BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV RADIOELEKTRONIKY

PROBABILISTIC NEURAL NETWORKS FOR SPECIAL TASKS IN ELECTROMAGNETICS

PRAVDĚPODOBNOSTNÍ NEURONOVÉ SÍTĚ PRO SPECIÁLNÍ ÚLOHY V ELEKTROMAGNETISMU

DISSERTATION THESIS

DISERTAČNÍ PRÁCE

AUTHOR: AUTOR PRÁCE	Ing. Vlastimil Koudelka
SUPERVISOR: VEDOUCÍ PRÁCE	prof. Dr. Ing. Zbyněk Raida

BRNO, 2014

ACKNOWLEDGEMENTS

The work described in this thesis has received funding from EC FP7 under grant no. 205294 (the HIRF SE project).
Further financing was provided by the Czech Ministry of Education (the grant no. 7R09008).

The described research was performed in laboratories supported by the SIX project; the registration number CZ.1.05/2.1.00/03.0072, the operational program Research and Development for Innovation.

A support of the project CZ.1.07/2.3.00/20.0007 Wireless Communication Teams financed by the operational program Education for Competitiveness is also gratefully acknowledged.

ABSTRACT

The thesis deals with behavioural modelling techniques capable solving special tasks in electromagnetics which can be formulated as approximation, classification, probability estimation, and combinatorial optimization problems. Concept of the work lies in applying a probabilistic approach to behavioural modelling. Examined methods address two general problems in machine learning and combinatorial optimization: ”bias vs. variance dilemma” and NP computational complexity. The Boltzmann machine is employed to simplify a complex impedance network. The Parzen window is regularized using the Bayesian strategy for obtaining a model selection criterion for probabilistic and general regression neural networks.

KEYWORDS

behavioural modelling, Boltzmann machine, combinatorial optimization, impedance network simplification, Bayesian regularization, probabilistic neural network, general regression neural network

ABSTRAKT

Tato práce pojednává o technikách behaviorálního modelování pro speciální úlohy v elektromagnetismu, které je možno formulovat jako problém aproximace, klasifikace, odhadu hustoty pravděpodobnosti nebo kombinatorické optimalizace. Zkoumané methody se dotýkají dvou základních problémů ze strojového učení a combinatorické optimalizace: ”bias vs. variance dilema” a NP výpočetní komplexity. Boltzmanův stroj je v práci navržen ke zjednodušování komplexních impedančních sítí. Bayesovský přístup ke strojovému učení je upraven pro regularizaci Parzenova okna se snahou o vytvoření obecného kritéria pro regularizaci pravděpodobnostní a regresní neuronové sítě.

KLÍČOVÁ SLOVA

behavioralní modelování, Boltzmanův stroj, kombinatorická optimalizace, zjednodušování impedanční sítě, Bayesovská regularizace, pravděpodobnostní neuronová síť, regresní neuronová síť

KOUDELKA, Vlastimil *Probabilistic neural networks for special tasks in electromagnetics*: doctoral thesis. Brno: Brno University of Technology, Faculty of Electrical Engineering and Communication, Department of Radio-electronics, 2014. 91 p. Supervised by prof. Dr. Ing. Zbyněk Raida,

DECLARATION

I declare that I have elaborated my doctoral thesis on the theme of "Probabilistic neural networks for special tasks in electromagnetics" independently, under the supervision of the doctoral thesis supervisor and with the use of technical literature and other sources of information which are all quoted in the thesis and detailed in the list of literature at the end of the thesis.

As the author of the doctoral thesis I furthermore declare that, concerning the creation of this doctoral thesis, I have not infringed any copyright. In particular, I have not unlawfully encroached on anyone's personal copyright and I am fully aware of the consequences in the case of breaking Regulation § 11 and the following of the Copyright Act No 121/2000 Vol., including the possible consequences of criminal law resulted from Regulation § 152 of Criminal Act No 140/1961 Vol.

Brno . . . .

(author's signature)

ACKNOWLEDGEMENT

I wish to thank prof. Dr. Ing. Zbynek Raida who served as my supervisor and also encouraged and challenged me throughout my academic program.

Brno . . . .

(author's signature)

Abbreviations

ANN	Artificial Neural Network
BM	Boltzmann Machine
CV	Cross-Validation
DT	Decision Tree
DOF	Degrees Of Freedom
EM	Electro Magnetic
EMC	Electro-Magnetic Compatibility
EV	Evidence
FCM	Fuzzy C-means Method
GA	Genetic Algorithm
GRNN	General Regression Neural Network
HIRF-SE	High Intensity Radiated Field Synthetic Environment
HM	Hopfield Machine
HMM	Hidden Markov Model
IN	Impedance Network
KDE	Kernel Density Estimation
MLP	Multi Layered Perceptron
MSE	Mean Squared Error
NP	Non-deterministic Polynomial time
LVQ	Learning Vector Quantization
P	Polynomial Time
PDF	Probability Density Function
PNN	Probabilistic Neural Network
PSO	Particle Swarm Optimization
RBF	Radial Basis Function network
SA	Simulated Annealing

SSE	Sum Squared Error
TE	Thermal Equilibrium
TM	Touring Machine
TSP	Travelling Salesman Problem

Symbols

$\boldsymbol{a}(L,T)$	quasi-stationary distribution
A	acceptance probability
b_i	the i–th neuron bias
C	consensus function
E	system energy function
$\langle E \rangle$	mean value of the system energy
f_k	frequency assigned to the k–th impedance sample
G	generation probability
$N_0(T)$	partition function
O	objective function
$\tilde{O}_i$	the i–th pseudo-objective function
$\boldsymbol{P}$	transition matrix
$\boldsymbol{q}(T)$	stationary distribution
T	temperature - control parameter
t_M	a maximal running time of the Touring machine
$w_{i,j}$	synaptic connection weight
$\boldsymbol{X}$	state vector
Y_{eq}	equivalent admittance of an impedance network
$\tilde{Y}_{eq}$	equivalent admittance of reduced impedance network
Z_{eq}	equivalent impedance of an impedance network
$\tilde{Z}_{eq}$	equivalent impedance of reduced impedance network
α	reduction coefficient
$\Gamma_{\Omega_{opt}}(k)$	function identifying optimal solutions
ϵ	maximal quasi-stationary distribution deviation
$\Im\{Z_{i,j}(f_k)\}$	imaginary part of the k–the frequency sample of the i–th impedance network element characterized by the j–th impedance pattern (access)

