals that in equilibrium, λ_1 must equal $\Leftrightarrow\infty$. So the critical point of $E_{\mathrm{hl,t}}$ is $(V_1, \lambda_1) = (1, \Leftrightarrow\infty)$ and we have run up against an unexpected difficulty. We have lost the pretty feature of the continuous Hopfield model of finding solutions corresponding to *finite* values of U_1. The reason is obvious: the 'hard' constraint $V_1 \Leftrightarrow 1$ restricts the solution space to $V_1 = 1$ with corresponding U_1-value equal to ∞.

There exists a simple solution for this problem 'in the spirit' of the continuous Hopfield model. If we relax the hard constraint (5.27) to

$$V_1 \Leftrightarrow 1 = \epsilon, \tag{5.29}$$

the new energy expression becomes

$$E_{\mathrm{hl,t'}} = V_1^2 + \lambda_1(V_1 \Leftrightarrow 1 + \epsilon) + \frac{1}{\beta}[(1 \Leftrightarrow V_1)\ln(1 \Leftrightarrow V_1) + V_1 \ln V_1], \tag{5.30}$$

having its critical point in $(V_1, \lambda_1) = (1 \Leftrightarrow\epsilon, \Leftrightarrow 2 + \Delta\lambda_1)$, where

$$\Delta\lambda_1 = 2\epsilon + \frac{1}{\beta}\ln(\frac{\epsilon}{1 \Leftrightarrow\epsilon}). \tag{5.31}$$

We see that the critical point is situated in the neighborhood of the original value if the error $\Delta\lambda_1$ (which is determined by ϵ and β) is small. Like in the original Hopfield model, it can be kept small if we choose large values of β. To determine the sensitivity of the parameters, we performed some calculations. We may conclude from the computational results as given in table 5.1 that sufficiently high values

$\epsilon = 0.001$		$\epsilon = 0.01$		$\epsilon = 0.1$	
β	$\Delta\lambda_1$	β	$\Delta\lambda_1$	β	$\Delta\lambda_1$
5000	$+0.0006$	5000	$+0.019$	5000	$+0.199$
500	$\Leftrightarrow 0.01$	500	$+0.010$	500	$+0.196$
50	$\Leftrightarrow 0.136$	50	$\Leftrightarrow 0.07$	50	$+0.156$
5	$\Leftrightarrow 1.38$	5	$\Leftrightarrow 0.90$	5	$\Leftrightarrow 0.239$
1	$\Leftrightarrow 6.9$	1	$\Leftrightarrow 4.58$	1	$\Leftrightarrow 1.997$

Table 5.1: The error $\Delta\lambda_1$ as a function of ϵ and β.

of β indeed guarantee a small error $\Delta\lambda_1$. In figure 5.2, some critical points have been put together. The position $(1, \Leftrightarrow 2)$ of the constrained minimum of $E_{\mathrm{pb,t}}$ is shown, together with some positions of the extrema of $E_{\mathrm{hl,t'}}$ for various values of β and $\epsilon = 0.01$. Clearly, the critical point $(V_1, \lambda_1) = (1, \Leftrightarrow\infty)$ of $E_{\mathrm{hl,t}}$ is absent in the figure.

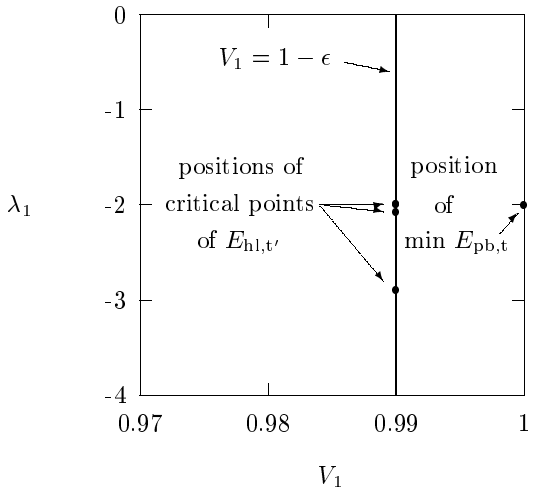


Figure 5.2: Positions of some critical points of $E_{\mathrm{hl,t'}}$ and of the minimum of $E_{\mathrm{pb,t}}$.

We note that the described difficulty of an infinite multiplier value only occurs if one constraint on its own, or several constraints together, are hard, by which we mean that the constraints extort that $\exists i : V_i = 0$ or $V_i = 1$. In practice, one often encounters constraints like

$$\sum_i V_i \Leftrightarrow 1 = 0. \tag{5.32}$$

If such a constraint stands alone or is independent of the other ones, the Hopfield term E_{h} generally drags the corresponding minima to the interior of the state space in the usual way as we have described in section 3.2.2.

5.4 Stability analysis, the constrained model

One could wonder whether a stability analysis is possible in case of combining the most general framework of chapter 4 – having 'arbitrary' cost functions and 'arbitrary' transfer functions – with the multiplier approach of this chapter. If so, this model can be used to build-in part of the constraints directly, while other ones are tackled using Lagrangian multipliers. In this subsection, the constraints $C_\alpha(V) = 0$ are assumed *only* to belong to the last category!

Let us take equation (4.41) of the most general framework as the starting point and then add multiplier terms to this expression. This yields the Lagrangian function

$$L(U, V, \lambda) = \Leftrightarrow H(V) + \sum_\alpha \lambda_\alpha C_\alpha(V) + \sum_i U_i V_i \Leftrightarrow G(U). \tag{5.33}$$

We want to determine the stationary points of $L(U, V, \lambda)$ since these points correspond to the solutions of the constrained optimization problem relating to this matter. It can be done by resolving the differential equations

$$\dot{U}_i = \Leftrightarrow \frac{\partial L}{\partial V_i} = \frac{\partial H}{\partial V_i} \Leftrightarrow \sum_\alpha \lambda_\alpha \frac{\partial C_\alpha}{\partial V_i} \Leftrightarrow U_i, \tag{5.34}$$

$$\dot{\lambda}_\alpha = +\frac{\partial L}{\partial \lambda_\alpha} = C_\alpha(V), \tag{5.35}$$

where, just like in equation (4.44), we keep permanently $V_i = g_i(U)$. We note that $\forall i : V_i = g_i(U)$ implies that $\forall i : \partial L / \partial U_i = 0$. Now, the following theorem can be proven which is a drastic generalization of theorem 5.1.

Theorem 5.4. *If the matrix* (d_{ij}) *defined by*

$$d_{ij} = \sum_k c_{ik} \frac{\partial V_k}{\partial U_j} + \delta_{ij}, \tag{5.36}$$

c_{ik} *being*

$$c_{ik} = \Leftrightarrow \frac{\partial^2 H}{\partial V_i \partial V_k} + \sum_\alpha \lambda_\alpha \frac{\partial^2 C_\alpha}{\partial V_i \partial V_k}, \tag{5.37}$$

first is or becomes and then remains positive definite, then the energy function (5.12) *is a Lyapunov function for the motion equations* (5.34) *and* (5.35)*, where* $\forall i : V_i = g_i(U)$.

Proof. In this case,

$$\ddot{U}_i = \Leftrightarrow \sum_j c_{ij} \sum_k \frac{\partial V_j}{\partial U_k} \dot{U}_k \Leftrightarrow \dot{U}_i \Leftrightarrow \sum_\alpha C_\alpha \frac{\partial C_\alpha}{\partial V_i}. \tag{5.38}$$

Taking the time derivative of (5.12), we obtain

$$\begin{aligned}
\dot{E}_{\text{kin+pot}} &= \sum_i \dot{U}_i \ddot{U}_i + \sum_{i,\alpha} C_\alpha \frac{\partial C_\alpha}{\partial V_i} \dot{U}_i \\
&= \sum_i \dot{U}_i \left(\Leftrightarrow \sum_j c_{ij} \sum_k \frac{\partial V_j}{\partial U_k} \dot{U}_k \Leftrightarrow \dot{U}_i \Leftrightarrow \sum_\alpha C_\alpha \frac{\partial C_\alpha}{\partial V_i} \right) + \sum_{i,\alpha} C_\alpha \frac{\partial C_\alpha}{\partial V_i} \dot{U}_i \\
&= \Leftrightarrow \sum_{i,j} \dot{U}_i \sum_k c_{ik} \frac{\partial V_k}{\partial U_j} \dot{U}_j \Leftrightarrow \sum_i \dot{U}_i^2 \\
&= \Leftrightarrow \sum_{i,j} \dot{U}_i d_{ij} \dot{U}_j \leq 0. \qquad (5.39)
\end{aligned}$$

The rest of the proof is analogous to the proof of theorem 5.1. In the end, we have $\forall i : \dot{U}_i = 0, \forall \alpha : \dot{\lambda}_\alpha = 0$, and $V_i = g_i(U)$, together implying that all partial derivatives of $L(U, V, \lambda)$ are zero. In other words, a constrained equilibrium point of the neural network has then been reached. □

The matrix (d_{ij}) is given in full by

$$d_{ij} = \sum_k \left(\Leftrightarrow \frac{\partial^2 H}{\partial V_i \partial V_k} + \sum_\alpha \lambda_\alpha \frac{\partial^2 C_\alpha}{\partial V_i \partial V_k} \right) \frac{\partial V_k}{\partial U_j} + \delta_{ij}. \qquad (5.40)$$

This matrix is even more complicated than matrix (b_{ij}), which was briefly analyzed in section 5.1.1. We must conclude that it will often be impossible to give an analytical proof of stability, implying that, in those cases, we should either rely on experimental results, or apply quadratic constraints.

We finish this theoretical part by observing that it seems also possible to select other updating rules for finding an equilibrium state of the general constrained Hopfield-Lagrange network (see theorem 4.10). However, these approaches have not been elaborated.

5.5 Computational results, the unconstrained model

5.5.1 Simple optimization problems

We started by performing some simple experiments by trying various quadratic cost functions with linear constraints. The general form equals

$$\begin{aligned}
&\text{minimize } E(V) = \tfrac{1}{2} \sum_{i=1}^{n} d_i (V_i \Leftrightarrow e_i)^2, \\
&\text{subject to : } a_i^\alpha V_i \Leftrightarrow b_i^\alpha = 0, \ \alpha = 1, .., m, \qquad (5.41)
\end{aligned}$$

where d_i is always chosen positive. The cost function is always such that its minimum belongs to the state space $[0, 1]^n$ and the constraints are non-contradictory.

Since for this class of problems

$$\frac{\partial^2 E}{\partial V_i \partial V_j} = d_i \delta_{ij} \quad \wedge \quad \frac{\partial^2 C_\alpha}{\partial V_i \partial V_j} = 0, \tag{5.42}$$

the corresponding time derivative of the sum of kinetic and potential energy equals

$$\dot{E}_{\mathrm{kin+pot,s}} = \Leftrightarrow \sum_{i=1}^{n} (d_i \frac{\mathrm{d}V_i}{\mathrm{d}U_i} + 1) \dot{U}_i^2 \leq 0. \tag{5.43}$$

Using the sigmoid as the transfer function, we expect convergence for all problem instances. All initializations of V_i, as well as of the multipliers, were chosen randomly. We started trying the 'toy problem' (5.5), which has been analyzed in section 5.3. Using $\Delta t = 0.0001$ and $\beta = 50$, U was still growing (to ∞) and λ was still shrinking (to -∞) after 10^7 iterations, which complies with the given theoretical conjectures. Cutting off the calculations, we found the 'final' values $V = 0.999959$ and $\lambda = \Leftrightarrow 2.404412$. Thereupon, we relaxed the constraint to $V \Leftrightarrow 1 = \epsilon$. Resolving the corresponding set of differential equations, choosing $\epsilon = 0.01$, and leaving the other parameters unchanged, we found asymptotic convergence to $V = 0.990000$ and $\lambda = \Leftrightarrow 2.163805$: the first value is the correct one and the second one approximates the theoretical value $\Leftrightarrow 2.07$ from table 5.1.

To investigate scalability, we extended the number of neurons and the number of constraints in formula (5.41). In all cases, we encountered proper convergence. E.g., taking

$$\text{minimize } V_1^2 + (V_2 \Leftrightarrow 1)^2 + V_3^2 + (V_4 \Leftrightarrow 1)^2 + \cdots + (V_{50} \Leftrightarrow 1)^2,$$

$$\text{subject to: } \begin{cases} V_1 + V_2 + \cdots + V_{10} = 5 \\ V_6 + V_7 + \cdots + V_{15} = 5 \\ V_{11} + V_{12} + \cdots + V_{20} = 5 \\ \vdots \quad\quad \vdots \quad\quad\quad\quad \vdots \quad\quad \vdots \\ V_{41} + V_{42} + \cdots + V_{50} = 5, \end{cases} \tag{5.44}$$

after 10^6 iterations with $\Delta t = 0.0001$ and $\beta = 50$, we found

$$\begin{aligned} \forall i : i \in \{1, 3, 5, \cdots, 49\} : V_i &= 0.056360 \\ \forall i : i \in \{2, 4, 6, \cdots, 50\} : V_i &= 0.943640, \end{aligned}$$

so, the constraints are exactly fulfilled. We also observe the expected effect of the Hopfield term. The values of the 9 multipliers λ_α all equal 0.000000, corresponding precisely to the theoretical ones, as can be easily verified. We repeated the experiment, now choosing $\beta = 100$. We found

$$\begin{aligned} \forall i : i \in \{1, 3, 5, \cdots, 49\} : V_i &= 0.033593 \\ \forall i : i \in \{2, 4, 6, \cdots, 50\} : V_i &= 0.966407. \end{aligned}$$

The influence of the Hopfield term has diminished, which also corresponds to the theoretical expectations.

5.5.2 The weighted matching problem

To investigate whether the Hopfield-Lagrange model is able to solve *combinatorial* optimization problems in an adequate way, we performed some other experiments. We first report the results of the computations concerning the WMP of section 2.2.2. Interpreting $V_{ij} = 1$ ($V_{ij} = 0$) as if point i is (not) linked to point j, where $1 \leq i < j \leq n$, we tried several formulations of the constraints. Using linear constraints, the corresponding system turned out to be *un*stable. Therefore, we continued by trying quadratic ones since then, stability is generally guaranteed, as was pointed out in section 5.2.3. The corresponding formulation of the problem is

$$\text{minimize } E(V) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} d_{ij} V_{ij},$$

subject to:

$$C_{1,i}(V) = \tfrac{1}{2}\Big(\sum_{j=1}^{i-1} V_{ji} + \sum_{j=i+1}^{n} V_{ij} \Leftrightarrow 1\Big)^2 = 0, \tag{5.45}$$

$$C_{2,ij}(V) = \tfrac{1}{2} V_{ij}(1 \Leftrightarrow V_{ij}) = 0. \tag{5.46}$$

The constraints (5.46) describe the requirement that finally, every V_{ij} must equal either 0 or 1. We note that every $C_{2,ij}$ corresponds to a concave function whose minima are boundary extrema. The corresponding multipliers are denoted by ν_{ij}. In combination with (5.46), the constraints (5.45) enforce that every point is linked

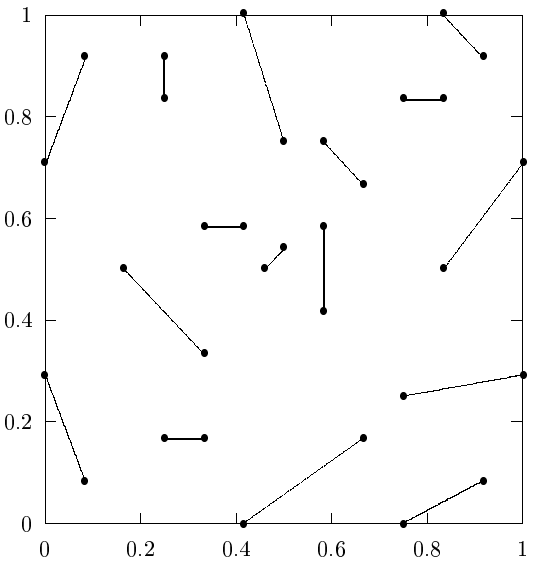


Figure 5.3: A solution of the WMP for $n = 32$.

to precisely one other point. The corresponding multipliers are λ_i, and the complete set of differential equations becomes

$$\begin{aligned}
\dot{U}_{ij} &= \Leftrightarrow d_{ij} \Leftrightarrow \lambda_i \Big(\sum_{k=1}^{i-1} V_{ki} + \sum_{k=i+1}^{n} V_{ik} \Leftrightarrow 1\Big) \Leftrightarrow \\
&\qquad \lambda_j \Big(\sum_{k=1}^{j-1} V_{kj} + \sum_{k=j+1}^{n} V_{jk} \Leftrightarrow 1\Big) \Leftrightarrow \nu_{ij}\big(\tfrac{1}{2} \Leftrightarrow V_{ij}\big) \Leftrightarrow U_{ij}, \qquad (5.47)\\
\dot{\lambda}_i &= \tfrac{1}{2}\Big(\sum_{j=1}^{i-1} V_{ji} + \sum_{j=i+1}^{n} V_{ij} \Leftrightarrow 1\Big)^2, \qquad (5.48)\\
\dot{\nu}_{ij} &= \tfrac{1}{2} V_{ij}(1 \Leftrightarrow V_{ij}). \qquad (5.49)
\end{aligned}$$

Again, the sigmoid was the selected transfer function. The multipliers were initialized with the value 0. The experiments showed proper convergence. Using 32 points, the corresponding system consists of 1024 differential equations and 528 multipliers. After 40 000 iterations using $\beta = 500$ and $\Delta t = 0.001$, the values of λ_i lay in the interval $[0.14; 0.83]$, while those of ν_{ij} were mostly of order 10^{-4} and sometimes of order 10^{-1}. The values of V_{ij} equalled 0.0000 or lay in the interval $[0.997; 1.000]$, which is interpreted as equal to 1. The corresponding solution is visualized in figure 5.3. We have repeated the experiment and always found solutions of similar quality, e.g., a solution where 13 (of the 16) links equal the links of the solution shown.