$\Re\{Z_{i,j}(f_k)\}$	real part of the k–the frequency sample of the i–th impedance network element characterized by the j–th impedance pattern (access)
σ	smoothing parameter
χ	acceptance ratio
Ω	set of all Boltzmann machine states
Ω_{opt}	set of all optimal states
$\mathcal{D}$	set of training patterns

Contents

1 Introduction **12**

1.1 State of the art . . . 13

1.1.1 Combinatorial computation . . . 13

1.1.2 Probabilistic neural network . . . 16

1.2 Objectives . . . 18

2 Mapping problem **19**

2.1 Mapping strategy . . . 19

2.2 Mapping procedure . . . 22

2.3 Additional criteria . . . 26

2.4 Intuitive aspect . . . 28

2.5 Conclusions on the mapping problem . . . 28

3 Neural dynamics **30**

3.1 Boltzmann machine characteristics . . . 30

3.2 Asymptotic Convergence . . . 31

3.3 Finite-Time approximation . . . 34

3.4 Unlimited parallelism . . . 42

3.5 Conclusions on neural dynamics . . . 45

4 Probabilistic neural networks **46**

4.1 PDF estimation . . . 46

4.2 Bias vs. Variance . . . 48

4.3 Likelihood criterion . . . 52

4.4 Bayesian strategy . . . 55

4.5 Bayesian procedure . . . 58

4.6 Conclusions on probabilistic neural networks . . . 68

5 Numerical validation **69**

5.1 Validation strategy . . . 69

5.2 Boltzmann machine simplifier . . . 70

5.2.1 Validation test-case . . . 70

5.2.2 Performance . . . 72

5.2.3 Reliability . . . 74

5.2.4 Parallel emulation . . . 76

5.3 Probabilistic neural networks . . . 76

5.3.1 A structural change of KDE . . . 76

5.3.2 Classification task . . . 78

5.3.3 Approximation task . . . 80

5.4 Discussion on numerical results . . . 82

6 Conclusion **83**

Bibliography **84**

A Probabilistic neural networks **89**

A.1 Mean squared error decomposition . . . 89

A.2 Likelihood decomposition . . . 90

1 Introduction

At present, virtual electro-magnetic compatibility (EMC) testing becomes to be seriously employed by EMC community. Commercial aspects force developers to simulate more and more complex structures like cars or aircrafts. On the other hand, extreme complexities of such systems obviously result in unbearable high computation demands. A possible solution to the problem lies in dividing given super-complex domain onto the several coupled sub-domains, each one worked out by a particular efficient method. Such sub-domains can be represented by problems like electromagnetic (EM) wave incidence on an aircraft, EM coupling between external environment and aircraft fuselage, EM field distribution inside the fuselage, calculation of induced currents in cables, cable structure modelling, or electronic equipment simulation on circuit level.

This approach makes possible to address the crucial sub-domains and search alternative solutions which would not be applicable on the whole super-complex domain. Behavioural modelling takes an important part in this growing approach due to its versatile applicability and efficiency. A behavioural model can learn behaviour of a particular sub-domain (e.g. shielding effectiveness of composite material), can act as a coupling device (e.g. estimation of the transfer function of aircraft fuselage), enables to estimate the probability of occurrence of an observed quantity (e.g. dangerous level of field intensity), can classify material structures, or systematically simplify a complex system according to its external behaviour (e.g. impedance network simplification). All mentioned tasks can be formulated as one of the following problems: approximation, probability density estimation, classification, and combinatorial optimization. This kind of special EM tasks can be performed using a probabilistic approach to machine learning and combinatorial simplification, what is the scope of the thesis.

Various applications of the neural networks and combinatorial optimization in EM have been published by the author in [1] - [4]. The present work is focused rather on extending theoretical background of the published approaches than illustrating all possible applications of the neural networks. The concept of the thesis lies in applying the probabilistic approach to behavioural modelling. The crucial questions like "bias vs. variance dilemma" in machine learning or non-deterministic polynomial time (**NP**) computational complexity in combinatorial simplification are addressed, discussed and possible solutions to the problems are proposed.

Basically, the thesis can be divided onto two particular parts. The first part can be assigned to the circuit level domain. It deals with simplification of a complex impedance network using a stochastic neural network called the Boltzmann machine (BM). The main purpose is to simplify an equivalent circuit and simultaneously maintain its external behaviour (e.g. equivalent impedance). An impedance network simplification problem is analytically mapped onto the BM which is capable to solve the combinatorial problem very efficiently. The motivation is to map more versatile equivalent circuits to be simplified by the BM.

The second part covers approximation, classification and kernel density estimation problems. Thus, it should cover the reminder of the special tasks. In all of the mentioned problems the ”bias vs. variance dilemma” is addressed since it strongly influence reliability (generalization) of the models. More specifically, the Parzen window is under the scope of the second part of the thesis since it is the core of the probabilistic (PNN) and general regression neural networks (GRNN). Developed model selection criterion based on Bayesian framework incorporates model fitting, generalization, regularization, and structural change in the probabilistic neural model.

1.1 State of the art

Applying artificial neural networks (ANNs) in EMC issues is narrowly focused discipline exploiting high computational performances of parallel systems which are built in accordance with the structure of the human brain. In open literature, one can not find many publications related to this topic. However, exploitation of ANNs in the EMC has been already described in several papers. In [6] and [7], the multi-layer perceptron (MLP) was used to extrapolate signals calculated by a finite difference time domain method in order to reduce computing costs of the EMC simulations. A method based on MLP used for prediction of electromagnetic fields radiated by generators of electrostatic discharges was presented in [8]. In [9], inverse neural modelling was used for identification of the parameters of metallic walls, a radial basis function (RBF) network was applied in order to solve the inverse problem.

Publications pointed above showed that neural networks are capable to solve real EMC problems formulated as an extrapolation, approximation or inverse problem. However, regarding to the probability based neural networks and their applications the key work has been done in different research fields.

1.1.1 Combinatorial computation

As mentioned in the introduction, impedance network simplification lies in searching a reduced network exhibiting similar behaviour like the original one. The idea is based on two following assumptions:

- Within the impedance network (a complex EM structure), several dominating elements mainly influence the total current flowing through the network (external behaviour) so that particular minor elements can be omitted.

or

- Pairs of elements influencing the total current can compensate each other so that the pairs can be omitted. As well as pairs, more complex impedance network (IN) substructures can also be omitted with negligible change in the total current.

Assumptions pointed above imply that impedance network simplification is an combinatorial problem. As will be discussed later, reducing IN complexity belongs to a class of **NP** (non-deterministic polynomial time) problems. Thus, it is essential to address the last main contributions in combinatorial optimization.

Definitions of polynomial time (**P**) and non-deterministic polynomial time (**NP**) classes can be found in [10] where the **P** =? **NP** question is also formulated. In simple words, the **P** class consists of the problems which can be solved by Turing deterministic machine (TM) in polynomial time (see Appendix of [10]). A problem is solved in polynomial time if a running time of the Turing machine t_M satisfies

$$t_M(n) \leq n^k + k \tag{1.1}$$

for arbitrary k, where the n is the number representing the problem size (e.g. the number of the elements inside the IN). The **NP** class consist of the problems which can be solved in polynomial time by non-deretministic machine (e.g. Boltzmann machine).

Basically, there are three possible ways to tackle the **NP** complexity [11]:

- The enumerative method is the most trivial solution to a combinatorial problem and consumes a huge amount of computational time. On the other hand, it always returns exact solution to the problem. In the fact, the enumerative method is the extreme case of local search approach described in [12].
- An approximation approach lies in substituting the original problem by similar one which can be solved in polynomial time. Suitable algorithms can guarantee good agreement between approximated solution and the original one. The main disadvantage is that each approximation algorithm specialises on a particular problem (e.g. Travelling salesman problem) and it is no longer valid for other problems. Theoretical background for approximation algorithms can be found in [13].
- The last way is to employ a non-deterministic algorithm based on local search technique. Typical and well known members of this class are genetic algorithms (GA) [14] and simulated annealing (SA) [15]. Such stochastic algorithms asymptotically approach the global optimum. The main advantage of the stochastic approaches is their versatility and wide applicability. We can benefit from this quality and obtain the method solving various simplification problem instances.

The first neural network solving NP-complete problem called travelling salesman (TSP), formally “the second order assignment problem”, was invented by Hopfield and Tank [16] in 1985. Hopfield and Tank proposed how to map constrained optimization problem onto the Hopfield neural network (HNN) also called the Hopfield machine (HM). Appropriate inspection of HM performance was done by Wilson and Pawley [17] who found that the deterministic HM is not suitable for problems having a real-world scale. Constrains of HM are caused by two following statements:

- The HM searches a solution space in a local gradient manner which causes that the HM can stack in local optimum with poor performance.
- If a combinatorial problem scale reaches an upper limit of the HM, energy function degrades and several artificial minima occur. The HM energy function no longer substitutes the objective function of the combinatorial problem. In [18] the HM ineffectiveness was explained in sense of aliasing.

Regarding to the first point, several improvements of the original HM has been proposed. In [19] and [20] external noise is injected into the network in order to escape from the local minima. A neuron model with chaotic dynamics was invented in [21] to improve the original deterministic concept. Although the performance of HM solving TSP was increased, parameters of additional noise and its influence on the network convergence was not proved properly. The second problem stated above always occurs due to the basic structure of HM (see [18]).

While the HM is naturally deterministic algorithm (each state of HM is followed by exactly determined next state) the Boltzmann machine (BM) is naturally stochastic neural network (each state can be followed by a finite number of next states having various probabilities of transitions). In contrast with HM, the BM inspired by statistical physics settles in global minimum of system energy function E if the BM is properly simulated. Naturally, stochastic BM doesn't need any external noise to be injected and the network convergence can be clearly investigated. It is due to stochastic neuron model used in BM architecture:

$$\boldsymbol{P}\{k(i) = 1 | k(i) = 0\} = \frac{1}{1 + exp(-\frac{\Delta E}{T})} \tag{1.2}$$

Here the $\boldsymbol{P}\{k(i) = 1 | k(i) = 0\}$ means the probability of transition if the i-th neuron output is changed from the value 0 to the value 1, while ΔE denotes related the change in system energy E. Symbol T denotes the system temperature which is taken form statistical mechanics [22]. A temperature T can be understood as a control parameter of logistic function providing probabilistic activation function of the BM neuron.

The key work related to the BM in combinatorial optimization have been done by Aarts and Korst. The most important publication [15] proofs convergence of the SA method and deals with convergence of BM with optimal cooling schedule. In [23] Aarts and Korst proposed the BM as a parallel variant of SA [24] which is naturally sequential process. In [24] the connection between statistical mechanics and combinatorial optimization was clarified.

The BM was applied to solve TSP in [25]. Further practical investigation of BM was published in [26] where the BM was used to solve block placement problem, which belongs to **NP**-complete class. In [27] the BM was employed to solve another **NP**-complete problem: the problem of satisfiability [28].

Dealing with BM optimizers several following issues should be addressed: the mapping problem, the cooling schedule, and the updating scheme problem. Since the IN simplification task is to be solved by the Boltzmann machine the mentioned issues have to be addressed. The objective function of simplification task O has to be formulated in terms of the BM energy function E [29]. The question how to map simplification problem onto the network has to be answered and cooling schedule for annealing process [15] has to be properly chosen. Finally, the updating scheme should be investigated to guarantee the convergence of BM. In the key work [15], the issue of BM parallelism is briefly discussed. It was proposed to distinguish between limited and unlimited parallelism. The conclusion of [15] still motivates to further investigation of BM convergence properties [30].

1.1.2 Probabilistic neural network

The probabilistic neural network (PNN) originally invented by D. F. Specht [31] in 1990 has been recently investigated in power delivery issues. In [32], [33], and [34] the PNN was used directly and PNN smoothing parameter was experimentally adjusted. In [35] the smoothing parameter was adjusted by particle swarm optimization (PSO) technique improving the PNN accuracy. In [36] fuzzy c-means (FCM) clustering method was used to determine a finite number of desired classes for PNN model definition. PNNs where compared with MLP and RBF neural networks which exhibit lower accuracy in these applications. Further investigation of the PNN performance was done in recently published paper [37], where the neural network approach was compared with Hidden Markov Model-based (HMM) method and decision tree (DT) classifier.

Mentioned applications usually implement the original concept of PNN proposed by Specht. The original PNN classifier operates on the basis of the Bayesian decision strategy (an average risk of misclassification is minimized). Since probability density functions (PDFs) of the classes are unknown, probabilistic functions have to be estimated from the training set. The core of PNN is the Parzen PDF estimator known as the Parzen window.