In order to show how difficult the stability analysis can be when using theorem 5.1, we determined the matrix (5.11) in case of $n = 4$. Enumerating rows and columns in the order (1,2), (1,3), (1,4), (2,3), (2,4), (3,4), we found:

$$(b^{\mathrm{wm}}_{ij,kl}) = \begin{pmatrix}
\Lambda_{12} & \lambda_1\Phi_{13} & \lambda_1\Phi_{14} & \lambda_2\Phi_{23} & \lambda_2\Phi_{24} & 0 \\
\lambda_1\Phi_{12} & \Lambda_{13} & \lambda_1\Phi_{14} & \lambda_3\Phi_{23} & 0 & \lambda_3\Phi_{34} \\
\lambda_1\Phi_{12} & \lambda_1\Phi_{13} & \Lambda_{14} & 0 & \lambda_4\Phi_{24} & \lambda_4\Phi_{34} \\
\lambda_2\Phi_{12} & \lambda_3\Phi_{13} & 0 & \Lambda_{23} & \lambda_2\Phi_{24} & \lambda_3\Phi_{34} \\
\lambda_2\Phi_{12} & 0 & \lambda_4\Phi_{14} & \lambda_2\Phi_{23} & \Lambda_{24} & \lambda_4\Phi_{34} \\
0 & \lambda_3\Phi_{13} & \lambda_4\Phi_{14} & \lambda_3\Phi_{23} & \lambda_4\Phi_{24} & \Lambda_{34}
\end{pmatrix}$$

where

$$\Lambda_{ij} = 1 + (\Leftrightarrow \nu_{ij} + \lambda_i + \lambda_j)\frac{\mathrm{d}V_{ij}}{\mathrm{d}U_{ij}} \quad \wedge \quad \Phi_{ij} = \frac{\mathrm{d}V_{ij}}{\mathrm{d}U_{ij}}. \qquad (5.50)$$

In general, we can not prove convergence because the properties of the matrix b^{wm} change dynamically. However, stability in the initial and final states can easily be demonstrated. Initially, we set all multipliers equal to 0. Then, b^{wm} reduces to the

unity matrix. On the other hand, if a feasible solution is found in the end, then $\forall i,j : V_{ij} \approx 0$ or $V_{ij} \approx 1$ implying that all $\Phi_{ij} \approx 0$. This again implies that b^{wm} reduces to the unity matrix. Since the unity matrix is positive definite, stability is guaranteed both at the start and in the end. However, during the updating process, the situation is much less clear. We have not further analyzed this theoretically.

5.5.3 The NRP and the TSP

To see whether the Hopfield-Lagrange model is useful for solving more difficult combinatorial optimization problems, we have tried to solve the TSP (section 2.2.2). We shall see that the NRP (section 3.4.1 and 4.4.5), itself being a purely combinatorial problem, is a special case of the combinatorial optimization TSP. We first consider a formulation of the TSP given by Hopfield and Tank [49]:

$$\text{minimize} \quad E_{\mathrm{tsp}}(V) = \sum_{i,j,k} V_{ij} d_{ik} V_{kj+1}, \tag{5.51}$$

subject to the constraints (3.53) to (3.55) of the NRP. Here, V_{ij} means that city i is visited in the j-th position, and d_{ij} represents the distance between city i and city j. Indices should be taken modulo n and it is supposed that $d_{ij} = d_{ji}$. Applying the Hopfield-Lagrange model, we search for the extrema of

$$E_{\mathrm{hl,u,tsp1}}(V,\lambda) = E_{\mathrm{tsp}}(V) + \sum_{\alpha=1}^{3} \lambda_\alpha C_\alpha(V) + E_{\mathrm{h}}(V). \tag{5.52}$$

The corresponding set of differential equations equals

$$\begin{aligned}
\dot{U}_{ij} &= \Leftrightarrow \sum_k d_{ik}(V_{kj+1} + V_{kj-1}) \Leftrightarrow \lambda_1 \sum_{k\neq j} V_{ik} \Leftrightarrow \\
&\qquad \lambda_2 \sum_{k\neq i} V_{kj} \Leftrightarrow \lambda_3 (\sum_{i,j} V_{ij} \Leftrightarrow n) \Leftrightarrow U_{ij}, && (5.53)\\
\dot{\lambda}_1 &= \sum_{i,j} \sum_{k>j} V_{ij} V_{ik}, && (5.54)\\
\dot{\lambda}_2 &= \sum_{j,i} \sum_{k>i} V_{ij} V_{kj}, && (5.55)\\
\dot{\lambda}_3 &= \tfrac{1}{2} (\sum_{i,j}^{n} V_{ij} \Leftrightarrow n)^2. && (5.56)
\end{aligned}$$

Comparing (5.52) to (3.56), we see that if $E_{\mathrm{tsp}}(V) = 0$, the TSP reduces to the NRP. It is clear that the applied constraints are quadratic. If $\forall \alpha : \lambda_\alpha > 0$, condition (2.37) holds for the multiplier terms, so in that case, they behave like penalty terms. We also note that $\forall i : \dot{\lambda}_i \geq 0$ and we therefore expect convergence of the set of differential equations.

The n-rook problem, again revisited

We already mentioned some computational results of the NRP using the soft approach (section 3.4.1) as well as the partially strong approach (section 4.4.5). Here, we want to test the Hopfield-Lagrange model for the same problem. We should apply the set of differential equations (5.53) to (5.56), where $\forall i, \forall k : d_{ik} = 0$.

Using random initializations of V_i, we found convergence provided that Δt is small enough. E.g., for $n = 25$, $\beta = 500$ and $\Delta t = 0.0001$ we found, after 2000 iterations, $\lambda_1 = 0.655935$, $\lambda_2 = 0.649828$, (a still growing multiplier) $\lambda_3 = 0.690099$, and an almost feasible solution. The increase of λ_3 can easily be explained by the theory of section 5.3 on hard constraints implying U_i-values equal to $+\infty$ or $\Leftrightarrow\infty$. We further note that all multipliers have become positive.

The Travelling Salesman Problem

Using the Hopfield-Lagrange model, the TSP can be grappled by searching the extrema of (5.52). In accordance with the observations as given in [86] (section 2.5), we found proper convergence to nearly feasible solutions, provided Δt was chosen small enough. Unfortunately, the quality of the solutions was very poor. Even problem instances of 4 cities did not yield optimal solutions every time. Trying instances with 32 cities yielded solutions like the bad one shown in figure 5.4.

Inspired by the success with the WMP, we tried to solve the TSP in a different way namely by taking other quadratic constraints with *one multiplier for every single constraint*. We expected to find better solutions, because in this approach many more multipliers are used, which should make the system more 'flexible'. The modified problem is to find an optimal extremum of

$$\begin{aligned} E_{\mathrm{hl,u,tsp2}}(V,\lambda) &= \sum_{i,j,k} V_{ij} d_{ik} V_{kj+1} + \sum_i \frac{\lambda_i}{2} (\sum_k V_{ik} \Leftrightarrow 1)^2 + \\ &\quad \sum_j \frac{\mu_j}{2} (\sum_k V_{kj} \Leftrightarrow 1)^2 + \sum_{i,j} \frac{\nu_{ij}}{2} V_{ij} (1 \Leftrightarrow V_{ij}). \end{aligned} \tag{5.57}$$

The corresponding set of differential equations equals

$$\begin{aligned} \dot{U}_{ij} &= \Leftrightarrow \sum_k d_{ik} (V_{kj+1} + V_{kj-1}) \Leftrightarrow \lambda_i (\sum_k V_{ik} \Leftrightarrow 1) \Leftrightarrow \\ &\quad \mu_j (\sum_k V_{kj} \Leftrightarrow 1) \Leftrightarrow \nu_{ij} (\tfrac{1}{2} \Leftrightarrow V_{ij}) \Leftrightarrow U_{ij}, \end{aligned} \tag{5.58}$$

$$\dot{\lambda}_i = \sum_i \tfrac{1}{2} (\sum_k V_{ik} \Leftrightarrow 1)^2, \tag{5.59}$$

$$\dot{\mu}_j = \sum_j \tfrac{1}{2} (\sum_k V_{kj} \Leftrightarrow 1)^2, \tag{5.60}$$

$$\dot{\nu}_{ij} = \sum_{i,j} \tfrac{1}{2} V_{ij} (1 \Leftrightarrow V_{ij}). \tag{5.61}$$

Again, the experiments showed proper convergence. For very small problem in-

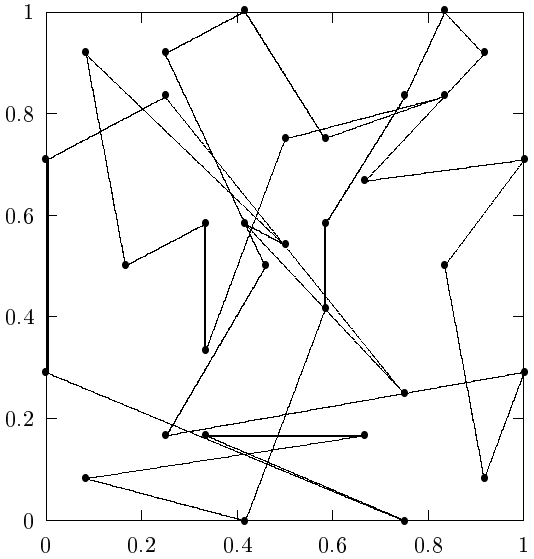


Figure 5.4: A solution of the TSP1 for $n = 32$.

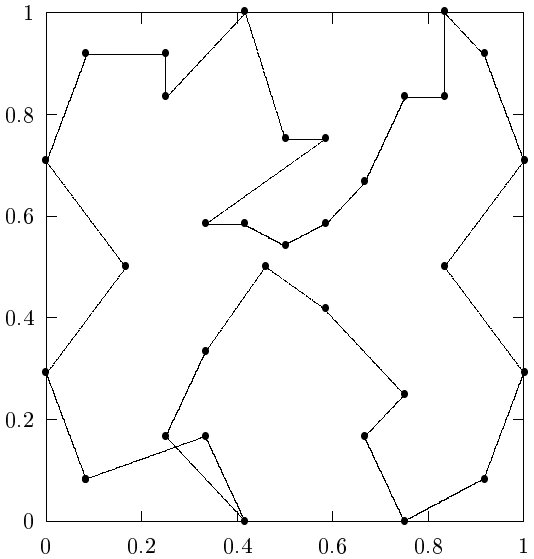


Figure 5.5: A solution of the TSP2 for $n = 32$.

stances we found optimal solutions. E.g., this time problem instances of 4 cities always yielded optimal solutions. Large problem instances also yielded feasible, but non-optimal solutions. An example is given in figure 5.5, where 32 cities were used, $\Delta t = 0.001$ and the applied number of iterations was 100 000. The encountered values of V_{ij} were either 0.0000 or lay in the interval [0.9988;1.0000]. The 1088 multipliers were still growing very slowly in order to realize exact fulfillment of the constraints, again owing to the problem with hard constraints. The quality of the solution is certainly better than the one we found in the previous subsection, although still not optimal. Apparently, the treatment of the constraints is now better, due to the use of much more multipliers. However, like in all other recurrent neural network approaches as known from literature, scalability appears to be a tough problem.

5.6 Computational results, the constrained model

It is interesting to experiment with combinations of the constrained Hopfield and the Hopfield-Lagrange model. Which part of the constraints is built-in and which part is tackled with multipliers, strongly depends on the structure of the problem. E.g., in case of the WMP (section 5.5.2), the constraints are highly interweaved and the constrained model is not at all applicable. Our approach will be the following one. Since building-in constraints has proven to be rather successful, we try to do this as much as possible. The remaining part of the constraints will be dealt with using multipliers.

5.6.1 For the last time, the n-rook problem

A constrained Hopfield-Lagrange formulation of the NRP resembles the formulation as given in section 4.4.5. The only difference concerns the penalty weight, which becomes a multiplier. We search an optimal extremum of the function

$$E_{\mathrm{hl,c,nr}}(V,\lambda) = \lambda_1 \sum_{j,i} \sum_{k>i} V_{ij} V_{kj} + E_{\mathrm{h}}(V), \tag{5.62}$$

using the motion equations

$$\dot{U}_{ij} = \Leftrightarrow\!\!\lambda_1 \sum_{k\neq i} V_{kj} \Leftrightarrow U_{ij}, \tag{5.63}$$

$$\dot{\lambda}_1 = \sum_{j,i} \sum_{k>i} V_{ij} V_{kj}, \tag{5.64}$$

where

$$V_{ij} = \frac{\exp(\beta U_{ij})}{\sum_l \exp(\beta U_{il})}. \tag{5.65}$$

The final multiplier value appears to depend on the initialization values of both the neurons and of λ_1. Moreover, for large problem instances, Δt should be chosen

n	$\lambda_{1,\mathrm{in}}$	Δt	$\lambda_{1,\mathrm{fin}}$
4	0	0.01	1.83
4	0	0.01	2.13
22	0	0.01	10.24
22	0	0.01	10.33
22	0	0.001	9.88
22	0	0.001	10.17
22	5	0.001	10.38
22	5	0.001	10.41
50	0	0.01	22.53
100	0	0.01	83.73
150	50	0.004	87.27
150	50	0.004	88.14

Table 5.2: Some initial and final multiplier values of the NRP.

small enough in order to avoid a too rapid increase of the multiplier value. For the rest, the experimental results are identical to those of section 4.4.5. We performed experiments up to $n = 150$. Using $\beta = 10$, we always encountered convergence. In the above-given table, we report (for certain problem instances) an appropriate value of Δt, the initial value $\lambda_{1,\mathrm{in}}$ and the final value $\lambda_{1,\mathrm{fin}}$ of multiplier λ_1. From the table we conclude that in case of $n = 22$, the critical value $\lambda_{1,\mathrm{cr}} \approx 10$. If $\lambda_1 < \lambda_{1,\mathrm{cr}}$, the set of motion equations appears to be unstable, yielding a constant increase of λ_1. As soon as its critical value has been reached, the system suddenly becomes stable and the constraints are rapidly fulfilled. This indeed resembles a phase transition which was conjectured in section 5.2.3.

5.6.2 The TSP

There are various ways to solve the TSP using the constrained model. The simplest approach concerns an adaptation of (5.62). It suffices to add the cost function $E_{\mathrm{tsp}}(V)$ as given by (5.51) to $E_{\mathrm{hl,c,nr}}(V, \lambda_1)$ and to adapt the corresponding motion equations. Again, the single multiplier appears to increase until a feasible solution is found. In this way, stability is always found. However, the quality of the solutions is rather poor. Even for $n = 4$, the solution found is not always the optimal one.

In a second approach, the constraint $C_2(V)$ is split into n separated ones, with a different multiplier for every one. Then, the problem is to find an optimal extremum of

$$E_{\mathrm{hl,c,tsp2}}(V, \lambda) = \sum_{i,j,k} V_{ij} d_{ik} V_{kj+1} + \sum_j \lambda_j \sum_i \sum_{k>i} V_{ij} V_{kj} + E_{\mathrm{h}}(V). \qquad (5.66)$$

The corresponding set of motion equations consists of a straightforward adaptation of the set (5.63) and (5.64). Unfortunately, the quality of the solutions remains poor. Even in this case, the solution found for $n = 4$ is not always the optimal one.

Thereupon, it is tried to resolve a constrained version of the approach of section 5.5.3 using other and much more quadratic constraints. The set of differential equations applied is

$$\begin{aligned}
\dot{U}_{ij} &= \Leftrightarrow\sum_k d_{ik}(V_{kj+1} + V_{kj-1}) \Leftrightarrow \mu_j(\sum_k V_{kj} \Leftrightarrow 1) \Leftrightarrow \\
&\qquad \nu_{ij}(\tfrac{1}{2} \Leftrightarrow V_{ij}) \Leftrightarrow U_{ij}, && (5.67)\\
\dot{\mu}_j &= \sum_j \tfrac{1}{2}(\sum_k V_{kj} \Leftrightarrow 1)^2, && (5.68)\\
\dot{\nu}_{ij} &= \sum_{i,j} \tfrac{1}{2} V_{ij}(1 \Leftrightarrow V_{ij}), && (5.69)
\end{aligned}$$

where the transfer function is (5.65). This set of equations correspond to the energy function defined by

$$\begin{aligned}
E_{\mathrm{hl,c,tsp3}}(V,\lambda) &= \sum_{i,j,k} V_{ij} d_{ik} V_{kj+1} + \sum_j \frac{\mu_j}{2}(\sum_k V_{kj} \Leftrightarrow 1)^2 + \\
&\qquad \sum_{i,j} \frac{\nu_{ij}}{2} V_{ij}(1 \Leftrightarrow V_{ij}) + E_{\mathrm{h}}(V). && (5.70)
\end{aligned}$$

Using $n = 4$, the optimal solution is always found, again proving the expected flexibility of the system. Trying a problem instance with $n = 15$, we encountered the optimal solution at times, but also a slightly worse one occasionally. Finally, trying an instance with $n = 32$, we did not find the optimal one. The quality is even worse than in case of using the unconstrained Hopfield-Lagrange model discussed in the previous section. Scalability again turns out to be a difficult issue.