The main disadvantage of methods employing the original approach is a huge number of the Gaussian kernels (hidden neurons) used for the PDF estimation which causes computational inefficiency. Dealing with this issue, several papers in neural journals were published. The letter [38] direcly responding to the [31] emphasized mentioned disadvantage of the original PNN and proposed possible solution employing cluster technique. The learning vector quantization (LVQ) used in [38] requires preliminary definition of desired number of clusters. The optimal number of clusters remains unknow as well as the optimal value of smoothing parameter. In [39] advanced training technique defining both the number of pattern units (hidden neurons) and the optimal smoothing parameter was proposed and directly compared with [38]. The algorithm developed in [39] exhibited better classification and simplification performances in comparison with [38]. The genetic algorithm (GA) was employed to adjust the smoothing parameter. Obviously, employing global optimization methods can dramatically increase computation time required for precise network training. The impact of GA on the total computational time was not investigated in [39]. The most promising contribution [40] exploited deterministic approach to clustering problem proposed by Berthold and Diamong in [41] to avoid computationally expensive global optimization. The algorithm utilized adjusting each Gaussian kernel variance individually to continually cover the whole input space by reduced number of hidden neurons (Gaussian kernels). On the other hand, papers [40] and [41] consider symmetric multi-scale Gaussian kernels which disabled an adaptive normalization of the input-space proposed by Specht in [42].

To conclude this section, recent publications [32]-[37] are mostly aimed to applying the PNNs. However, one can find a lack of advance methods proposed in many neural journals pointed above. Consequently, the developed applied models are not validated in a proper way. In most cases the number of testing patterns is smaller than the number of the training ones which does not allow a proper model validation.

On the other hand, theoretical papers [38]-[42] deal with specific problems related to the PNN separately. Each contribution proposes its original point of view which usually doesn't cover the functionalities of the other papers (i.e. methods [41] and [42]).

It is essential to develop a robust training approach to the PNN providing probabilistic modelling of complex electromagnetic (EM) structures. Investigated methodology should also examine a validation procedure of developed models and discuss computation demands of developed training an selection algorithms. We argue that a clustering technique has to be connected with sufficient kernel width estimator and visa-versa since both of the tasks deal with model complexity and contributes to the model bias and variance [44] in the similar way. For such a complex approach we can see a lack of suitable criterion which can evaluate models having various numbers of the neurons and different kernel widths.

1.2 Objectives

The purpose of this section is to formulate the main work objectives and also to summarize all challenges resulting from the investigated problem.

Dealing with simplification task, the first objective results from the fact that there is no general method how to map combinatorial problems onto BM.

Objective 1

The impedance network simplification has to be formulated in accordance to the Boltzmann machine energy function E and appropriate mapping method $O \to E$ has to be developed.

Generally, the cooling schedule has to be chosen properly according to a particular problem. Since the BM was intended for efficient simulation of annealing process known from statistical mechanics, the parallel implementation influencing the BM dynamics should be studied.

Objective 2

The methodology of cooling schedule definition and suitable updating scheme have to be developed to guarantee the BM convergence to the closely optimal solution.

Regarding to probabilistic modelling, it is essential to decrease the number of hidden neurons and increase the performance both the PNN and GRNN models (see section 1.1.2). A clustering technique has to be connected with sufficient kernel width estimator and visa-versa since both of the tasks deal with model complexity and contribute to the model bias and variance in the similar way.

Objective 3

An accurate PDF estimation has to be implemented in PNN and GRNN to increase their accuracy and computational efficiency.

The last objective is to evaluate probability-based neural networks in a proper way.

Objective 4

The proper validation technique has to be developed to measure qualities of stochastic neural optimizers and probabilistic neural approaches.

2 Mapping problem

2.1 Mapping strategy

As mentioned above, a combinatorial problem has to be mapped onto the neural network so that the neural network state related to the highest system energy E equals the problem solution. Note that the Boltzmann machine maximizes its energy function and its maximum has to be related to the minimum of the objective function. One should distinguish between the training of a feed forward neural network and a mapping problem. In Figure 2.1 a scheme of an impedance network simplification task (top) and a scheme of the BM (bottom) can be seen.

Firstly, the combinatorial problem in hand has to be defined. Each cable from the bundle in Figure 2.1 is characterized by input impedance of a circuit (access) interfacing the aircraft electronic equipment. Let us suppose, that these impedances $Z_{i,j}$ mainly influence the total current I_c flowing through the bundle. Than, each i–th cable is connected with one of the accesses identified by symbol j in Figure 2.1. In this approximative form, input impedances interfacing the bundle (accesses) form impedance network (IN) which describes a particular cable bundle configuration so that the cable bundle simplification can be formulated as an impedance network simplification problem (see Figure 2.1):

$$O(Z_{eq}, \tilde{Z}_{eq}, r, \alpha) = \alpha r + SSE(Z_{eq}, \tilde{Z}_{eq}). \tag{2.3}$$

Here, we have to minimize a number of elements r appearing in the reduced IN characterized by an equivalent impedance $\tilde{Z}_{eq}$. Simultaneously, a sum squared error (SSE) between equivalent impedances Z_{eq} and $\tilde{Z}_{eq}$ representing original and reduced networks has to be minimized. Z_{eq} and $\tilde{Z}_{eq}$ are vectors each one containing K frequency samples of equivalent impedances whose derivation is described later. Since these two criteria are obviously conflicting, the coefficient α is proposed to balance the objectives.

Secondly, the BM machine is required to have exactly the same number of neurons L as the number of elements in the IN. Each input impedance $Z_{i,j}$ is represented by an individual binary neuron in the BM. If the neuron is “on”, the related element appears in a simplified bundle; if the neuron is “off”, the related element is omitted. This is the proposal how to represent an IN using the Boltzmann machine.