Two things can still be tried. First, other mappings of the TSP on the Hopfield-Lagrange model using other cost functions (e.g., those having higher order terms [24, 80]) can be investigated. A second thing to do is to apply the technique of (mean field) annealing. These experiments have yet to be done.

Chapter 6

Elastic networks

We dwell upon Simic's claim that statistical mechanics is the underlying theory of both 'neural' and 'elastic' optimizations. We shall explain why we think his derivation is incorrect. In our view, the elastic net algorithm (ENA) as sketched in section 2.6 should be considered as a specific dynamic penalty method. We next give an analysis of the ENA by considering elastic net forces as well as various energy landscapes. This analysis further underpins our view. Finally, we formulate two alternative elastic net algorithms and report some computational results.

Parts of this chapter will soon be published [19]. A substantial part can be found in the technical reports [16, 18].

6.1 The ENA is a dynamic penalty method

In his analysis of the relationship between neural and elastic networks [77], Simic applies the stochastic binary constrained Hopfield model of chapter 4. The motivation for using a stochastic model is based upon the idea of considering stochastic 'particle trajectories'. Using the customary statistical mechanical arguments, the particle should spontaneously find the path of minimal length (the path length is the Hamiltonian of the problem). Like has been explained in chapter 2, this phenomenon can also be described by a minimization process of the corresponding free energy.

Stated more precisely, a 'statistical mechanics' is defined regarding particle trajectories as an ensemble, where the paths of legal trajectories must obey the global constraints of the TSP. That is, the particle cannot visit two space-points at the same time and it visits all the points once and only once. The legal trajectory with the shortest path length equals the wanted shortest path and coincides with the shortest tour of the travelling salesman. The chosen representation of the tour length is the Hamiltonian

$$H_{\mathrm{tsp}}(S) = \tfrac{1}{4}\sum_i \sum_{p,q} d_{pq}^2 S_p^i (S_q^{i+1} + S_q^{i-1}) + \tfrac{\alpha}{4}\sum_i \sum_{p,q} d_{pq}^2 S_p^i S_q^i, \qquad (6.1)$$

where S_p^i denotes whether the salesman at time i occupies space-point p or not ($S_p^i = 1$ or $S_p^i = 0$), and where d_{pq} represents the distance between the space points p and q. The first term of (6.1) equals the sum of distance-squares between cities visited, while the second term is a penalty term which penalizes the simultaneous presence of the salesman at more than one position. Other constraints should guarantee that any city is visited once and only once. They are built-in in the strong way by imposing

$$\forall p : \sum_i S_p^i = 1. \tag{6.2}$$

A mean field approximation of the free energy of general stochastic Hopfield networks submitted to the constraints (6.2) can easily be found by applying theorem 4.1 n times. Also taking $I_i = 0$ and replacing w_{ij} by $\Leftrightarrow w_{ij}$ (see the note at the end of section 4.1), one finds [77]

$$F_{\mathrm{c1,g}}(V) = \Leftrightarrow \tfrac{1}{2} \sum_{i,j} \sum_{p,q} w_{pq}^{ij} V_p^i V_q^j \Leftrightarrow \tfrac{1}{\beta} \sum_p \ln \big[\sum_i \exp(\Leftrightarrow \beta \sum_{j,q} w_{pq}^{ij} V_q^j) \big], \tag{6.3}$$

where the stationary points of $F_{\mathrm{c1,g}}(V)$ equal the solutions of

$$V_p^i = \frac{\exp(\Leftrightarrow \beta \sum_{j,q} w_{pq}^{ij} V_q^j)}{\sum_l \exp(\Leftrightarrow \beta \sum_{j,q} w_{pq}^{lj} V_q^j)}. \tag{6.4}$$

Substituting the cost function (6.1) in (6.3) yields the actual free energy approximation of the TSP, being

$$\begin{aligned} F_{\mathrm{tsp}}(V) \;=\; & \Leftrightarrow \tfrac{1}{4} \sum_i \sum_{p,q} d_{pq}^2 V_p^i (V_q^{i+1} + V_q^{i-1}) \Leftrightarrow \tfrac{\alpha}{4} \sum_i \sum_{p,q} d_{pq}^2 V_p^i V_q^i \Leftrightarrow \\ & \tfrac{1}{\beta} \sum_p \ln \big[\sum_i \exp(\Leftrightarrow \tfrac{\beta}{2} \sum_q d_{pq}^2 (\alpha V_q^i + V_q^{i+1} + V_q^{i-1})) \big]. \end{aligned} \tag{6.5}$$

It is interesting to note Simic's observation that expression (6.3) has the 'wrong' sign. The structure of the equation indeed suggests that its stationary points correspond to *maxima* (compare the results of the sections 3.2 and 4.2), while those of the ENA are *minima.* Especially this phenomenon aroused our suspicions regarding his derivation. From here, we continue to sketch Simic's derivation, eventually resulting into the ENA. At the same time, we shall formulate our objections.

Objection 1. In order to derive a free energy expression in the standard form (2.8), Simic applies a Taylor series expansion on the last term of (6.5). We shall do the same. We first define

$$f(x) \;=\; \sum_p \ln \big[\sum_i \exp(x_p^i) \big], \tag{6.6}$$

$$a_p^i \;=\; \Leftrightarrow \beta \tfrac{\alpha}{2} \sum_q d_{pq}^2 V_q^i, \quad \text{and} \tag{6.7}$$

$$h_p^i \;=\; \Leftrightarrow \beta \tfrac{1}{2} \sum_q d_{pq}^2 (V_q^{i+1} + V_q^{i-1}), \tag{6.8}$$

implying that

$$\frac{\partial f}{\partial x_p^i}(a_p^i) = \frac{\exp(a_p^i)}{\sum_l \exp(a_p^l)}. \tag{6.9}$$

For the TSP, the mean field equations (6.4) can be written as

$$V_p^i = \frac{\exp(a_p^i + h_p^i)}{\sum_l \exp(a_p^l + h_p^l)} \approx \frac{\exp(a_p^i)}{\sum_l \exp(a_p^l)}, \tag{6.10}$$

provided that $\mid h_p^i \mid \ll \mid a_p^i \mid$ (this can be arranged by setting $\alpha \gg 1$). Now, combining (6.6), (6.7), (6.8), (6.9), and (6.10), we obtain

$$\begin{aligned} f(a+h) &= \sum_p \ln\Big[\sum_i \exp(a_p^i)\Big] + \sum_{i,p} h_p^i \frac{\partial f}{\partial x_p^i}(a_p^i) + \mathcal{O}(h^2) && (6.11)\\ &\approx \sum_p \ln \sum_i \exp\big(\Leftrightarrow\beta\tfrac{\alpha}{2}\sum_q d_{pq}^2 V_q^i\big) \Leftrightarrow \\ &\qquad \tfrac{\beta}{2}\sum_i \sum_{p,q} d_{pq}^2 V_p^i (V_q^{i+1} + V_q^{i-1}). && (6.12) \end{aligned}$$

Substitution of this result in (6.5) yields

$$\begin{aligned} F_{\mathrm{tsp,app}}(V) &= \tfrac{1}{4}\sum_i \sum_{p,q} d_{pq}^2 V_p^i (V_q^{i+1} + V_q^{i-1}) \Leftrightarrow \tfrac{\alpha}{4}\sum_i \sum_{p,q} d_{pq}^2 V_p^i V_q^i \Leftrightarrow \\ &\qquad \tfrac{1}{\beta}\sum_p \ln \sum_i \exp\big(\Leftrightarrow\beta\tfrac{\alpha}{2}\sum_q d_{pq}^2 V_q^i\big). && (6.13) \end{aligned}$$

Simic found a slightly different expression with the weight value $\frac{\alpha}{2}$ instead of the value $\Leftrightarrow\frac{\alpha}{4}$. He simply ignores this term by saying that it vanishes if the constraints are obeyed. Doing the same (although it is in itself dubious), the following expression of the free energy is obtained:

$$\begin{aligned} F_{\mathrm{tsp,sim}}(V) &= \tfrac{1}{4}\sum_i \sum_{p,q} d_{pq}^2 V_p^i (V_q^{i+1} + V_q^{i-1}) \Leftrightarrow \\ &\qquad \tfrac{1}{\beta}\sum_p \ln \sum_i \exp\big(\Leftrightarrow\beta\tfrac{\alpha}{2}\sum_q d_{pq}^2 V_q^i\big). && (6.14) \end{aligned}$$

However, inspection of equation (6.11) reveals that the chosen first-order Taylor-approximation does not hold for low values of the temperature, i.e., for high values of β, since h_p^i as defined in (6.8) is proportional to β. This observation concerns a fundamental objection since, during the execution of the ENA, the parameter β is increased step by step until it has reached a relatively high value in the end. □

Objection 2. In order to transform the Hopfield network formulation of the TSP into the elastic net one, Simic performs a 'decomposition of the particle trajectory':

$$x^i = <x(i)> = \sum_p x_p <S_p^i> = \sum_p x_p V_p^i. \tag{6.15}$$

Here, $x(i)$ is the (stochastic) position of the particle at time i, x_p is the vector denoting the position of city point p, and x^i denotes the *average* (or expected) position of the particle at time i. Using the decomposition, he writes

$$\tfrac{1}{4}\sum_i \sum_{p,q} d_{pq}^2 V_p^i (V_q^{i+1} + V_q^{i-1}) = \tfrac{1}{2}\sum_i \mid x^{i+1} \Leftrightarrow x^i \mid^2, \tag{6.16}$$

which is correct, and

$$\sum_q d_{pq}^2 V_q^i = \mid x_p \Leftrightarrow x^i \mid^2 . \tag{6.17}$$

The last equation concerns both a notable and a crucial transformation from a *linear* function in V_p^i into a *quadratic* one in x^i. Using (6.16) and (6.17), the free energy (2.54) of the ENA with $m = n$ and $\alpha_1 = \alpha_2 = 1$ is obtained. For reasons of convenience, we here restate that free energy expression:

$$E_{\mathrm{en}}(x) = \tfrac{\alpha_2}{2}\sum_{i=1}^{m} \mid x^{i+1} \Leftrightarrow x^i \mid^2 \Leftrightarrow \tfrac{\alpha_1}{\beta}\sum_{p=1}^{n} \ln \sum_{j=1}^{m} \exp(\tfrac{-\beta^2}{2} \mid x_p \Leftrightarrow x^j \mid^2). \tag{6.18}$$

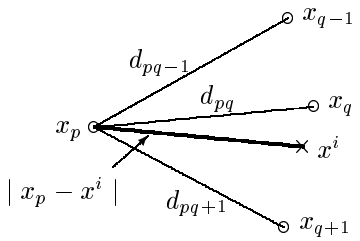


Figure 6.1: An elucidation of the inequality in (6.19).

However, careful analysis shows that in general

$$\sum_q d_{pq}^2 V_q^i = \sum_q (x_p \Leftrightarrow x_q)^2 V_q^i \neq \mid x_p \Leftrightarrow x^i \mid^2 . \tag{6.19}$$

The left-hand side of this inequality represents the expected sum of the distance squares between city point p and the particle position at time i, while the right-hand side represents the square of the distance between city point p and the expected particle position at time i. Under special conditions (e.g., if the constraints are fulfilled), the inequality sign must be replaced by the equality sign, but in general, the inequality holds (see also figure 6.1). □

Objection 3. The free energy expressions (6.5) and (6.18) appear to have very different properties. As can be concluded from (6.4), any of the free energy expressions (6.3) and (6.5) has the peculiar property that – whatever the value of the temperature parameter – the stationary points are found at states where, on average,

all strongly submitted constraints are automatically fulfilled. In other words, the stationary points by themselves meet the constraints

$$\forall p : \sum_i V_p^i = 1, \tag{6.20}$$

which signify that, on average, every city p is visited once. Moreover, the stationary points of (6.3) are often maxima (compare the results of chapter 4).

However, inspection of the free energy (6.18) yields a very different view: an analysis of that expression (see below) clarifies that each term *on its own* creates a set of local minima, the first one trying to minimize the tour length, the second one trying to force a valid solution. The current value of the temperature, which is a weight factor of the second term, determines the overall effect of summation over all these minima. E.g., it determines which of the two types will dominate. Thus, a competition between feasibility and optimality takes place. This phenomenon is remarkable, since the competition is similar to the one found by applying the classical penalty method. A difference from that classical method is that in the present case – as in case of the Hopfield-Lagrange model – the weights of the penalty terms change dynamically. It is surprising to see that in case of the ENA, the weights (which all equal $T = 1/\beta$) *de*crease during the updating of the motion equations, while in case of the Hopfield-Lagrange model, the weights (the multipliers) often *in*crease. The given view on the ENA explains why we consider it a *dynamic penalty method*. □

We think the last observation corresponds to the theory of so-called deformable templates [70, 90]. In that approach, the elastic net is considered as a 'template trajectory' (corresponding to Simic's particle trajectory), whose correct parameters should be determined. These parameters are the 'template coordinates' (the elastic net points) and the binary Potts spins S_{pj} (where $\forall p : \sum_j S_{pj} = 1$). We note that $S_{pj} = 1$ has the meaning that net point j is assigned to template coordinate p. The corresponding Hamiltonian equals

$$E_{\mathrm{dt}}(S, x) = \tfrac{\alpha_2}{2} \sum_i \mid x^{i+1} \Leftrightarrow x^i \mid^2 + \sum_{p,j} S_{pj} \mid x_p \Leftrightarrow x^j \mid^2 . \tag{6.21}$$

Thus, the energy E_{dt} is a function of both binary decision functions S_{pj} and of continuous template coordinates x^i. The first term in (6.21) equals the first term in the elastic net energy expression (6.18) and minimizes the tour length. The second term enforces a match between each city and one of the elastic net points. In other words, the energy (6.21) describes a penalty method. A statistical analysis of E_{dt} using the fact that the binary spins S_{pj} are stochastic, yields the free energy expression (6.18) of the elastic net. The derivation is straightforward [70, 90], among other things because E_{dt} is a linear function in the Potts spins. By inspection of both (6.18) and (6.21) we conclude that the first energy expression is derived from the second by adding stochastic noise exclusively to the penalty terms of (6.21). Therefore, one might say that the deformable template method applies stochastic penalty terms, whose noise level depends on the current value of the decreasing

temperature. This underpins, in yet another way, the idea that the ENA is based on a dynamic penalty model: the elastic net model can considered to be a *thermal* or *noisy penalty model*, where the current temperature (i.e., the current noise level) controls the actual form and weight of the penalty terms.

6.2 Energy landscapes

6.2.1 Energy landscapes and elastic net forces

The ENA can be analyzed at two levels, namely at the level of the energy equation (6.18) by inspection of the energy surface, and at the level of the updating rule (2.55) being

$$\Delta x^i = \frac{\alpha_2}{\beta}(x^{i+1} \Leftrightarrow 2x^i + x^{i-1}) + \alpha_1 \sum_p \Lambda^p(i)(x_p \Leftrightarrow x^i), \tag{6.22}$$

by an analysis of the various forces acting upon every net point. Afterwards, we shall deal – in a direct mathematical way – with the properties of the energy equation on lowering the temperature. We adopt the parameter values of the algorithm as given in subsection 2.2.

Let us start regarding the first, so-called elastic ring term (ert) of (6.18). It is composed of a sum of m (the number of elastic net points) quadratic position differences. Of course, this term is minimized if all points coincide at some place. However, if the elastic net has a given length, this term is minimized whenever all space-points are equidistant. In figure 6.2 and 6.3, the 2-dimensional energy

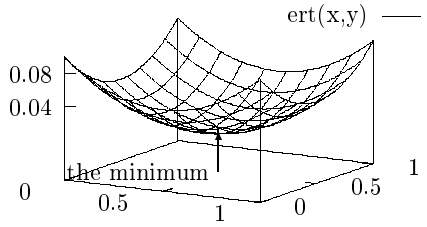


Figure 6.2: The elastic ring term for point (0.5,0.5), $d = 0.02$.