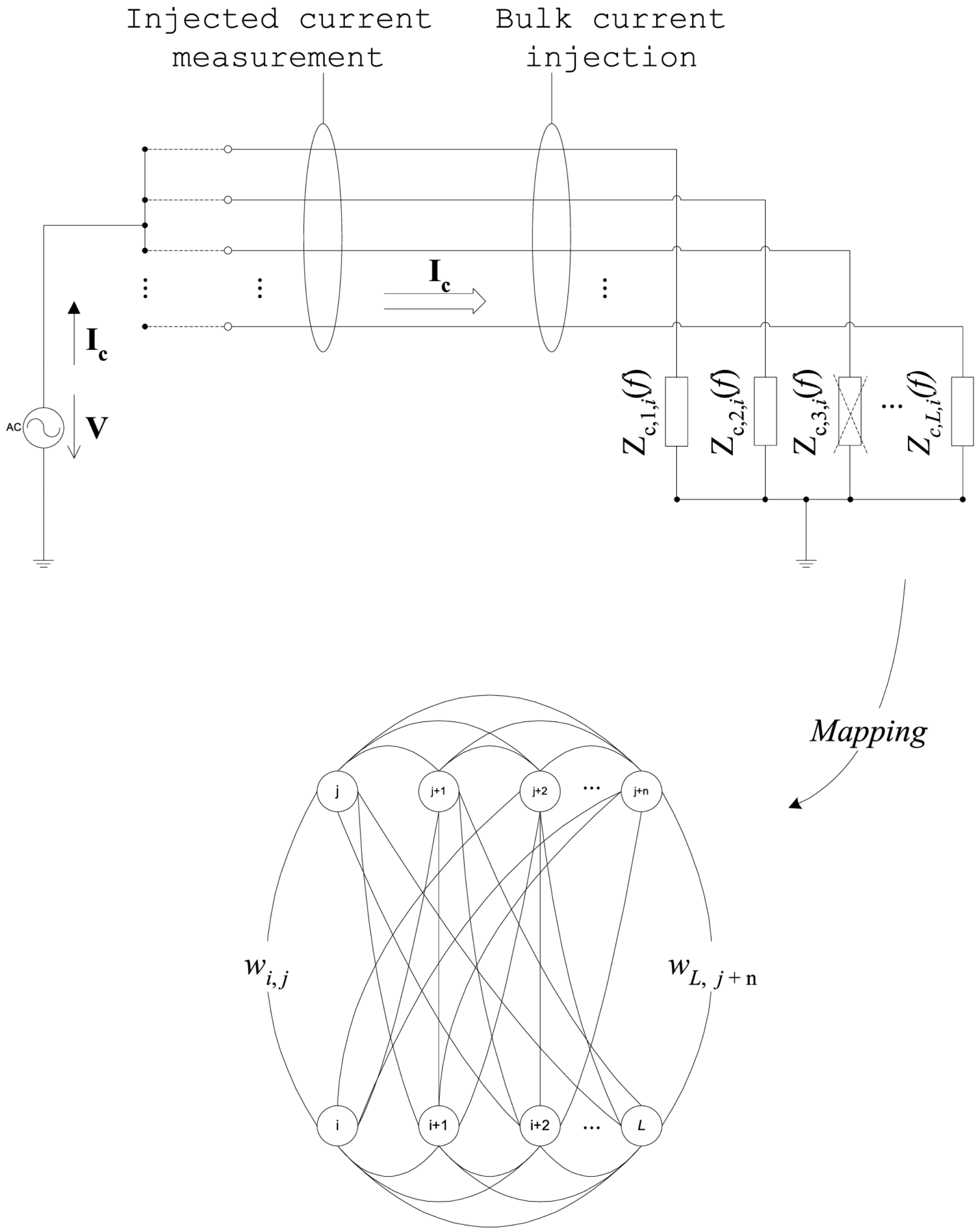

Figure 2.1 *Schematic description of the mapping problem*

An IN simplification task has to be encoded into the BM via the network weights $w_{i,j}$. In order to provide this mapping the system energy function of the BM has to be defined. The system energy $E(k)$ is a monotony increasing function of the BM neuron states k changing in time. From an analogy with thermodynamics, the energy function of the BM can be expressed as follows [22]

$$E(k) = \sum_{i=1}^{L} \sum_{j=1}^{L} w_{i,j}\ k(i)k(j), \tag{2.4}$$

where $w_{i,j}$ is the symmetric connection weight between the i–th and j–th neuron, $k(i)$ is the output of the i–th binary neuron, and L is the number of neurons. A network bias is represented by weight $w_{i,j}$ if $i = j$. The system energy $E(k)$ is in area of neural networks called *consensus function* and its usually denoted by $C(k)$.

Since the mapping problem is based on the equality $E = -O$,the cost function of the IN simplification problem has to be defined. In the following formulation we assume that particular access impedance is represented by two functions of frequency: the real $\Re\{Z_{i,j}(f_k)\}$ and the imaginary $\Im\{Z_{i,j}(f_k)\}$ parts. Than the equivalent impedance for each frequency sample $Z(f_k)$ can be simply calculated:

$$Z_{eq}(f_k) = \frac{1}{\frac{1}{Z_{1,j}(f_k)} + \frac{1}{Z_{2,j}(f_k)} + \ldots + \frac{1}{Z_{L,j}(f_k)}}. \tag{2.5}$$

Here k denotes the number of the impedance frequency sample. Notice that in (2.5), values of the access identifier j define the aircraft equipent configuration. In other words, the symbol j assigns an impedance characteristic to a particular IN element.

Since the objective is to reduce IN complexity some criteria evaluating the performance of a particular solution to the problem (particular state of the BM) have to be defined (see (2.3)):

- Since our first objective is to simplify impedance matrix complexity the first criterion is the number of elements r in the impedance network.

- The second objective is the IN model accuracy which is represented by the mean squared error (MSE) between reduced equivalent impedance $\tilde{Z}_{eq}$ and the original one Z_{eq} over all frequency samples f_k.

With respect to both of the criteria pointed above, one can express objective function O in such a way as to obtain all of the BM weights from the equality $E = -O$. This task is called the mapping problem.

To conclude this introduction to mapping strategy, the simplification task is a combinatorial optimization problem having the following characteristics:

- In general, all of the configurations of reduced IN are allowed. Thus, we are dealing with an unconstrained optimization problem.

- The criteria pointed above are obviously conflicting. The simplification problem is naturally multi-criterial.

- From the computational complexity point of view, it belongs to the class of **NP** problems because the running time of the Turing machine t_M is rather an exponential function of problem complexity L (number of elements in the IN): $t_M = 2^L$ (see (1.1)).

2.2 Mapping procedure

The mapping approach is based on the derivation of connection weights of the Boltzmann machine from the objective function defining a simplification problem. If proper mapping is provided the energy function of the Boltzmann machine is called *order-preserving* and satisfies the following condition:

$$\forall k, l \in \Omega : E(k) > E(l) \Rightarrow O(k) < O(l). \tag{2.6}$$

Here k and l denote two states of the Boltzmann machine belonging to the set of all possible BM (Boltzmann Machine) states Ω. Since, the state of the BM directly defines a configuration of reduced IN the objective function O can be evaluated directly as it is on the right side of expression 2.6.