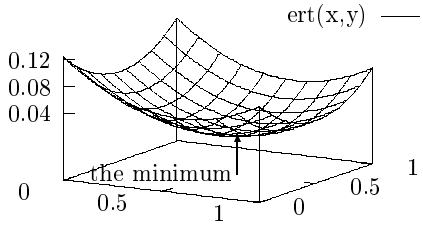


Figure 6.3: The elastic ring term for point (0.6,0.5), $d = 0.2$.

landscape of one net point $x^i = (x, y)$ is shown at two different positions, once using $d = 0.02$ as the mutual distance between neighbouring net points, the other time taking $d = 0.2$. The shapes of the two landscapes do not differ much: in both cases, the variable point is forced to the middle of the other two (temporally fixed) points: in figure 6.2, these points are (0.49,0.5) and (0.51,0.5), in figure 6.3 they are (0.5,0.5) and (0.7,0.5). At the level of the motion equation (6.22), we see by writing

$$x^{i+1} \Leftrightarrow 2x^i + x^{i-1} = (x^{i+1} \Leftrightarrow x^i) + (x^{i-1} \Leftrightarrow x^i), \tag{6.23}$$

that every x^i is forced to the *midpoint* between x^{i-1} and x^{i+1}. Summarizing, if the elastic ring term would be the only one, the ring points would become equidistant and, eventually, would coincide at one position, somewhere in state space.

But the second so-called mapping term (mpt) of (6.18), makes its influence felt too. It is composed of a sum of n logarithms, each logarithm having a sum of m exponentials as its argument. Every exponential is a Gaussian function with one local extremum, namely at the position where x_p coincides with x^j. We may conclude, that the total mapping term (with the minus sign) corresponds to a set of 'pits' in the energy landscape. The width and depth of these pits depend on two factors, namely on the temperature and on the distance between a city and the other elastic net points, especially the nearest elastic net point. Initially, when the temperature T is relatively high, the attraction of elastic net points by every city is more or less uniformly distributed. This corresponds to a wide and shallow pit in the energy landscape around every city. The resulting, total energy land-

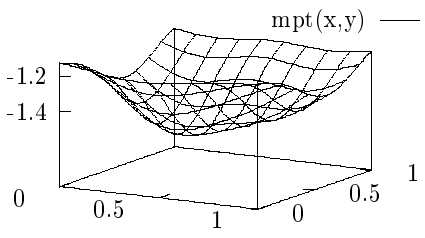


Figure 6.4: The mapping term, initially at high temperature.

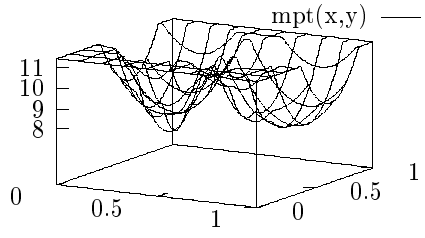


Figure 6.5: The mapping term, in case of no feasibility.

scape shelves slightly and is lowest in regions with a high city density. This phenomenon is quite independent of the position of the elastic net points in the unit square. A simple example is given in figure 6.4: again, the energy landscape of one of five elastic net points is shown, while the positions of the city points are (0.2,0.63), (0.8,0.63), (0.65,0.37), (0.37,0.37) and (0.33,0.37). The city positions will be kept the same in the next examples and can be found in figure 6.7. As can be seen in figure 6.4, the lowest part of the energy landscape of the mapping term is found around the last two, closely situated, cities. Experiments show, that the positions of the other four elastic net points do not matter much, i.e., whatever these positions are, in all cases approximately the same energy surface is found, provided that the initially high temperature $T = 0.2$ is used.

On lowering the temperature, a city will attract more and more nearby net points and fewer and fewer distant net points because, in general, the pit in the energy landscape around a city becomes narrower. However, the second factor plays an important part. If a city remains without a nearby elastic net point, the width of the pit shrinks only slowly and the depth even grows: apparently, the city persists in trying to catch a not too remote elastic net point. In figure 6.5, an example is given at $T = 0.027$, which is an almost final temperature of the algo-

rithm. The four net points are still chosen around the center of the unit square, far away from any city. The basins of attraction around every city are clearly present.

If, on the other hand, a city has been able to (almost) catch a net point, the surrounding pit in the energy landscape will become very narrow and shallow. In figure 6.6, an example is given with, once more, four temporally fixed net points. Again, $T = 0.027$. The position of one net point coincides exactly with a city, the position of a second one is chosen close to a city, a third net point is situated on a somewhat larger distance from another city, and the position of the fourth net point is precisely in the middle between two close city points. The city point and net point positions are shown in figure 6.7. The energy landscape in figure 6.6 shows narrow and shallow pits around cities: the smaller the distance of the most neighbouring elastic net point is, the narrower and shallower the pit. The figure also demonstrates an unpleasant phenomenon concerning the elastic net point in the middle of the two close cities. Both cities seem to consider themselves owner of that elastic net point. Consequently, the surrounding energy landscape of the two cities will generally not be able to catch another elastic net point, so, in those circumstances, the system persists in non-feasibility.

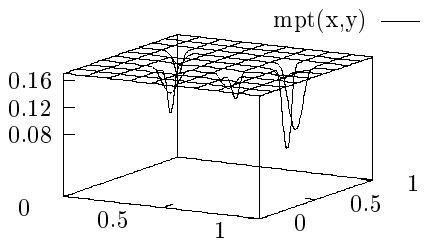


Figure 6.6: The mapping term, in case of an almost feasible solution.

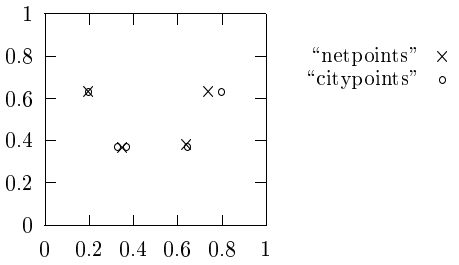


Figure 6.7: Net and city point positions.

6.2.2 The total energy landscape

Of course, we should analyze the combined effect of the elastic ring and the mapping term. For that purpose, we selected some, more or less representative examples, starting with an initial elastic net situated around the center of the unit square at $T = 0.2$. There, the energy landscape appears to resemble that of figure 6.4 (as expected): the mapping term dominates, pushing the elastic net to regions of high city density. In practice, the cities are distributed over the unit square, resulting in a stretching out of the net. In the background, the elastic net term keeps the net more or less together. On lowering the temperature a little bit until $T = 0.15$, the mapping term becomes more important as long as feasibility has not been reached. In figure 6.8, the energy landscape of the free elastic net point is

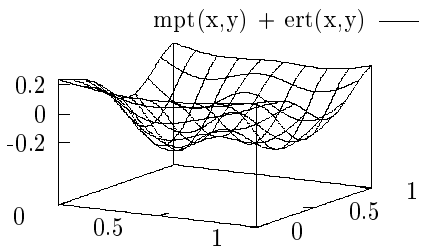


Figure 6.8: The total energy landscape, an initial state at $T = 0.15$.

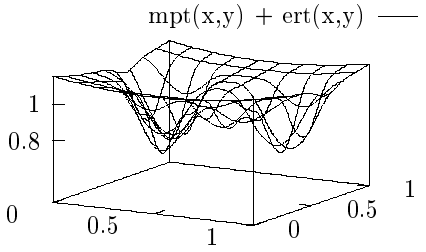


Figure 6.9: The total energy landscape, an intermediate state at $T = 0.08$.

shown under the assumption that the initial configuration of all other points have remained the same. It is clear that the landscape has become somewhat steeper. Thus, the system is trying to reach feasibility with a bit more strength. Now supposing the more realistic scenario that the elastic net has stretched out somewhat (with elastic net positions (0.57,0.44), (0.43,0.44), (0.35,0.56), (0.65,0.56), while the 'free' elastic net point is supposed to be somewhere between the last two given positions). Then, more details in the energy landscape are apparent. In figure 6.9, the energy landscape is shown at $T = 0.08$.

Next, we show two potential, nearly final, states. In figure 6.10, a solution is

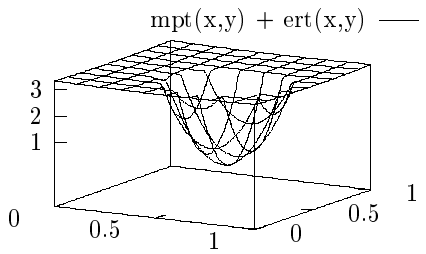


Figure 6.10: The total energy landscape, a non-feasible state in the end.

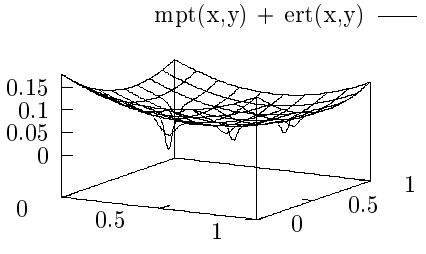


Figure 6.11: The total energy landscape, a feasible state in the end.

shown, where all cities except one have caught an elastic net point. If the remaining net point is not too far away from the non-visited city, it can still be attracted by it, otherwise this city will never be visited. This shows, that a too rapid lowering of the temperature may lead to a non-valid solution, because a further lowering of the temperature will lead to a further narrowing of the energy pit of figure 6.10. Note also that in this case, the pits corresponding to the elastic ring term are not visible: comparatively, they are too small. In figure 6.11, an almost feasible solu-

tion is shown, where the positions of three net points coincide with the position of a city, while a fourth elastic net point is precisely in the middle between the two close cities. Because an almost feasible solution has been reached, the mapping term becomes relatively small (corresponding to some small pits), and the remaining elastic net point is forced to the middle of its neighbors. The final state will be equidistant, but not feasible! The example shows clearly that in case of (almost) feasibility the influence of the mapping term becomes small. At the same time, this term is capable to maintain this (almost) feasibility. Under these conditions, the algorithm tries to realize equidistance.

6.2.3 Non-feasibility and not annealing

The analysis of the previous subsection reveals that it is possible to end up in a non-feasible solution for at least two reasons[1]:

- The parameter T may be lowered too rapidly yielding a non-feasible solution, where one or more cities have not 'caught' any elastic point.

- Two close cities may have received the same elastic net point as the nearest one.

The determination of the optimal schedule for decreasing T is often mentioned in literature and is often associated with 'optimal simulated annealing'. We wish to emphasize here, that the similarity is less than would appear. In simulated annealing [2], the temperature should be decreased carefully in order to *escape* from local minima. Here, this lowering should be done carefully in order to gain and keep on to a valid solution, in other words, to *end up* in a local (constrained) minimum!

Just like any other penalty method, the ENA tries to fulfill two competing requirements: in this case these are *minimal equidistance* and *feasibility* (a tour through all city points). To be able to fulfill both requirements, it is generally necessary to use more elastic net points than city points. This is further explained in figure 6.12. It should be clear that, the more diversity exists in the shortest distances between cities, the more elastic net points are needed [2]. Using a large number of elastic net points gives rise to the additional drawback of increasing computation time. Finally, we note that the property of equidistance – which is a consequence of the quadratic distance measure of the ENA – is not at all a necessary qualification of the final solution. The above-mentioned observations that (a) a non-feasible solution might be found and (b) the ENA pursues equidistance, motivated us to investigate alternative elastic net algorithms.

[1] Another non-feasibility is a so-called spike [78]. There, one city has caught two non-neighbouring beads of the elastic ring.

[2] We already mentioned in section 2.6 that usually m and n are chosen conform $m = 2.5\,n$.

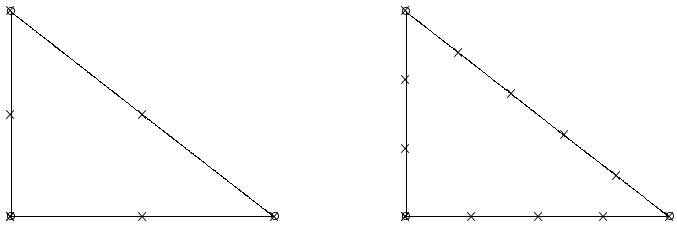


Figure 6.12: To realize both feasibility and equidistance many net points are needed.

6.3 Alternative elastic networks

6.3.1 A non-equidistant elastic net algorithm

In order to get rid of the equidistance property, we only need to change the first term of the original energy expression (6.18). Here, a *linear* distance function is chosen[3], whose minimal constrained length equals by definition the global minimal tour length. The new energy function is:

$$E_{\mathrm{lin}}(x) = \alpha_2 \sum_i \mid x^{i+1} \Leftrightarrow x^i \mid \Leftrightarrow \frac{\alpha_1}{\beta} \sum_p \ln \sum_j \exp(\frac{-\beta^2}{2} \mid x_p \Leftrightarrow x^j \mid^2). \qquad (6.24)$$

Applying gradient descent, the corresponding motion equations are found [35]:

$$\Delta x^i = \frac{\alpha_2}{\beta} \left(\frac{x^{i+1} \Leftrightarrow x^i}{\mid x^{i+1} \Leftrightarrow x^i \mid} \right) + \left(\frac{x^{i-1} \Leftrightarrow x^i}{\mid x^{i-1} \Leftrightarrow x^i \mid} \right) + \alpha_1 \sum_p \Lambda^p(i)(x_p \Leftrightarrow x^i), \qquad (6.25)$$

where again, the time-step Δt equals the current temperature. We notice that all elastic net forces are normalized now. Moreover, if $\exists i : x^{i+1} = x^i$, we get into trouble[4] . As self-evident analysis [35] shows, the elastic net forces try to push elastic net points onto a straight line, just like in the original ENA. However, once a net point is situated at *any* point on the straight line between its neighbouring net points, it no longer feels any elastic net force since the resulting force F_{res} equals zero. This is simply caused by the normalization of the elastic net forces: see figure 6.13. This means that equidistance is no longer pursued. Consequently, elastic net points will have more freedom in moving towards cities. It is therefore hoped that application of the non-equidistant elastic net algorithm (NENA) will (a) nearly always yield feasible solutions (of high quality), if the same number of elastic net points is used as in the original ENA and-or (b) often yield feasible solutions too, if a smaller number of elastic net points is chosen[5]. Stated in more general terms, it is hoped that the new algorithm will yield valid solutions more easily.

[3] At some places in the literature [77, 90], a linear distance measure is suggested, but nowhere did we find an elaborated implementation of this idea.

[4] In practice fortunately, this never occurred.

[5] It is even conjectured [77] that by using a linear distance measure, the number of elastic net points could be equal to the number of cities.

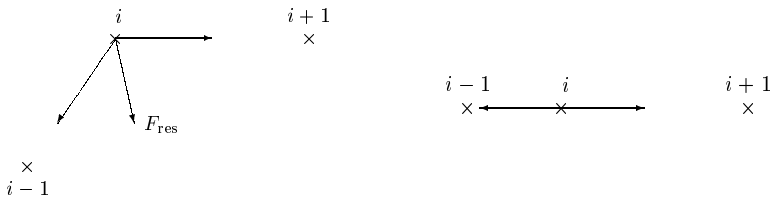


Figure 6.13: The new elastic net forces: general case (left), 3 points in line (right).

Since the elastic net forces are normalized by the new algorithm (those of the old one are not), a tuning problem arises. To solve this problem, the following simple approach is chosen: in the motion equations (6.25), all elastic net forces will be multiplied by the same factor

$$A(x) = \frac{1}{m} \sum_{1}^{m} \mid x^{i+1} \Leftrightarrow x^{i} \mid, \tag{6.26}$$

which represents the average distance between two elastic net points. Thus, the average elastic net force is roughly equal to the average in the original algorithm, and the final updating rule becomes:

$$\Delta x^{i} = \frac{\alpha_2}{\beta} A(x) \left(\frac{x^{i+1} \Leftrightarrow x^{i}}{\mid x^{i+1} \Leftrightarrow x^{i} \mid} + \frac{x^{i-1} \Leftrightarrow x^{i}}{\mid x^{i-1} \Leftrightarrow x^{i} \mid} \right) + \alpha_1 \sum_{p} \Lambda^{p}(i)(x_p \Leftrightarrow x^{i}), \tag{6.27}$$

where the values α_1, α_2 and β are chosen conform the original ENA.

6.3.2 The hybrid approach

A fundamental problem of the ENA is, that it might lead to non-feasible solutions due to the fact that the elastic net points adhere to equidistance. Moreover, equidistance is not required for the final solution of the elastic net (although it might be very useful in the initial phase of the algorithm in order to realize a smooth stretching out of the elastic ring). A fundamental problem of NENA is, that net points may become too lumpy (see the next section), which at least for larger problem instances, leads to non-feasibility and a lower quality of the subsequent solutions. Contemplating these considerations we tried to merge the two algorithms into a hybrid one retaining the best properties of both. The approach of the hybrid elastic net algorithm (HENA) is simple: the algorithm starts using ENA and, after a certain number of iterations, switches to NENA. The first phase is used in order to get a balanced stretching of the elastic net which is hoped to lead to solutions of high quality, the second phase is used in order to try to guarantee feasibility in the end. A consequence of this hybrid approach is the introduction of two new parameters. First, we have to decide at what time the switch should take place, and then, we have to choose the starting temperature after the switch.

6.4 Computational results

We now describe some of the experimental results as obtained with the NENA and the HENA, and compare them with computational results found using the original ENA.

6.4.1 A small problem instance

We start by using the configuration of cities as described in the theoretical analysis of section 4 (the 5 cities are situated as given in figure 6.7). In all cases, we used the following initialization of the elastic net: elastic net points are put in a small ring in the center of the state space, where the position of every net point is slightly randomized.

Using 5, 7, 10 or 12 elastic net points, the ENA produced only non-feasible solutions: in all experiments, one elastic net point is found in the middle between the two closely situated city points. The other 3 cities are always visited, while all other net points are more or less spread equidistantly. However, using 15 elastic net points, the optimal and feasible solution is always found: apparently, the number of elastic net points is now large enough to guarantee both feasibility and optimality.

Using 5 elastic net points, the NENA nearly always produced a non-feasible solution, but sometimes the optimal, feasible one. A gradual increase of the number of elastic net points results into a rise of the percentage of optimal solutions found. Using only 10 elastic net points yields a 100% score. An inspection of the final results reveals that the elastic net points become lumpy: they appear to come together around a city, which is of course, a consequence of their increased freedom. The number of net points per city depends the initialization as well as the location of the city.

We conclude that for this small problem instance the NENA produces better results than the ENA, or, stated more precisely, using a smaller number of elastic net points the NENA finds the the same optimal solution as the original ENA. The described experimental results are completely consonant with the theoretical conjectures of section 4.

6.4.2 Larger problem instances

Using a 15-city-problem, we had the similar experiences: it is easier to arrive at a feasible solution using the NENA. E.g., using 30 elastic net points, the NENA always yielded the same solution (namely the best solution found with both the ENA and the NENA), while the ENA sometimes yielded that solution, and sometimes a non-valid one.

However, the picture starts to change, if 30-city problem instances are chosen. As a rule, both algorithms are equally disposed to finding a valid solution, but another phenomenon turns up: the quality of the solutions found by the original ENA was generally better. Inspection of the solutions found by the NENA, demonstrated a strong lumping effect. The lumping can be so strong that

sometimes a city is left out completely. Especially cities which are situated at a point where the final tour bends substantially, may be overlooked. Apparently, by disregarding the property of equidistance, a new problem has originated. Re-evaluating, we conclude that the equidistance property of the ENA has an important contribution towards finding solutions of high quality, i.e., short tours.

At this point, the hybrid approach of HENA comes to mind. Because for small problem instances the NENA works better than the ENA, we only tried larger problem instances. Unfortunately, in our experiments the HENA appears to be slightly worse than the original ENA both in relation to the quality of the solution and in relation to feasibility. E.g., taking a 100-city problem, the ENA usually yielded a solution where 99 of the 100 cities are visited, while in case if the HENA, on average 98 to 99 cities are visited. Moreover, the encountered tour length using the ENA is, on average, slightly better than the tour length found by the HENA. Trying larger problem instances, we were unable to find parameters of the HENA, which yield better solutions than the original ENA or which guarantee feasibility of solutions.

Chapter 7

Conclusions, discussions, and outlook

In this final chapter, we first sum up our results, we then discuss the most important ones (especially in relation to the research objectives as mentioned in chapter 1), and finally, we dwell on what is left and can be grappled with in future research.

7.1 A list of results

We start by calling to mind that the results concerning the Hopfield models generally also hold for the relaxation phase of recurrent neural networks with learning capabilities. Assembling several outcomes, we arrive at the following list:

- The unconstrained stochastic binary Hopfield model, as well as the one constrained by equation (4.1), can, in mean field approximation, be described by a corresponding continuous Hopfield model.

  In both cases, two approximating free energy expressions have been found whose stationary points coincide. However, the types (minimum or maximum) of these stationary points are not necessarily identical: this striking phenomenon appears to be connected to the structure of the given problem instance, i.e., to the weight values w_{ij}.

  One of the two mean field free energy expressions can be written in the standard form (2.8), known from thermodynamics, for either of the models. This form yields an explicit approximating formula for the entropy and therefore for the effect of noise (thermal fluctuations) in the system at the same time. In general, the effect of noise is a displacement of solutions towards the interior.

  Conditions that guarantee the stability of various motion equations of both models have been given, some of which are easy, and some of which may be

hard to check. They appear to depend on either the transfer function chosen or the properties of the matrix (w_{ij}).

The apotheosis of chapters 3 and 4 is the 'most general framework' including the corresponding stability theorem. Provided certain general mathematical conditions are fulfilled, the generalized continuous Hopfield networks can model almost arbitrary energy expressions and can, in principle, incorporate several types of constraints. The corresponding free energy expressions are functions in both the input and the output of all neurons.

The experimental results did not falsify the theoretical conjectures: simple (quadratic and non-quadratic) optimization problems and 'purely combinatorial' ones have been resolved successfully. In relation to the required computation time, it appeared to be advantageous to build-in the constraints as much as possible (instead of applying a penalty approach).

However, a startling outcome of certain experiments is that there exist formulations of the built-in constraints that destroy the usual statistical mechanical interpretation, that is, the usual mean field approximation where the energy of the continuous Hopfield model can be written as a free energy expression of the form (2.8). Stability may still occur while, at the same time, neither the ordinary solutions at high temperatures, nor the usual approximation of the original cost (energy) function at low temperatures, are found.

- A new potential Lyapunov function for both the unconstrained and the constrained continuous Hopfield-Lagrange model has been presented. In the unconstrained case, all constraints can be grappled with using Lagrangian multipliers. In the constrained case, part of the constraints are tackled this way whereas the other constraints are built-in directly (or, if desired, handled by means of penalty terms). The Lyapunov function may serve as a tool to prove stability of the corresponding differential equations, although the analysis of the dynamic conditions may be hard.

  Quadratic constraints and others that meet the penalty terms condition (2.37) generally guarantee stability of the Hopfield-Lagrange model, at the same time degenerating the model to a dynamic penalty model. In these cases, the multiplier values grow during the updating of the differential equations. If the multipliers are smaller than a set of critical values, the model is unstable. Otherwise, the system is stable. The transition from unstable to a stable behavior in some respect resembles a phase transition in statistical mechanics.

  If the formulation of the constraints is such that, in spite of the presence of thermal noise in the system, solutions are situated in corner points of the hypercube $[0,1]^n$, we call these constraints 'hard'. The nasty property of this type of constraints is that the solution values of the U_i's equal $\pm\infty$. This problem can be resolved by explicitly relaxing the constraints in such a way that the constrained solutions are slightly dragged towards the interior.

  Various practical problems have been resolved using the Hopfield-Lagrange model. Simple quadratic optimization problems subject to linear constraints

were always solved correctly. In other problems, linear constraints often resulted in instability. Alternatively, using quadratic constraints, the experiments with the weighted matching problem always yielded solutions of 'good' quality, and even the travelling salesman problem could be resolved in a 'reasonable' way. However, the scale of the last two problems was still rather small, while in case of the TSP, the computation time of the (sequential) simulation could rise up to several hours. In general, the larger the number of multipliers is, the better the quality of the solutions appears to be. We further noticed that in case of the TSP, the (partially) strong approach using built-in constraints yielded solutions of a worse quality than the soft approach using only multipliers.

- Contrary to a well-known conclusion in the technical literature [77], the elastic net algorithm can not be derived from a constrained stochastic Hopfield network with Hamiltonian (6.1). Instead, the ENA should be considered a type of dynamic penalty method (which we have termed a thermal or noisy penalty method) where, unlike the degenerating Hopfield-Lagrange model, the weight values of the penalty terms gradually decrease. The lowering of the temperature should be viewed as a tuning process between the cost function and the penalty terms, unlike optimal simulated annealing.

  Trying a non-equidistant elastic net algorithm with a correct distance measure, as well as a hybrid algorithm, only small problem instances yielded better solutions suggesting that the quadratic distance measure of the original ENA is an essential ingredient.

7.2 Discussion

Summarizing the previous list, we hold that the various theorems on generalized Hopfield models, the stability theorems on the Hopfield-Lagrange models, and the notion and existence of various dynamic penalty models (both in case of the Hopfield-Lagrange and of the elastic neural networks) are central points of this thesis.

We now look more precisely at how far our ends have been achieved. The main objectives of *explaining* the relaxation dynamics and of *generalizing* existing theories have certainly been achieved in some measure. Various theorems concerning the (un)constrained continuous Hopfield models as well as Hopfield-Lagrange models have been derived which give general conditions that guarantee stability. Besides, the statistical mechanical interpretation of the continuous Hopfield models discussed, elucidates their working and suggests the application of mean field annealing. In relation to the Hopfield-Lagrange model, the unmasking of the effect of quadratic constraints (guaranteeing stability and showing behavior like dynamic penalty terms) is essential. The exposure of the elastic net algorithm as a noisy penalty method is an extension of this 'dynamic penalty view'.

Furthermore, the discovery of the most general framework leads to increased freedom in configuring continuous Hopfield models. Given a formulation of the

problem at hand, one can choose between various updating rules, various transfer functions, various formulations of the constraints, and even various models: constraints may be built-in or otherwise be tackled using Lagrange multipliers or (dynamic) penalty terms. This can be exploited in applications (see also the crucial observations at the end of section 4.3.3). It is quite interesting that very recently and independently of the analysis as given in this thesis, articles have been published [24, 80], which, in fact, can be considered as primary explorations of the practical capabilities of the most general framework.

On the other hand, certain unanswered, though quite general, theoretical questions still remain. E.g., we could ask ourselves whether stochastic Hopfield neural networks can also be generalized to the most general framework of this thesis. Considering the general framework in relation to continuous Hopfield models, it is very important to discover the precise conditions on the built-in constraints which assert the statistical mechanical interpretation. It is also desirable to find out which general classes of cost functions, subject to several (non-)linear constraints, guarantee stability of the Hopfield-Lagrange model. Last but not least, it is all-important to investigate in which ways domain knowledge of problems can be systematically incorporated in recurrent neural networks.

The very last observation also stems from the study of the relationship between Hopfield and elastic neural networks, whose explanation fulfills one of the secondary purposes of this thesis. This relationship has become quite clear and the elastic net itself beautifully shows how domain knowledge can be incorporated in a recurrent neural network, yielding a comparatively excellent algorithm.

Finally, we consider the models from the viewpoint of applicability, which refers to the other secondary objective of this thesis. First of all, we note that the afore-mentioned freedom in configuring the neural networks at hand, is also a drawback, since there are so many choices to be made. At the same time, it is often quite unclear in advance, which choice will yield the best results. Likewise, the analytical verification of the conditions that guarantee stability (i.e., the definite positiveness of dynamically changing matrices) might be a tough task.

We further realize that our experimental tests done so far do not yield the decisive answer of whether neural networks can always adequately be applied to solve combinatorial optimization problems. Evaluating our computational results and those known from the technical literature, it seems true that, for 'purely combinatorial' problems, ANNs work fine. However, if optimization joins in the game, a high quality of the constrained solutions is not guaranteed. In general, we observed the following, maybe not too surprising tendency: the more difficult the problems are (e.g., those from the class of $\mathcal{P}$ compared to those from the class of $\mathcal{NP}$-hard problems), the worse is the quality of the encountered solutions and the longer is the required computation time. In cases of difficult problems, a tailored approach where domain knowledge is applied seems to be necessary, as is a lot of 'tuning' work. If seemingly small adaptations of a well-balanced algorithm like the elastic net algorithm are tried (e.g., the discussed non-equidistance and the hybrid elastic net algorithms), new tuning work is immediately necessary and the solutions found may be of lesser quality. This demonstrates the high sensitivity of the adjustable parameters of these tailored algorithms.

We finish this section by observing that it is always an option to use dedicated hardware or specially tailored software in order to implement successful applications which require a lot of computation time. It is hoped that these implementations can be parallelized in order to speed up performance.

7.3 Outlook

In this section, we try to list a number of questions that beg for an answer. We first take up the theoretical ones.

- Which combinations of new transfer functions[1] g_i and summation functions h_i fit in the most general framework of chapter 4 and turn out to yield stable models?
- In which ways can the general free energy F_{mgf} of the most general framework be interpreted, or, stated more precisely, which general conditions should hold for the transfer functions $V_i = g_i(U)$ such that the continuous Hopfield model can be considered a mean field approximation of a corresponding stochastic model?
- Which general conditions, especially those relating to *linear* constraints, can guarantee stability of the Hopfield-Lagrange model?
- Which theoretical improvements of constrained Hopfield networks can be learned from the 'deformable template' approach, or, stating this more generally, in which systematic ways can domain knowledge be incorporated in Hopfield neural nets?
- Can the effect of the application of mean field annealing be better understood, e.g., by an energy surface analysis, as has been done for the elastic net algorithm?
- Which similarities and differences exist between an iterative updating strategy like (2.29) and the continuous ones like (2.30) as studied in this thesis?
- Can stochastic Hopfield neural networks, i.e. Boltzmann machines, also be generalized to the most general framework?

In the practical field, the following questions beg for an answer:

- Considering the amounts of computation time and the qualities of solutions, what is the relation between the results found by stochastic Hopfield models and those found by continuous ones[2]?
- Applying either continuous or stochastic Hopfield models, which annealing schemes can best improve the quality of solutions of hard problems?

[1] Inspiration can be gleaned from the literature, e.g., from [82].

[2] As mentioned above, certain results have already been reported in the technical literature, e.g., in [68].

- Which transfer functions g_i and which summation functions h_i work well in practice?
- Which (combinations of) ways to grapple with a set of given constraints (i.e, those using penalty terms, those which build them in, and those which apply multipliers) work best?
- Which alternative dynamic penalty models (including the noisy penalty models) can improve the application of Hopfield and allied networks?
- Can the performance of the non-equidistant elastic algorithms be further improved by a better tuning of the parameters involved?
- By means of which hardware and / or software is it possible to speed up the calculations of the models discussed in this thesis?

7.4 In conclusion

We finish this thesis by stating that there is still a lot of work to do in order to understand in detail the behavior of the Hopfield(-Lagrange) models and in order to gain a lucid understanding of which type of these models should be chosen to solve a given combinatorial optimization problem in an adequate way. Several important insights relating to these questions have been gained by the work reported in this thesis. And although certain theoretical questions continue to exist, and beg to be resolved, it seems to be appropriate to shift, at the same time, our attention towards a more practical approach. This will probably yield an even better insight in how the models behave in practice, which, in fact, is one of the all-important motives to study these models. It is hoped that this practical approach will also produce new, indispensable inspiration as well as intuition for a refreshed theoretical approach in times to come.

Appendix A

Lagrange multipliers

The Lagrange[1] multiplier method is a method for analyzing constrained optimization problems defined by

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to :} & C_\alpha(x) = 0,\ \alpha = 1, \ldots, m, \end{array} \tag{A.1}$$

where $f(x)$ is called the objective function, $x = (x_1, x_2, \ldots, x_n)$, and the equations $C_\alpha(x) = 0$ are m side conditions or constraints. The *Lagrangian function* L is defined by a linear combination of the objective function f and the m constraint functions C_α conform

$$L(x, \lambda) = f(x) + \sum_\alpha \lambda_\alpha C_\alpha(x), \tag{A.2}$$

where $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$ and where the λ_i's are called *Lagrange multipliers*. The class of functions whose partial derivatives are continuous we shall denote by $\mathcal{C}^1$. The following theorem gives a *necessary* condition for f to have a local extremum subject to the constraints (A.1):

Theorem A.1. *Let $f \in \mathcal{C}^1$ and all functions $C_\alpha \in \mathcal{C}^1$ be real functions on an open set T of R^n. Let W be the subset of T such that $x \in W \Rightarrow \forall \alpha : C_\alpha(x) = 0$. Assume further that $m < n$ and that some $m \times m$ submatrix of the Jacobian associated with the constraint functions C_α is nonsingular at $x^0 \in W$, that is, we assume that the following Jacobian is nonsingular at x^0*

$$J(x^0) = \begin{pmatrix} C_1^1(x^0) & C_1^2(x^0) & \ldots & C_1^m(x^0) \\ \vdots & \vdots & & \vdots \\ C_m^1(x^0) & C_m^2(x^0) & \ldots & C_m^m(x^0) \end{pmatrix} \tag{A.3}$$

[1] Joseph-Louis Lagrange (1736–1813) worked in many branches of both mathematics and physics. He mentioned the multiplier method in a letter to Euler in 1755, where he applied it to infinite-dimensional problems. In 1797, his book 'Theory of analytic functions' appeared containing the extension to finite-dimensional problems (historical remarks from [83]).

where $C^j_\alpha(x^0) = \partial C_\alpha(x^0)/\partial x_j,\ j \in \{1, \ldots, n\}$.
If f *assumes a local extremum at* $x^0 \in W$*, then there exist real and unique numbers* $\lambda^0_1, \ldots, \lambda^0_m$ *such that the Lagrangian* $L(x, \lambda)$ *has a critical point in* (x^0, λ^0).

The proof of this theorem [8] rests on the 'implicit function theorem'. The condition of the nonsingularity of the defined Jacobian matrix makes the vector λ^0 unique. The theorem can also be extended by giving *sufficient* conditions for f to have a local constrained minimum or a local constrained maximum.

Appendix B

Dynamic systems

The theory of dynamic systems [8, 54] refers to the analysis of systems in the course of time. Here, we confine ourselves to systems which are described by a set of differential or, respectively, a set of difference equations

$$\dot{x} = f(x,u), \tag{B.1}$$
$$x^{t+1} = f(x^t,u^t), \tag{B.2}$$

where $x=(x_1,\ldots,x_n)$ is called the state of the system, where $u=(u_1,\ldots,u_r)$ is the control or input of the system, and where $f=(f_1,\ldots,f_n)$ is a function vector with all f_i (non)linear functions in x and u. Both x and u are functions of time ($t\geq 0$). For fixed $u(t)$, the system is said to be *free* or unforced. We assume that there exists a function $x(t)$ which satisfies (B.1) or (B.2) for $t\geq 0$ starting at an initial state $x(0)=x_0$. Such a function is called a solution, motion, or trajectory. A state of the system is called an equilibrium state x_e if $\forall t : x(t)=x_e$.