In order to provide sufficient mapping, the objective function (see (2.3)) has to be more precisely defined. For that propose, let us define a binary vector $\mathbf{X} = \{x_1, x_2, \dots x_n\}^T$ identifying omitted elements. Simultaneously, let us consider that equivalent admittances Y_{eq} and $\tilde{Y}_{eq}$ are used instead of impedances to simplify (2.5) to sum the $Y_{i,j}(f_k)$ samples. Then, the objective function defined by equation 2.3 can be rewritten into the form suitable for our mapping strategy:

$$\begin{aligned} O(\mathbf{X}, \alpha) \quad &= \quad \underbrace{\sum_{k=1}^{K}\left(\sum_{i=1}^{L} x_i \Re\{Y_{i,k}\} - \sum_{i=1}^{L} \Re\{Y_{i,k}\}\right)^2}_{SSE \text{ over the real parts}} \\ &+ \quad \underbrace{\sum_{k=1}^{K}\left(\sum_{i=1}^{L} x_i \Im\{Y_{i,k}\} - \sum_{i=1}^{L} \Im\{Y_{i,k}\}\right)^2}_{SSE \text{ over the imaginary parts}} \\ &+ \quad \underbrace{\alpha \sum_{i=1}^{L} x_i}_{\text{degree of reduction}} \quad . \end{aligned} \tag{2.7}$$

Here, K is the number of frequency samples, symbol L denotes the number of IN elements and x_i states as the activator of the i–th IN element. The symbol j is an index pointing onto a particular load impedance from a bank of possible loads. It is obvious, if the vector $\mathbf{X}$ consists of ones, the SSE will be zero, while the last term in (2.7) (a reduction criterion) reaches its maximum value and visa-versa. Thus, the proposed equation (2.7) satisfies our requirements defined in (2.3) and it is simultaneously suitable for transformation in (2.4). Furthermore, due to the form of equation 2.7, we avoid complex numbers which cannot be dealt by the Boltzmann machine.

The following part of this chapter is focused on the exact derivation of the BM weights from the formulated objective function O. In order to describe our mapping strategy in a transparent way three pseudo-objective functions $\tilde{O}_1$-$\tilde{O}_3$ with increasing level of generality are defined:

- First, only one frequency sample of each IN element is assumed ($K = 1$) and only the imaginary part $\Im\{Y_i\}$ of each IN element is taken into account. Since the function $\tilde{O}_1$ involves no reduction criterion it doesn't represent a simplification task. However, it is a good starting point for the mapping procedure. Let us define the first pseudo objective function as follows:

$$\tilde{O}_1 = \left(\sum_{i=1}^{L} x_i \Im\{Y_i\} - \underbrace{\sum_{i=1}^{L} \Im\{Y_i\}}_{c} \right)^2 , \tag{2.8}$$

  where constant c denotes a negative value of the imaginary part of the original matrix equivalent admittance.

- In the second step, only the reduction criterion is added. Notice that, in spite of this simplification the pseudo-objective function $\tilde{O}_2$ could lead to simplified IN since the imaginary parts of Y_i can both vanish. Obviously, IN would be simplified according to the only one frequency sample observation.

$$\tilde{O}_2 = \left(\sum_{i=1}^{L} x_i \Im\{Y_i\} - \underbrace{\sum_{i=1}^{L} \Im\{Y_i\}}_{c} \right)^2 + \alpha \sum_{i=1}^{L} x_i \tag{2.9}$$

- The third pseudo-objective function $\tilde{O}_3$ extends function $\tilde{O}_2$ such that the K frequency samples are incorporated:

$$\tilde{O}_3 = \sum_{k=1}^{K} \left(\sum_{i=1}^{L} x_i \Im\{Y_{i,k}\} - \underbrace{\sum_{i=1}^{L} \Im\{Y_{i,k}\}}_{c_k} \right)^2 + \alpha \sum_{i=1}^{L} x_i \tag{2.10}$$

- It is obvious that the last extension, incorporating the real part $\Re\{Y_i\}$ of each IN element, leads to the original objective function O.

The artificial simplification of objective function O proposed above suggests that the mapping strategy will be described in four steps each one increasing in degree of generality. Before we start with mapping it is suitable to rewrite expression 2.4 into the following form:

$$E(\mathbf{X}) = \sum_{i=1}^{L} \sum_{j=1, j\neq i}^{L} w_{i,j}\ x_i x_j + \sum_{i=1}^{L} b_i x_i, \tag{2.11}$$

which holds the form of the energy function since the output of the i–th neuron equals the i–th entry of vector $\mathbf{X}$ ($k(i) = x_i$) and $x_i^2 = x_i$. A bias of the i–th neuron is denoted by symbol b_i instead of $w_{i,i}$. We don't introduce a weight vector in the brackets on the left side of (2.11) to emphasize that connection weights are fixed during the optimization process. Notice that, one can observe a similarity between the objective function in (2.7) and the recently rewritten energy function in (2.11), where the reduction term is represented as the sum of neuron states multiplied by their biases and the accuracy of reduced IN is denoted by sums of the mutual neuron states (pairs of IN elements).

Now, we have everything prepared to map the first pseudo-objective function:

$$\begin{aligned}\tilde{O}_1 &= \underbrace{x_1\left(\Im\{Y_1\}^2 + 2c\Im\{Y_1\}\right) + \ldots + x_L\left(\Im\{Y_L\}^2 + 2c\Im\{Y_L\}\right)}_{\sum_{i=1}^{L} x_i(\Im\{Y_i\}^2 + 2c\Im\{Y_i\})} \\ &+ \underbrace{\sum_{i=1, i\neq 1}^{L} x_1 x_i \Im\{Y_1\}\Im\{Y_i\} + \ldots + \sum_{i=1, i\neq L}^{L} x_L x_i \Im\{Y_L\}\Im\{Y_i\}}_{\sum_{i=1}^{L}\sum_{j=1,j\neq i}^{L} x_i x_j \Im\{Y_i\}\Im\{Y_j\}} + c^2 \\ &= \sum_{i=1}^{L} x_i\left(\Im\{Y_i\}^2 + 2c\Im\{Y_i\}\right) + \sum_{i=1}^{L}\sum_{j=1,j\neq i}^{L} x_i x_j \Im\{Y_i\}\Im\{Y_j\} + c^2.\end{aligned}$$

Now, the similarity between (2.11) and the derived pseudo-objective function is even more transparent. Condition (2.6) is satisfied because c^2 is a constant value.