We often want to know whether a (free) dynamic system is *asymptotically stable*, that is, we want to know under which conditions the system eventually, that is for $t\to\infty$, converges to an equilibrium state, even if we do not have knowledge of the trajectory. Here, the techniques using a Lyapunov[1] function L appear on the scene. We confine ourselves to free systems, which implies that x is a vector function only of time. We start considering the continuous case (B.1), where $\mathcal{C}^1$ is defined as in appendix A.

Theorem B.1. *Assume there exists a scalar function $L(x)\in\mathcal{C}^1$, bounded below by a real constant B such that*

$$L(x)\begin{cases} =B & \text{if } x=x_e \\ >B & \text{otherwise,}\end{cases} \tag{B.3}$$

and suppose the time derivative $\dot{L}$ of L along a solution of the system fulfills

$$\forall x\neq x_e : \dot{L}(x(t))<0, \tag{B.4}$$

[1]The approach given here is Lyapunov's direct or second method for obtaining stability information without explicit knowledge of the solutions. His first method involves an explicit representation of the solutions. The approach of Lyapunov (1857–1918) was published in 1892 [8].

then the equilibrium state x_e is a locally asymptotically stable *point.*

The precise proof of the theorem as well as rigorous definitions of the concepts of local and global (asymptotic) stability can be found in [8]. Roughly, the idea of the proof is that when time advances the function L strictly decreases until, eventually, an equilibrium state is reached. We further note that the monotonicity of L is a *sufficient*, but not a *necessary* condition for asymptotic stability [54]. This implies that inability to generate a Lyapunov function proves nothing.

A slightly weaker form of the given theorem is obtained by replacing the strict inequality (B.4) by

$$\forall x \neq x_e : \dot{L}(x(t)) \leq 0. \tag{B.5}$$

In this case, the system is simply called *stable.* Unlike for the asymptotically stable systems, the solution of a stable system need not reach the equilibrium point, but may hover arbitrary close to it [54].

For linear systems, there exists an explicit method for obtaining a Lyapunov function, where a certain linear system of equations should be resolved [8]. For nonlinear systems, such an explicit method does not exist. In case of studying physical systems, there is often an energy function which is minimized and which, at the same time, plays the part of a Lyapunov function.

The given theorem also holds for difference equation systems (B.2), provided we make some obvious modifications such as replacing integrals with sums , $\dot{x}$ with $\Delta x/\Delta t$, and making the continuous time t a discrete one. The Lyapunov function is discrete too and is decreased step by step by the corresponding updating rule.

Appendix C

Gradient descent

A very well-known method for finding a *local* extremum of a function $f(x)$ is the gradient method, where a gradient descent is applied in order to find a minimum, and a gradient ascent to find a maximum. Confining ourselves to the first one, the idea is to slide downhill from a certain starting point along the n-dimensional surface of the graphic of $f(x),\ x \in \mathbb{R}^n$. The gradient descent rule equals [44]

$$\dot{x}_i = \Leftrightarrow\kappa \frac{\partial f}{\partial x_i}, \tag{C.1}$$

where κ is a positive constant. Thus, we slide downhill with a 'speed' proportional to the slope of the hill.

Supposing that $f(x) \in \mathcal{C}^1$ ($\mathcal{C}^1$ defined as in appendix A) and that $f(x)$ is bounded below, asymptotic stability of (C.1) can be proven easily using the theory of appendix B: $f(x)$ itself is a Lyapunov function since

$$\dot{f}(x(t)) = \sum_i \frac{\partial f}{\partial x_i}\dot{x}_i = \Leftrightarrow\kappa \sum_i (\frac{\partial f}{\partial x_i})^2 \begin{cases} = 0 & \text{if } \forall i : \partial f/\partial x_i = 0 \\ < 0 & \text{otherwise,} \end{cases} \tag{C.2}$$

As long as at least one $\dot{x}_i \neq 0$, the way down continues until finally all $\dot{x}_i = 0$ and a local stationary point, defined by $\forall i : \partial f/\partial x_i = 0$, has been reached. In general, this stationary point is a local minimum.

Appendix D

Lemmas and their proofs

Lemma 1. *If A is a symmetric and nonsingular matrix then*

$$\exp(\tfrac{\beta}{2}x^T Ax) = \frac{\int \exp(\Leftrightarrow\frac{\beta}{2}\phi^T A^{-1}\phi \pm \beta\phi^T x)\prod_i \mathrm{d}\phi_i}{\int \exp(\Leftrightarrow\frac{\beta}{2}\phi^T A^{-1}\phi)\prod_i \mathrm{d}\phi_i}, \tag{D.1}$$

where the n-fold integrals on the right-hand side are improper integrals defined over $\mathbb{R}^n$.

Proof. The lemma is a generalization of the following trick

$$\exp(\tfrac{\beta}{2}x^2) = \frac{\int_\infty^\infty \exp(\Leftrightarrow\frac{\beta}{2}\phi^2 \pm \beta\phi x)\mathrm{d}\phi}{\int_\infty^\infty \exp(\Leftrightarrow\frac{\beta}{2}\phi^2)\mathrm{d}\phi}, \tag{D.2}$$

This trick can simply be derived by elaborating the integral of the numerator of the right-hand side of (D.2). Applying it with

$$xy = \left(\frac{x+y}{2}\right)^2 \Leftrightarrow \left(\frac{x \Leftrightarrow y}{2}\right)^2,$$

and knowing that

$$\int e^\phi \mathrm{d}\phi \int e^\psi \mathrm{d}\psi = \int e^\phi e^\psi \mathrm{d}\phi\mathrm{d}\psi,$$

we can write:

$$\begin{aligned}\exp(\tfrac{\beta}{2}xy) &= \frac{\int \exp[\Leftrightarrow\frac{\beta}{2}(\phi^2 \Leftrightarrow\psi^2) \pm \frac{\beta}{2}(\phi x + \phi y \Leftrightarrow\psi x + \psi y)]\mathrm{d}\phi\mathrm{d}\psi}{\int \exp[\Leftrightarrow\frac{\beta}{2}(\phi^2 \Leftrightarrow\psi^2)]\mathrm{d}\phi\mathrm{d}\psi} \\ &= \frac{\int \exp[\Leftrightarrow\frac{\beta}{2}\tilde{\phi}\tilde{\psi} \pm \frac{\beta}{2}(\tilde{\phi}x + \tilde{\psi}y)]\mathrm{d}\tilde{\phi}\mathrm{d}\tilde{\psi}}{\int \exp[\Leftrightarrow\frac{\beta}{2}\tilde{\phi}\tilde{\psi}]\mathrm{d}\tilde{\phi}\mathrm{d}\tilde{\psi}},\end{aligned}$$

where $\tilde{\phi} = \phi \Leftrightarrow\psi$ and $\tilde{\psi} = \phi + \psi$. If x and y are n-dimensional vectors and if A is an $n \times n$-matrix, we can generalize this result to

$$\exp(\tfrac{\beta}{2}x^T Ay) = \frac{\int \exp[\Leftrightarrow\frac{\beta}{2}\phi^T\psi \pm \frac{\beta}{2}(\phi^T x + \psi^T Ay)]\mathrm{d}\phi\mathrm{d}\psi}{\int \exp[\Leftrightarrow\frac{\beta}{2}\phi^T\psi]\mathrm{d}\phi\mathrm{d}\psi},$$

where ϕ and ψ are n-dimensional vectors too. If the matrix A is symmetric and nonsingular, then the substitution $\psi \rightarrow A^{-1}\psi$ implies that

$$\psi^T A y \rightarrow (A^{-1}\psi)^T A y = \psi^T (A^{-1})^T A y = \psi^T y.$$

Applying this, we find

$$\exp(\tfrac{\beta}{2} x^T A y) = \frac{\int \exp[\Leftrightarrow \frac{\beta}{2}\phi^T A^{-1}\psi \pm \frac{\beta}{2}(\phi^T x + \psi^T y)]\mathrm{d}\phi\mathrm{d}\psi}{\int \exp[\Leftrightarrow \frac{\beta}{2}\phi^T A^{-1}\psi]\mathrm{d}\phi\mathrm{d}\psi}.$$

By finally substituting $y \rightarrow x$ and writing $\mathrm{d}\phi = \prod_i \mathrm{d}\phi_i$, the theorem is found. □

Lemma 2. *If the integrand in both the numerator and the denumerator is expanded in a Taylor series expansion around its saddle point, then the following equation holds*

$$\frac{\int \exp(\Leftrightarrow \frac{\beta}{2}\phi^T A^{-1}\phi \pm \beta\phi^T x) \prod_i \mathrm{d}\phi_i}{\int \exp(\Leftrightarrow \frac{\beta}{2}\phi^T A^{-1}\phi) \prod_i \mathrm{d}\phi_i} = \exp(\tfrac{\beta}{2} x^T A x). \tag{D.3}$$

Proof. We only furnish a detailed proof in the one-dimensional case. Since β is a scaling factor, we can simply equalize it to one without effecting the course of the proof. Under these conditions, we merely have to proof that

$$\frac{\int_{-\infty}^{\infty} \exp(\Leftrightarrow \frac{\phi^2}{2a} \pm \phi x)\mathrm{d}\phi}{\int_{-\infty}^{\infty} \exp(\Leftrightarrow \frac{\phi^2}{2a})\mathrm{d}\phi} = \exp(\tfrac{1}{2} a x^2).$$

Taking $f_x(\phi) = \exp(\Leftrightarrow \frac{\phi^2}{2a} \pm \phi x)$, the saddle point $\hat{\phi} = \pm ax$ of the numerator is found by solving $\mathrm{d}f_x/\mathrm{d}\phi = 0$. Application of a Taylor series expansion around this saddle point yields

$$\begin{aligned}
& f_x(\phi) \\
&= f_x(\hat{\phi}) + f_x^{''}(\hat{\phi})(\phi \Leftrightarrow \hat{\phi})^2/2 + f_x^{'''}(\hat{\phi})(\phi \Leftrightarrow \hat{\phi})^3/6 + f_x^{''''}(\hat{\phi})(\phi \Leftrightarrow \hat{\phi})^4/24 + \ldots \\
&= f_x(\pm ax) \mp f_x(\pm ax)(\phi \mp ax)^2/2a + f_x(\pm ax)(\phi \mp ax)^4/8a^2 + \ldots \\
&= f_x(\pm ax)\left[1 \mp (\phi \mp ax)^2/2a + (\phi \mp ax)^4/8a^2 + \ldots\right]
\end{aligned}$$

It follows that

$$\begin{aligned}
& \frac{\int_{-\infty}^{\infty} \exp(\Leftrightarrow \frac{\phi^2}{2a} \pm \phi x)\mathrm{d}\phi}{\int_{-\infty}^{\infty} \exp(\Leftrightarrow \frac{\phi^2}{2a})\mathrm{d}\phi} \\
&= \frac{\int_{-\infty}^{\infty} f_x(\pm ax)\left[1 \mp (\phi \mp ax)^2/2a + (\phi \mp ax)^4/8a^2 + \ldots\right]\mathrm{d}\phi}{\int_{-\infty}^{\infty} f_0(0)\left[1 \mp \phi^2/2a + \phi^4/8a^2 + \ldots\right]\mathrm{d}\phi} \\
&= \exp(\tfrac{1}{2} a x^2) \frac{\int_{-\infty}^{\infty}\left[1 \mp (\phi \mp ax)^2/2a + (\phi \mp ax)^4/8a^2 + \ldots\right]\mathrm{d}\phi}{\int_{-\infty}^{\infty}\left[1 \mp \phi^2/2a + \phi^4/8a^2 + \ldots\right]\mathrm{d}\phi} \\
&= \exp(\tfrac{1}{2} a x^2) \times \lim_{p\rightarrow\infty} \frac{\left[\phi \mp (\phi \mp ax)^3/6a + (\phi \mp ax)^5/40a^2 + \ldots\right]_{-p}^{0} + [\ldots]_0^p}{\left[\phi \mp \phi^3/6a + \phi^5/40a^2 + \ldots\right]_{-p}^{0} + [\ldots]_0^p} \\
&= \exp(\tfrac{1}{2} a x^2) \times \lim_{p\rightarrow\infty} (1 + \mathcal{O}(\frac{1}{p})) = \exp(\tfrac{1}{2} a x^2).
\end{aligned}$$

This completes the proof of the one-dimensional case. In a similar way, by application of an n-dimensional Taylor expansion, the correctness of equation (D.3) can be proven. □

Note. Since equations (D.1) and (D.3) coincide, lemma 2 seems to be superfluous: conform lemma 1, the Taylor series expansion as applied in the proof of lemma 2, should yield equation (D.3). The reason to yet furnish this proof is to explain the difference between this approach (yielding an exact result) and that of lemma 4, where, for mathematical complications, the Taylor series expansion is cut off (yielding an approximating result).

Lemma 3. *If S passes through all 2^n states from $(0, 0, \ldots, 0)$ to $(1, 1, \ldots, 1)$, then*

$$\sum_S \exp(\beta \sum_i S_i \phi_i) = \exp(\sum_i \ln(1 + \exp(\beta \phi_i))). \tag{D.4}$$

Proof. The proof can be done by induction on the number of neurons S_i. For one S_i, we find

$$\begin{aligned} \sum_S \exp(\beta \sum_i S_i \phi_i) &= \sum_S \exp(\beta S_1 \phi_1) \\ &= 1 + \exp(\beta \phi_1) = \exp(\ln(1 + \exp(\beta \phi_1))). \end{aligned}$$

Suppose the lemma is true for $(n \Leftrightarrow 1)$ neurons S_i, then we can write

$$\begin{aligned} \sum_S \exp(\beta \sum_{i=1}^{n} S_i \phi_i) &= \sum_S \exp(\beta \sum_{i=1}^{n-1} S_i \phi_i) \times \exp(\beta S_n \phi_n) \\ &= \exp(\sum_{i=1}^{n-1} \ln(1 + \exp(\beta \phi_i))) \times (1 + \exp(\beta_n \phi_n)) \\ &= \exp(\sum_{i=1}^{n} \ln(1 + \exp(\beta \phi_i))). \end{aligned}$$

This completes the proof. □

Lemma 4. *A first order saddle point approximation in the numerator and denumerator (both regarded as a function in ϕ) of*

$$Z_{hu} = \frac{\int \exp\left[\Leftrightarrow \frac{\beta}{2} \sum_{ij} \phi_i w_{ij}^{-1} \phi_j + \sum_i \ln\left(1 + \exp(\beta(\phi_i + I_i))\right)\right] \prod_i \mathrm{d}\phi_i}{\int \exp\left[\Leftrightarrow \frac{\beta}{2} \sum_{ij} \phi_i w_{ij}^{-1} \phi_j\right] \prod_i \mathrm{d}\phi_i},$$

where

$$E(\phi, I) = \tfrac{1}{2} \sum_{ij} \phi_i w_{ij}^{-1} \phi_j \Leftrightarrow \tfrac{1}{\beta} \sum_i \ln\left[1 + \exp\left(\beta(\phi_i + I_i)\right)\right],$$

yields

$$V_i \approx \Leftrightarrow \frac{\partial E(\tilde{\phi}, I)}{\partial I_i}.$$

Proof. The proof can be done in the same way of that of lemma 2. However, this time the Taylor expansion is *cut off* after the second term conform

$$E(\phi, I) = E(\tilde{\phi}, I) + \sum_i \frac{\partial E(\tilde{\phi}, I)}{\partial \phi_i}(\phi \Leftrightarrow \tilde{\phi}) + \mathcal{O}(\phi^2) \approx E(\tilde{\phi}, I).$$

Using this approximation and a similar one in the denumerator, we find that

$$Z_{hu} \approx \frac{\int \exp(\Leftrightarrow \beta E(\tilde{\phi}, I)) \mathrm{d}\phi}{\int \exp(0) \mathrm{d}\phi} = \exp(\Leftrightarrow \beta E(\tilde{\phi}, I)).$$

Substituting this result in (2.14), we find

$$V_i = \tfrac{1}{\beta} \frac{\partial \ln Z_{hu}}{\partial I_i} \approx \Leftrightarrow \frac{\partial E(\tilde{\phi}, I)}{\partial I_i}.$$

This completes the proof. □

Lemma 5. *If*

$$V_i = \frac{1}{1 + \exp(\Leftrightarrow U_i)}, \tag{D.5}$$

then

$$\ln\left(1 + \exp(U_i)\right) = \Leftrightarrow V_i \ln V_i \Leftrightarrow (1 \Leftrightarrow V_i)\ \ln(1 \Leftrightarrow V_i) + V_i U_i.$$