Once we have mapped the fundamental objective, the reduction criterion can be involved:

$$\begin{aligned}\tilde{O}_2 &= \sum_{i=1}^{L} x_i\left(\Im\{Y_i\}^2 + 2c\Im\{Y_i\}\right) + \sum_{i=1}^{L}\sum_{j=1,j\neq i}^{L} x_i x_j \Im\{Y_i\}\Im\{Y_j\} + \alpha\sum_{i=1}^{L} x_i + c^2 \\ &= \sum_{i=1}^{L} x_i\left(\Im\{Y_i\}^2 + 2c\Im\{Y_i\} + \alpha\right) + \sum_{i=1}^{L}\sum_{j=1,j\neq i}^{L} x_i x_j \Im\{Y_i\}\Im\{Y_j\} + c^2.\end{aligned}$$

The reduction coefficient α occurs in the first term of $\tilde{O}_2$. This means, α influences only the neural network biases b_i, which is important to find and it will be discussed later.

As mentioned before, the third pseudo objective function $\tilde{O}_3$ incorporates K frequency samples. The following expression illustrates, how it could be added into the form of the BM energy function:

$$\begin{aligned}
\tilde{O}_3 &= \sum_{k=1}^{K} \tilde{O}_{1,k} + \alpha \sum_{i=1}^{L} x_i \\
&= \underbrace{\sum_{i=1}^{L} x_i \left(\Im\{Y_{i,1}\}^2 + 2c_1 \Im\{Y_{i,1}\}\right) + \ldots + \sum_{i=1}^{L} x_i \left(\Im\{Y_{i,K}\}^2 + 2c_K \Im\{Y_{i,K}\}\right)}_{\sum_{i=1}^{L} x_i \sum_{k}^{K} \left(\Im\{Y_{i,k}\}^2 + 2c_k \Im\{Y_{i,k}\}\right)} \\
&+ \underbrace{\sum_{i=1}^{L} \sum_{j=1, j\neq i}^{L} x_i x_j \Im\{Y_{i,1}\} \Im\{Y_{j,1}\} + \ldots + \sum_{i=1}^{L} \sum_{j=1, j\neq i}^{L} x_i x_j \Im\{Y_{i,K}\} \Im\{Y_{j,K}\}}_{\sum_{i=1}^{L} \sum_{j=1, j\neq i}^{L} x_i x_j \sum_{k=1}^{K} \Im\{Y_{i,k}\} \Im\{Y_{j,k}\}} \\
&+ \sum_{k=1}^{K} c_k^2 + \alpha \sum_{i=1}^{L} x_i,
\end{aligned}$$

this expression can be finally rewritten according to the particular terms in braces and incorporate the reduction criterion in function $\tilde{O}_2$:

$$\tilde{O}_3 = \sum_{i=1}^{L} x_i \left[\sum_{k}^{K} \left(\Im\{Y_{i,k}\}^2 + 2c_k \Im\{Y_{i,k}\}\right) + \alpha \right] + \sum_{i=1}^{L} \sum_{j=1, j\neq i}^{L} x_i x_j \sum_{k=1}^{K} \Im\{Y_{i,k}\} \Im\{Y_{j,k}\}. \tag{2.12}$$

Notice that, the sum of constants c_k was omitted since it has no effect on criterion (2.6) as mentioned above.

The last modification of the expressions above leads to the objective function (2.7). One can easily clarify that extending the function $\tilde{O}_3$ by the real parts of admittance leads to the following equation:

$$\begin{aligned}
O(\mathbf{X}, \alpha) &= \sum_{i=1}^{L} x_i \left\{ \sum_{k=1}^{K} \left[\Re\{Y_{i,k}\}^2 + \Im\{Y_{i,k}\}^2 + 2\left(\Re\{Y_{i,k}\} d_k + \Im\{Y_{i,k}\} c_k\right) \right] + \alpha \right\} \\
&+ \sum_{i=1}^{L} \sum_{j=1, j\neq i}^{L} x_i x_j \sum_{k=1}^{K} \left(\Im\{Y_{i,k}\} \Im\{Y_{j,k}\} + \Re\{Y_{i,k}\} \Re\{Y_{j,k}\} \right), \qquad (2.13)
\end{aligned}$$

where symbol d_k has a similar meaning as symbol c_k. The only difference is that c_k is the sum of the imaginary parts of the equivalent admittances and d_k is the sum of real ones.

Since all of the criteria from (2.7) are involved in (2.13), which holds the form of the energy function in (2.11), the mapping procedure is ended. According to the equality $O = -E$, the BM weights and biases can be simply expressed by comparing

(2.11) and (2.13):

$$w_{i,j} = -\sum_{k=1}^{K} \left(\Im\{Y_{i,k}\}\Im\{Y_{j,k}\} + \Re\{Y_{i,k}\}\Re\{Y_{j,k}\}\right) \tag{2.14}$$

$$b_i = -\sum_{k=1}^{K} \left[\Re\{Y_{i,k}\}^2 + \Im\{Y_{i,k}\}^2 + 2\left(\Re\{Y_{i,k}\}d_k + \Im\{Y_{i,k}\}c_k\right)\right] - 2 \cdot K \cdot \alpha \tag{2.15}$$

Equations 2.14 and 2.15 are the original results obtained by mapping of a simplification task onto the BM. Here, $2 \cdot K$ (the number of frequency samples) was introduced to hold a sufficient range for the coefficient α. The proposed coefficient is derived from the mean squared error over the frequency samples $MSE = (SSE_{\Re} + SSE_{\Im})/(2 \cdot K)$. It is obvious from equations (2.7) and (2.15) that, MSE is finally minimized instead of SSE, which was used for the simplicity of the mapping expressions. If the BM weights and biases are defined according to (2.14) and (2.15) the neural network will converge to the minimum of function O. The chapter 2.5 is dedicated to the study of BM convergence characteristics. An interpretation of the results and its further exploitation is under the scope of the following section.

2.3 Additional criteria

In the previous section, BM weights and biases were derived according to the formulation of our simplification problem in (2.3). Basically, the BM minimizes expected deviance between equivalent impedance of the original and reduced impedance networks. The expected error can be calculated over more frequency samples so that frequency dependent load impedance is incorporated. Thus, the resulting reduced impedance network exhibits similar behavior to the original network within a finite frequency interval.