Proof. Equation (D.5) implies that

$$1 \Leftrightarrow V_i = \frac{1}{1 + \exp(U_i)}. \tag{D.6}$$

Using (D.5) and (D.6), we can proof the lemma directly:

$$\begin{aligned}
&\Leftrightarrow V_i \ln V_i \Leftrightarrow (1 \Leftrightarrow V_i) \ln(1 \Leftrightarrow V_i) + V_i U_i = \\
&= \frac{\Leftrightarrow 1}{1 + \exp(\Leftrightarrow U_i)} \ln\left(\frac{\Leftrightarrow 1}{1 + \exp(\Leftrightarrow U_i)}\right) \Leftrightarrow \\
&\qquad \frac{\Leftrightarrow 1}{1 + \exp(U_i)} \ln\left(\frac{\Leftrightarrow 1}{1 + \exp(U_i)}\right) + \frac{U_i}{1 + \exp(\Leftrightarrow U_i)} \\
&= \frac{\ln(1 + \exp(\Leftrightarrow U_i)) + \ln \exp(U_i)}{1 + \exp(\Leftrightarrow U_i)} + \frac{\ln(1 + \exp(U_i))}{1 + \exp(U_i)} \\
&= \ln(1 + \exp(U_i))(V_i + 1 \Leftrightarrow V_i) \ = \ \ln(1 + \exp(U_i)).
\end{aligned}$$

□

Lemma 6. *If S passes through all n states from $(1,0,\ldots,0)$, $(0,1,\ldots,0)$, $\ldots$, to $(0,0,\ldots,1)$, then*

$$\sum_S \exp(\beta \sum_i S_i\phi_i) = \exp(\ln \sum_i \exp(\beta\phi_i)),$$

Proof. The proof is almost trivial:

$$\sum_S \exp(\beta(\sum_{i=1}^{n} S_i\phi_i)) = \sum_i \exp(\beta\phi_i) = \exp(\ln \sum_i \exp(\beta\phi_i)).$$

□

Lemma 7. *If*

$$V_i = \frac{\exp(\pm\beta U_i)}{\sum_l \exp(\pm\beta U_l)}, \tag{D.7}$$

then

$$\ln \sum_i \exp(\pm\beta U_i) = \Leftrightarrow \sum_i V_i \ln V_i \pm \sum_i \beta U_i V_i.$$

Proof. From equation (D.7) it follows that

$$\pm\beta U_i = \ln\big(V_i \sum_l \exp(\pm\beta U_l)\big).$$

Using this result and the fact that $\sum_i V_i = 1$ we can write

$$\begin{aligned}
\pm \sum_i \beta U_i V_i &= \sum_i \ln\big(V_i \sum_l \exp(\pm\beta U_l)\big) V_i \\
&= \sum_i V_i \ln V_i + \sum_i V_i \ln(\sum_l \exp\big(\pm\beta U_l)\big) \\
&= \sum_i V_i \ln V_i + \ln\big(\sum_l \exp(\pm\beta U_l)\big).
\end{aligned}$$

Rewriting this equation, the lemma is found immediately. □

Lemma 8. *If* (D.7) *holds using the plus sign, if $l \geq 2$, and if $l \neq i$, then*

$$\frac{\partial V_i}{\partial U_i} = \beta V_i(1 \Leftrightarrow V_i) > 0 \text{ and } \frac{\partial V_i}{\partial U_l} = \Leftrightarrow \beta V_i V_l < 0. \tag{D.8}$$

Proof.

$$\begin{aligned}
\frac{\partial V_i}{\partial U_i} &= \frac{\sum_l \exp(\beta U_l).\exp(\beta U_i).\beta \Leftrightarrow \exp(\beta U_i).\exp(\beta U_i).\beta}{(\sum_l \exp(\beta U_l))^2} \\
&= \frac{\beta \exp(\beta U_i).(\sum_l \exp(\beta U_l) \Leftrightarrow \exp(\beta U_i)}{(\sum_l \exp(\beta U_l))^2} \\
&= \frac{\beta \exp(\beta U_i).\sum_{l\neq i} \exp(\beta U_l)}{(\sum_l \exp(\beta U_l))^2} = \beta V_i(1 \Leftrightarrow V_i) > 0.
\end{aligned}$$

The second result is found in the same way. Taking $l \neq i$ we find

$$\frac{\partial V_i}{\partial U_l} = \frac{0 \Leftrightarrow \exp(\beta U_i).\exp(\beta U_l).\beta}{(\sum_l \exp(\beta U_l))^2} = \Leftrightarrow \not{\beta} V_i V_l < 0.$$

Bibliography

[1] E.H.L. Aarts and J.K. Lenstra, editors. *Local Search in Combinatorial Optimization*. John Wiley & Sons, to appear in 1996.

[2] E.H.L. Aarts and J. Korst. *Simulated Annealing and Boltzmann Machines, A Stochastic Approach to Combinatorial Optimization and Neural Computing*. John Wiley & Sons, 1989.

[3] D.H. Ackley, G.F. Hinton, and T.J. Sejnowski. A Learning Algorithm for Boltzmann Machines. *Cognitive Science* 9, 147–169, 1985.

[4] M. Alonso and E.J. Finn. *Fundamentele Natuurkunde, Deel 6: Statistische Fysica*. Elsevier, 1979.

[5] B. Angéniol, G. de La Croix Vaubois, and J.-Y. Le Texier. Self-Organizing Feature Maps and the Travelling Salesman Problem. *Neural Networks* 1, 289–293, 1988.

[6] P.J. Antsaklis. Control theory approach. In *Mathematical Approaches to Neural Networks*, ed. J.G. Taylor, 261–292, Elsevier, 1993.

[7] A.M. Arthurs. *Calculus of Variations*. Routledge & Kegan Paul, 1975.

[8] A. Benavie. *Mathematical Techniques for Economic Analysis*. Prentice-Hall, 1972.

[9] J. van den Berg. A general framework for Hopfield neural networks. *Proceedings of the Seventh Dutch Conference on Artificial Intelligence*, NAIC'95, eds. J.C. Bioch and Y.-H. Tan, 265–274, Rotterdam, 1995.

[10] J. van den Berg. Sweeping generalizations of continuous Hopfield and Hopfield-Lagrange networks. Submitted to: *International Conference on Neural Information Processing*, ICONIP'96, Hong Kong, 1996.

[11] J. van den Berg. The most general framework of continuous Hopfield neural networks. To appear in: *Proceedings of the 1966 International Workshop on Neural Networks for Identification, Control, Robotics, and Signal/Image Processing*, NICROSP'96, Venice, IEEE Computer Society Press, 1996.

[12] J. van den Berg and J.C. Bioch. Capabilities of the Symbiosis between the Hopfield Model and Lagrange Multipliers in Resolving Constrained Optimization Problems. *Proceedings of Computing Science in the Netherlands 1994*, 54–65, Utrecht, 1994.

[13] J. van den Berg and J.C. Bioch. Constrained Optimization with a continuous Hopfield-Lagrange Model. *Technical Report EUR-CS-93-10*, Erasmus University Rotterdam, Comp. Sc. Dept., Faculty of Economics, 1993.

[14] J. van den Berg and J.C. Bioch. Constrained Optimization with the Hopfield-Lagrange Model. *Proceedings of the 14th IMACS World Congress*, 470–473, ed. W.F. Ames, Atlanta, GA 30332 USA, 1994.

[15] J. van den Berg and J.C. Bioch. On the (Free) Energy of Hopfield Networks. In *Neural Networks: The Statistical Mechanics Perspective*, Proceedings of the CTP-PBSRI Joint Workshop on Theoretical Physics, eds. J.H. Oh, C. Kwon, S. Cho, 233–244, World Scientific, Singapore, 1995.

[16] J. van den Berg and J.C. Bioch. On the Statistical Mechanics of (Un)Constrained Stochastic Hopfield and 'Elastic' Neural Networks. *Technical Report EUR-CS-94-08*, Erasmus University Rotterdam, Comp. Sc. Dept., Faculty of Economics, 1994.

[17] J. van den Berg and J.C. Bioch. Some Theorems Concerning the Free Energy of (Un)Constrained Stochastic Hopfield Neural Networks. Second European Conference on Computational Learning Theory, *Lecture Notes in Artificial Intelligence* 904: 298–312, ed. P. Vitányi, EuroCOLT'95, Springer, 1995.

[18] J. van den Berg and J.H. Geselschap. An analysis of various elastic net algorithms. *Technical Report EUR-CS-95-06*, Erasmus University Rotterdam, Comp. Sc. Dept., Faculty of Economics, 1995.

[19] J. van den Berg and J.H. Geselschap. A non-equidistant elastic net algorithm. To appear in: *Annals of Mathematics and Artificial Intelligence*, 1996.

[20] A.R. Bizzarri. Convergence Properties of a Modified Hopfield-Tank Model. *Biological Cybernetics* 64, 293–300, 1991.

[21] H. Bourlard. Theory and Applications to Speech Recognition. *Neural Networks Summer School*, University of Cambridge Engineering Department, 1992.

[22] M. Budinich. A Self-Organising Neural Network for the Travelling Salesman Problem that is Competitive with Simulated Annealing. Submitted to: *Neural Computation*, 1995.

[23] E. Charniak and D. McDermott. *Introduction to Artificial Intelligence*. Addison-Wesley, 1985.

[24] B.S. Cooper. Higher Order Neural Networks for Combinatorial Optimisation - Improving the Scaling Properties of the Hopfield Network. *Proceedings of the IEEE International Conference on Neural Networks*, 1855–1860, Perth, Western Australia, 1995.

[25] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., 1991.

[26] D.C. Dennett. *Darwin's Dangerous Idea - Evolution and the Meanings of Life*. Penguin, 1995.

[27] P. van Diepen and J. van den Herik. *Schaken voor computers*. Academic Service, Amsterdam, 1987.

[28] R. Durbin and D. Willshaw. An Analogue Approach of the Travelling Salesman Problem Using an Elastic Net Method. *Nature*, 326:689–691, 1987.

[29] R.C. Eberhart. Computational Intelligence: A Snapshot. In *Computational Intelligence, A Dynamic System Perspective*, eds. M. Palaniswami, Y. Attikiouzel, R.J. Marks II, D. Fogel, T. Fukuda, IEEE Press, 1995.

[30] L. Fausett. *Fundamentals of Neural networks, Architectures, Algorithms, and Applications*. Prentice Hall, 1994.

[31] B. Fritzke and P. Wilke. FLEXMAP–A Neural Network For The Traveling Salesman Problem With Linear Time And Space Complexity. Submitted to: *International Joint Conference on Neural Networks*, Singapore, 1991.

[32] Limin Fu. *Neural Networks In Computer Intelligence*. McGraw-Hill, 1994.

[33] M. Gardner. Forword in R. Penrose's book: *The Emperor's New Mind. Concerning Computers, Mathematics, and The Laws of Physics*. Oxford University Press, 1989.

[34] M.R. Garey and D.S. Johnson. *Computers and Intractability, A guide to the Theory of NP-Completeness*. W.H. Freeman, 1979.

[35] J.H. Geselschap. Een Verbeterd 'Elastic Net' Algoritme (An Improved Elastic Net Algorithm). *Master's thesis*, Erasmus University Rotterdam, Comp. Sc. Dept., Faculty of Economics, 1994.

[36] D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.

[37] K.M. Gutzmann. Combinatorial Optimization Using a Continuous State Boltzmann Machine. *IEEE First International Conference on Neural Networks*, San Diego, 1987.

[38] S. Haykin. *Neural Networks, a comprehensive foundation*. Macmillan Publishing Company, 1994.

[39] D.O. Hebb. *The Organization of Behavior*. Wiley, 1949.

[40] R. Hecht-Nielsen. Application of counterpropagation networks. *Neural networks* 1, 131-140, 1988.

[41] R. Hecht-Nielsen. *Neurocomputing*. Addison Wesley, 1990.

[42] C.G. Hempel. *Aspects of Scientific Explanation*. New York, 1965.

[43] L. Hérault and J.-J. Niez. Neural Networks & Combinatorial Optimization: a Study of NP-complete Graph problems. In *Neural Networks: Advances and Applications*, ed. E. Gelenbe, 165–213, Elsevier, 1991.

[44] J. Hertz, A. Krogh, and R.G. Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley, 1991.

[45] D.R. Hofstadter. *Gödel, Escher, Bach: an eternal golden braid*. Basic Books, New York, 1979.

[46] J.J. Hopfield. Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proceedings of the National Academy of Sciences, USA*, 79, 2554–2558, 1982.

[47] J.J. Hopfield. Neurons with Graded Responses Have Collective Computational Properties Like Those of Two-State Neurons. *Proceedings of the National Academy of Sciences, USA*, 81, 3088–3092, 1984.

[48] J.J. Hopfield and D.W. Tank. "Neural" Computation of Decisions in Optimization Problems. *Biological Cybernetics* 52, 141–152, 1985.

[49] J.J. Hopfield and D.W. Tank. Computing with Neural Circuits: A Model. *Science* 233, 625–633, 1986.

[50] S. Ishii. Chaotic Potts Spin. *Proceedings of the IEEE International Conference on Neural Networks*, 1578–1583, Perth, Western Australia, 1995.

[51] A. Joppe, H.R.A. Cardon, and J.C. Bioch. A Neural Network for Solving the Travelling Salesman Problem on the Basis of City Adjacency in the Tour. In *Proceedings of the International Joint Conference on Neural Networks*, 961–964, San Diego, 1990.

[52] G. Kindervater. Exercises in Parallel Combinatorial Computing. CWI Tract 78, Centrum voor Wiskunde en Informatica, 1991.

[53] M. Kline. *Mathematical Thought from Ancient to Modern Times*. Oxford University Press, 1972.

[54] B. Kosko. *Neural Networks and Fuzzy Systems, a dynamical systems approach to machine intelligence*. Prentice-Hall, 1992.

[55] B. Kosko. *Neural Networks for Signal Processing*. Prentice-Hall, 1992.

[56] J.R. Koza. *Genetic programming, On the programming of computers by means of natural selection*. A Bradford book, 1992.

[57] T.S. Kuhn. *The structure of scientific revolutions*. University of Chicago Press, 1962, 2nd. ed., 1969.

[58] Y. LeCun et al. Learning Algorithms for Classification: A Comparison on Handwritten Digit Recognition. In *Neural Networks: The Statistical Mechanics Perspective*, Proceedings of the CTP-PBSRI Joint Workshop on Theoretical Physics, eds. J.H. Oh, C. Kwon, S. Cho, 261–276, World Scientific, 1995.

[59] Ch. von der Malsburg and D.J. Willshaw. How to label nerve cells so that they can interconnect in an ordered fashion. *Proceedings of the National Academy of Sciences, USA*, 74, 5176–5178, 1977.

[60] W.S. McCulloch and W. Pitts. A logical calculus of ideas imminent in nervous activity. *Bulletin of Mathematical Biophysics* 5, 115–133, 1943.

[61] M. Minsky. *The Society of Mind*. Simon and Schuster, New York, 1986.

[62] M. Minsky and S. Papert. *Perceptrons*. MIT Press, 1969.

[63] A. Newell, J.C. Shaw, and H.A. Simon. Empirical exploration with the logic theory machine: A case study in heuristics. In *Computers and Thought*, McGraw-Hill, 1963.

[64] G. Parisi. *Statistical Field Theory*. Frontiers in Physics, Addison-Wesley, 1988.

[65] Y-K. Park and G. Lee. Applications of Neural Networks in High-Speed Communication Networks. *IEEE Communications* 33-10, 68–74, 1995.

[66] M.C. Pease. *Methods of Matrix Algebra*. Academic Press, 1965.

[67] R. Penrose. *The Emperor's New Mind. Concerning Computers, Mathematics, and The Laws of Physics*. Oxford University Press, 1989.

[68] C. Peterson and J.R. Anderson. A Mean Field Theory Learning Algorithm for Neural Networks. *Complex Systems* 1, 995–1019, 1987.

[69] C. Peterson and B. Söderberg. A New Method for Mapping Optimization Problems onto Neural Networks. *International Journal of Neural Systems* 1, 3–22, 1989.

[70] C. Peterson and B. Söderberg. Artificial Neural Networks and Combinatorial Optimization Problems. To appear in: E.H.L. Aarts and J.K. Lenstra, eds., *Local Search in Combinatorial Optimization*, John Wiley & Sons.

[71] J.C. Platt and A.H. Barr. Constrained Differential Optimization. *Proceedings of the IEEE 1987 NIPS Conference*, 612–621, 1988.

[72] K.R. Popper. *Conjectures and Refutations: the growth of scientific knowledge*, 3rd ed. [rev.]. Routledge & Kegan Paul, 1969.

[73] E. Postma. SCAN: A Neural Model of Covert Attention. *PhD thesis*, Maastricht, 1994.

[74] I. Prigogine and I. Stengers. *Orde uit Chaos, de nieuwe dialoog tussen de mens en de natuur* (translation into Dutch of 'Order out of Chaos'). Bert Bakker, 1993.

[75] D.E. Rumelhart and J.L. McCelland and the PDP Research Group. *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, Volume 1: Foundations, Volume 2: Psychological and Biological Models. MIT Press, 1986.

[76] D. Sherrington. Neural networks: The Spin Glass Approach. In *Mathematical Approaches to Neural Networks*, ed. J.G. Taylor, 261–292, Elsevier, 1993.

[77] P.D. Simic. Statistical Mechanics as the Underlying Theory of 'Elastic' and 'Neural' Optimisations. *Network* 1, 88–103, 1990.

[78] M.W. Simmen. *Neural Network Optimization*. Dept. of Physics Technical Report 92 / 521, PhD Thesis, University of Edinburgh, 1992.

[79] P.K. Simpson. *Artificial Neural Systems, Foundations, Paradigms, Applications, and Implementations*. Pergamon Press, 1990.

[80] J. Starke, N. Kubota, and T. Fukuda. Combinatorial Optimization with Higher Order Neural Networks - Cost Oriented Competing Processes in Flexible Manufactoring Systems. *Proceedings of the IEEE International Conference on Neural Networks*, 1855–1860, Perth, Western Australia, 1995.

[81] T. Takada, K. Sanou, and S. Fukumura. A Neural Network System for Solving an Assortment Problem in the Steel Industry. *Technical Report*, 1–18, Systems Laboratory, Systems Planning and Data Processing Department, Kawasaki Steel Corporation, Japan.

[82] Y. Takefuji. *Neural Network Parallel Computing*. Kluwer Academic Publishers, 1992.

[83] V.M. Tikhomirov. *Fundamental Principles of the Theory of Extremal Problems*. English translation, Wiley & Sons, 1986.

[84] H.L. Trentelman. Representation and Learning in Feedforward Neural Networks. *CWI Quaterly* 6:4, 385–408, Centrum voor Wiskunde en Informatica, 1993.

[85] D.E. Van den Bout and T.K. Miller. Improving the Performance of the Hopfield-Tank Neural Network Through Normalization and Annealing. *Biological Cybernetics* 62, 129–139, 1989.

[86] E. Wacholder, J. Han, and R.C. Mann. A Neural Network Algorithm for the Traveling Salesman Problem. *Biological Cybernetics* 61, 11–19, 1989.

[87] P. Wesley. *Elementaire Wetenschapsleer*. Boom, 1982.

[88] G.V. Wilson and G.S. Pawley. On the Stabilitity of the Travelling Salesman Problem Algorithm of Hopfield and Tank. *Biological Cybernetics* 58, 63–70, 1988.

[89] J.M. Yeomans. *Statistical Mechanics of Phase Transitions*. Oxford University Press, 1992.

[90] A.L. Yuille. Generalized Deformable Models, Statistical Physics, and Matching Problems. *Neural Computation* 2, 1–24, 1990.

Index

Samenvatting

Het onderwerp van onderzoek

In dit proefschrift wordt de relaxatie-dynamica van zogenaamde recurrente neurale netwerken bestudeerd. Meer specifiek richt de analyse zich op Hopfield- en aanverwante recurrente netwerken. Een neuraal netwerk wordt recurrent genoemd, indien de outputs van de neuronen (op een gewogen manier) worden teruggekoppeld naar de inputs. Een belangrijke consequentie van deze architectuur is dat na een 'random initialisatie' een recurrent netwerk in het algemeen niet in evenwicht is. Echter, onder bepaalde voorwaarden blijkt relaxatie naar een evenwichtstoestand van het neurale netwerk spontaan op te treden. Het begrijpen van deze relaxatie en het vinden van de voorwaarden waaronder relaxatie is gegarandeerd voor diverse, zo algemeen mogelijk gedefinieerde klassen van recurrente netwerken is het hoofddoel van deze dissertatie.

Teneinde de gevonden theorie te toetsen (en, naar aanleiding van de uitkomsten, de theorie verder te verbeteren) zijn allerlei relatief eenvoudige simulaties uitgevoerd. Om tevens inzicht te verkrijgen in de toepasbaarheid van de onderzochte modellen zijn daarnaast op het terrein van de combinatorische optimalisering een aantal simulaties uitgevoerd. Bij problemen uit dit vakgebied gaat het in beginsel om het vinden van de 'beste' oplossing uit een grote verzameling van potentiële oplossingen, waarbij de oplossingen veelal moeten voldoen aan een reeks nevenvoorwaarden. De aanwezigheid van deze nevenvoorwaarden heeft de keuze van de diverse netwerkmodellen sterk beïnvloed: de netwerken van hoofdstuk 3,4 en 5 verschillen bovenal in de manier waarop de nevenvoorwaarden worden behandeld. Als laatste is een aantal zogenaamde 'elastische' netwerkmodellen geanalyseerd. Deze neurale netwerken zijn speciaal ontworpen voor het oplossen van het 'handelsreizigerprobleem', misschien wel het beroemdste combinatorische optimaliseringsprobleem.

De resultaten

Na een inleidend hoofdstuk waarin het 'waarom', het 'wat' en het 'hoe' van het onderzoek zijn uiteengezet en gemotiveerd, worden in hoofdstuk 2 de vertrekpunten beschreven. Het betreft hier een overzicht van de relevante netwerkmodellen zoals voorkomend in de literatuur, voorafgegaan door een inleiding in de statistische fysica en gevolgd door een verzameling in de literatuur aangetroffen toepassingen van Hopfieldmodellen. Er is voor gekozen om belangrijke stukken van het gebruikte wiskundig gereedschap op te nemen in de bijlagen A t/m D.

De weergave van het nieuw gevondene start in hoofdstuk 3 met de analyse van het klassieke stochastische Hopfieldmodel. Een statistisch-fysische analyse levert twee 'mean field' approximaties op voor de vrije energie van het systeem, waarvan de stationaire punten exact samenvallen. Indien de sigmoïde gekozen wordt als de overdrachtsfunktie in de neuronen, valt één van deze twee approximaties precies samen met de uitdrukking van de energie van het klassieke continue Hopfieldmodel: continue Hopfieldmodellen hebben volgens deze zienswijze een statistisch-mechanische interpretatie. Bij die veelgebruikte keuze van de sigmoïde als overdrachtsfunktie kan het effect van de integraal, zoals voorkomend in de uitdrukking van de energie van het continue model, nauwkeurig worden geanalyseerd en kunnen enige misverstanden, zoals aangetroffen in de literatuur, worden rechtgezet. De andere approximatie van de vrije energie is de sleutel tot het vinden van een zeer algemene energiefunktie van het oorspronkelijke continue Hopfieldmodel. Deze funktie, die een uitdrukking is in zowel de input- als de outputwaarde van alle neuronen, blijkt de toestand van een Hopfieldnetwerk volledig te beschrijven. De extreme waarden van deze funktie corresponderen precies met het *complete* stelsel van evenwichtsvoorwaarden van het oorspronkelijke continue Hopfieldmodel.

Naast expressies van de vrije energie worden de stabiliteitsvoorwaarden van de bijbehorende continue bewegingsvergelijkingen besproken. Het hoofdstuk wordt afgesloten met een paar relatief eenvoudige experimenten gebruikmakend van een aantal van die bewegingsvergelijkingen. Allereerst wordt het n-toren probleem opgelost, onder toepassing van de zogeheten penalty methode. Daarnaast wordt getoond hoe het toevoegen van thermische energie aan het continue model het vinden van de globale (i.p.v. een locale) oplossing kan bevorderen. Het betreft hier de aanpak met behulp van 'mean field annealing', welke gezien kan worden als een deterministische approximatie van de bekende aanpak met behulp van 'simulated annealing'.

In hoofdstuk 4 wordt als eerste een analyse gedaan van stochastische Hopfieldnetwerken die onderworpen zijn aan een bepaalde, eenvoudige nevenvoorwaarde: de betreffende nevenvoorwaarde wordt 'ingebouwd' in het neurale netwerk. Eenzelfde mean field benadering blijkt mogelijk te zijn als welke in hoofdstuk 3 is uitgevoerd. De eigenschappen van de twee in dit hoofdstuk gepresenteerde approximaties van de vrije energie zijn wat ingewikkelder dan die uit het vorige hoofdstuk, maar vertonen verder een grote gelijkenis. Bovendien kan eenzelfde generalisatie worden uitgevoerd. Deze levert een derde vrije energiefunktie op waarvan de extrema wederom precies corresponderen met de complete verzameling van evenwichtsvoorwaarden van het (nu aan nevenvoorwaaarden onderworpen) neurale netwerk.

Een zeer belangrijke stap wordt vervolgens gezet door de generalisatie verder door te trekken. In de eerste plaats wordt de wiskundige beschrijving zodanig verruimd dat 'willekeurige' nevenvoorwaarden in het neurale netwerk kunnen worden ingebouwd. De beschrijving wordt nog algemener omdat vervolgens ook bijna willekeurige kostenfunkties (i.p.v. louter kwadratische) worden toegelaten. Stap voor stap passeren steeds algemenere expressies voor de vrije energie de revue en worden de stabiliteitsvoorwaarden van diverse bijbehorende syste-

men van bewegingsvergelijkingen besproken. Dit levert uiteindelijk het *meest algemene raamwerk* van continue Hopfieldmodellen op. Ook dit hoofdstuk eindigt weer met de resultaten van enige uitgevoerde simulaties, waarbij de nevenvoorwaarden zoveel mogelijk zijn ingebouwd in het neurale netwerk. Het gevonden stabiliteitsgedrag bij deze experimenten is in overeenstemming met de theoretische verwachtingen en een aantal problemen is correct opgelost. Daarnaast is een belangrijke experimentele uitkomst dat bepaalde, in het netwerk ingebouwde, nevenvoorwaarden de gebruikelijke statistisch-mechanische interpretatie van continue Hopfieldmodellen teniet doen, waardoor een ándere dan de oorspronkelijke oplossing van het gegeven probleem gevonden wordt. Dit maakt duidelijk dat het algemene raamwerk zekere beperkingen kent, welke nog om nader onderzoek vragen. Voor het overige is het van belang te vermelden dat het inbouwen van nevenvoorwaarden de convergentiesnelheid van de bewegingsvergelijkingen ten zeerste blijkt te bevorderen.

In hoofdstuk 5 wordt een derde methode besproken voor het behandelen van de nevenvoorwaarden. Gebruikmakend van multiplicatoren van Lagrange wordt het zogenaamde Hopfield-Lagrangemodel gedefinieerd. Naar analogie van het bekende fysisch model van een veer-massa-systeem worden stabiliteitscondities van het Hopfield-Lagrangemodel afgeleid. Dit blijkt te kunnen bij toepassing van zowel het Hopfieldmodel zonder nevenvoorwaarden als van dat met 'willekeurige' nevenvoorwaarden en 'willekeurige' kostenfunkties zoals geanalyseerd in hoofdstuk 4. Evenwel, deze condities zijn in diverse gevallen moeilijk analytisch verifieerbaar. Voorts wordt de werking van kwadratisch geformuleerde nevenvoorwaarden bij toepassing van het Hopfield-Lagrangemodel ontmaskerd. Het model blijkt bij gebruik van dat type nevenvoorwaarden te degenereren tot wat is genoemd een *dynamische penalty methode*. De multipliers krijgen bij het niet vervuld zijn van de nevenvoorwaarden steeds grotere waarden en de multiplierermen gaan zich gedragen als penaltytermen (de multipliers vallen precies samen met de gewichten van de penaltytermen). De waarde van elke multiplier blijft toenemen totdat het systeem een toestand aanneemt waarbij aan de bijbehorende nevenvoorwaarde is voldaan. Als op de duur aan alle nevenvoorwaarden is voldaan, treedt er een verschijnsel op dat sterk lijkt op dat van een fasetransitie in de statistische fysica. Daarbij wordt het systeem plotseling stabiel.

Met beide genoemde Hopfield-Lagrangemodellen (zonder en met nevenvoorwaarden) zijn allerlei experimenten uitgevoerd te beginnen met een aantal eenvoudige en eindigend met een moeilijke, te weten, het handelsreizigersprobleem. De experimenteel gevonden stabiliteitseigenschappen zijn in overeenstemming met hetgeen theoretisch verwacht werd. Ook stemmen de simulatieresultaten overeen met de intuïtief te verwachten stelregel dat hoe moeilijker het probleem is, des te langer de rekentijd en-of des te slechter de kwaliteit is van de aangetroffen oplossingen. Een andere bevinding is dat Hopfield-Lagrangemodellen met veel multipliers het veel beter doen dan die met weinig (waarbij meerdere nevenvoorwaarden bij elkaar worden genomen). Een verrassing is het fenomeen dat pure Hopfield-Lagrangeformuleringen van het handelsreizigersprobleem betere oplossingen opleveren dan die waarbij een deel van de nevenvoorwaarden wordt ingebouwd en de rest wordt aangepakt met multipliers.

In hoofdstuk 6 wordt onderzoek gedaan naar zogenaamde 'elastische' netwerken. Begonnen wordt met het klassieke elastische netwerk voor het oplossen van het handelsreizigerprobleem. Aangetoond wordt dat ook dit netwerk beschouwd kan worden als een toepassing van de dynamische penalty methode, waarbij onder invloed van verlaging van de temperatuur de gewichten van de penalty termen langzaam afnemen. Bovendien veranderen ze van vorm. De analyse laat verder zien dat een aantal in de literatuur voorkomende opvattingen (over de relatie van het elastische net met bepaalde Hopfieldnetwerken) onjuist is. Voorts worden twee alternatieve elastische netwerken geanalyseerd die een correcte afstandsmaat toepassen t.a.v. de afstanden tussen de opeenvolgende punten in het elastische netwerk. Voor kleine probleeminstanties doen deze netwerken het beter, voor grotere netwerken iets slechter dan het oorspronkelijke elastische netwerk.

Tot slot

In het laatste hoofdstuk worden de resultaten gegroepeerd gepresenteerd en bediscussieerd. Daarbij komt naar voren dat het gevonden 'meest algemene raamwerk' van continue Hopfieldmodellen (met bijbehorende theorema's) en de invoering van het begrip 'dynamische penalty methode' (met bijbehorende voorbeelden) zeer belangrijke elementen zijn van deze dissertatie. Voorts wordt geconcludeerd dat de gegenereerde simulatieresultaten (samen met die in de literatuur aangetroffen zijn) doen vermoeden dat Hopfieldnetwerken uitstekend geschikt zijn voor het vinden van een oplossing van combinatorische problemen waarbij een oplossing wordt gezocht die aan allerlei nevenvoorwaarden moet voldoen, maar waarbij er geen kostenfunktie geminimaliseerd hoeft te worden. Echter, indien er naast nevenvoorwaarden ook optimalisatie in het spel is, kan het vinden van een kwalitatief goede oplossing niet zonder meer gegarandeerd worden. Het toepassen van zoveel mogelijk domeinkennis lijkt voor moeilijke combinatorische optimaliseringsproblemen een onmisbaar ingrediënt.

Het hoofdstuk wordt besloten met een reeks van aanbevelingen voor verder onderzoek. Een aantal theoretische vraagstukken dat is blijven liggen kan alsnog worden aangepakt. Het is o.a. interessant om te onderzoeken welke klassen van nevenvoorwaarden op een zodanige wijze kunnen worden ingebouwd dat een zinvolle statistisch-mechanische interpretatie van de resultaten mogelijk is. Zeker intrigerend is de vraag of het meest algemene raamwerk ook geldig is voor stochastische Hopfieldnetwerken. Voorts lijkt het verstandig om de in dit proefschrift naar voren gebrachte theoretische resultaten te gaan toepassen op allerlei praktijkproblemen. Daarbij moet waarschijnlijk heel wat ervaring worden opgedaan om voldoende inzicht en intuïtie op te bouwen voor daadwerkelijk succes: het maken van allerlei juiste keuzes (bijvoorbeeld t.a.v. de architectuur van het netwerk, het schema van verlagen van de temperatuur, de manier van afbeelden van het probleem op het neurale net, de wijze van initialisatie van het netwerk) is zeker geen triviale zaak. Deze praktijkgerichte stap lijkt ook zinvol ten einde toekomstig theoretisch onderzoek op dit specifieke terrein van recurrente neurale netwerken op directe en indirecte wijze te ondersteunen.

Curriculum vitae

Jan van den Berg werd op 24 juni 1951 geboren te Rotterdam. Zijn jeugd bracht hij door in Dordrecht. Na het doorlopen van het Gemeentelijk Lyceum (met afsluitend diploma gymnasium-β) ging hij in 1970 aan de Technische Universiteit Delft electrotechniek studeren. In het tweede jaar stapte hij over naar wiskunde. In de periode 1972-1973 werd de studie een jaar onderbroken voor werkzaamheden in de lokale en landelijke studentenpolitiek. Na voltooiing van zijn afstudeerwerk inzake het gebruik van de 'eindige elementen methode' bij variationeel geformuleerde problemen uit de sterkteleer, behaalde hij in 1977 het doctoraal examen wiskunde en verkreeg hij het diploma van wiskundig ingenieur.

Vervolgens was hij in de periode 1977-1981 als docent wis-en natuurkunde verbonden aan het (toenmalige) Dr Struycken Instituut, een hbo-instelling voor de opleiding van medisch analisten. In de periode 1981-1983 werkte hij in dienst van de regering van Mozambique en gaf hij wis- en natuurkundelessen aan de middelbare school van Nampula. Na terugkeer in Nederland belandde hij op het IHBO-Eindhoven waar hij o.a. wiskunde en informatica doceerde binnen de studierichtingen der electrotechniek en informatica. In 1989 vond de overstap plaats naar de vakgroep informatica van de economische faculteit van de Erasmus Universiteit te Rotterdam. Zijn onderwijswerkzaamheden lagen (en liggen) vooral op het terrein van computerarchitectuur, operating systemen en computernetwerken. Zijn onderzoek op het gebied van neurale netwerken binnen de sectie kunstmatige intelligentie startte in het najaar van 1992.