Similarly as was done for frequency dependent impedance, we can incorporate more quantities e.g. temperature. The expected deviance has to be calculated over various reference temperatures and frequencies simultaneously. Then, the i-th particular load impedance from the IN is represented by a matrix of the complex admittance values $Y_{i,k,l}$. Each column of the admittance matrix acts as a frequency pattern while each row represents temperature dependence. Minimizing expected deviance over a matrix of equivalent impedance leads to a simplified IN whose behaviour can be guaranteed within a closed interval of temperature and frequency values. Objective function in (2.7) can be easily modified to incorporate additional dependence (see (2.16)). Here, M denotes the number of temperature samples and $Y_{i,k,l}$ is the i-th IN element admittance measured or simulated for the k-th value of frequency and the l-th value of temperature.

$$
\begin{aligned}
O(\mathbf{X},\alpha) &= \underbrace{\sum_{l=1}^{M}\sum_{k=1}^{K}\left(\sum_{i=1}^{L} x_i\Re\{Y_{i,k,l}\} - \sum_{i=1}^{L}\Re\{Y_{i,k,l}\}\right)^2}_{SSE \text{ over the real parts}} \\
&+ \underbrace{\sum_{l=1}^{M}\sum_{k=1}^{K}\left(\sum_{i=1}^{L} x_i\Im\{Y_{i,k,l}\} - \sum_{i=1}^{L}\Im\{Y_{i,k,l}\}\right)^2}_{SSE \text{ over the imaginary parts}} \qquad (2.16) \\
&+ \underbrace{\alpha\sum_{i=1}^{L} x_i}_{\text{degree of reduction}} \quad .
\end{aligned}
$$

Obviously, the modified objective function expressed above leads to a change in values of the weights and biases. The modification in weights and biases can be easily provided by adding one extra sum incorporating the new observable - temperature.

$$
w_{i,j} = -\sum_{l=1}^{M}\sum_{k=1}^{K}\left(\Im\{Y_{i,k,l}\}\Im\{Y_{j,k,l}\} + \Re\{Y_{i,k,l}\}\Re\{Y_{j,k,l}\}\right) \qquad (2.17)
$$

$$
b_i = -\sum_{l=1}^{M}\sum_{k=1}^{K}\left[\Re\{Y_{i,k,l}\}^2 + \Im\{Y_{i,k,l}\}^2 + 2\left(\Re\{Y_{i,k,l}\}d_k + \Im\{Y_{i,k,l}\}c_k\right)\right] - 2MK\cdot\alpha \qquad (2.18)
$$

The expression above illustrates two advantages of our approach. The first one lies in simplicity of incorporating an extra objective. Since the BM works with samples of load impedance, it can simplify an arbitrary IN, whose elements are characterized by measurement or numerical simulation.

The second advantage is the computational efficiency. One would expect increased computational complexity of a simplification algorithm if some new behaviour is encountered. It is the case of simulated annealing or genetic algorithm techniques, where it is required to evaluate the objective function in (2.16) in each iteration. The BM fixes its weights and biases within the annealing process. The extended objective function leads to a more complex expression for weight values which are calculated at the beginning. The remaining process of BM annealing is based on difference in energy (3.36) which is not effected by complexity of the objective function. A comparative study of the BM computational efficiency versus conventional optimization routines is given in chapter 4.6.

2.4 Intuitive aspect

Expressions 2.17 and 2.18 can be interpreted in an intuitive way motivating further investigation of the designed BM model. In order to give an intuitive picture about the obtained results, synaptic weights and biases are described in relation to the BM energy function.

Synaptic weight $w_{i,j}$ can be understood as the level of desirability that the i–th and the j–th neurons will be activated simultaneously. If the connection weight $w_{i,j} < 0$ the Boltzmann machine will tend not to activate both of the neurons simultaneously since this activation would lead to decreasing the energy function in (2.11). Those weights are called *inhibitory* weights or *inhibitory* connections. On the other hand, if $w_{i,j} > 0$ it is desirable to activate this connection because it will increase the energy function. In this case the weight is called *excitatory* weight.

Each particular neuron activation desirability is defined by neuron bias. If $b_i < 0$ the i–th neuron will tend to be switched off and visa-versa (see (2.11)). Let us further investigate equations 2.14 and 2.15 in connection with energy function in (2.11) and our objectives formulated in (2.3).

The neural network weight $w_{i,j}$ will be *excitatory* if imaginary parts of the admittance samples $\Im\{Y_{i,k}\}$ and $\Im\{Y_{j,k}\}$ have different signs and the real parts $\Re\{Y_{i,k}\}$ and $\Re\{Y_{j,k}\}$ are as small as possible over the K frequency samples (see equation 2.14). It is obvious that only admittances with different signs can cancel each other (see equation 2.5). Thus, the i–th and the j–th neurons are encouraged to be switched on by weight $w_{i,j}$ to maintain the SSE criterion. On the other hand, the i–th and the j–th neurons are forced to be switched off by the negative biases b_i and b_j.

The situation described above illustrates how the BM tackles the conflicting criteria. The most important finding here is, that one can increase the influence of reduction criterion simply by increasing coefficient α which leads to decreasing all biases (see equation 2.15). In other words, we can change the priorities between our two conflicting criteria simply by changing values of biases. This is promising result since it should be possible to force the BM to find a pareto-optimal set of solutions. In that case we would obtain a highly parallel multi-criteria combinatorial optimizer.

2.5 Conclusions on the mapping problem

A combinatorial approach to IN simplification was published by the author in [3]. Here, the mapping problem was solved via exact derivation of the weights and biases - parameters of an equivalent Boltzmann machine. As mentioned above, the found expressions maps the simplification problem onto the Boltzmann machine annealing. Naturally, the mapping procedure leads to a fast BM combinatorial solver since the BM energy function is computationally cheaper than the original objective function. Moreover, we showed that more criteria can be incorporated in the objective function with preserving the same computational complexity of annealing which is also superior to the conventional approaches.

From an intuitive point of view, the trade-off between two conflicting criteria (fitting and complexity) can be straightly driven via BM biases. This is an important result for further exploration of this approach in the field of multi-objective optimization.

3 Neural dynamics

3.1 Boltzmann machine characteristics

In the previous chapter, mapping the simplification task onto the BM was formulated. Once all of the weights of BM are defined its required to simulate annealing by BM in a proper way to obtain the highest system energy E. Thus, it is essential to investigate the Boltzmann machine dynamics.

A model of the stochastic neuron was defined in section 1.1.1 and the probability of the change of the binary neuron value was expressed in (1.2). Considering a sequential updating scheme, only one neuron i is updated according to equation (1.2) at each iteration step. It is required to calculate a system energy difference ΔE and set a system temperature T in order to obtain the desired probabilistic activation function $\mathbf{P}\{k(i) = 1|k(i) = 0\}$. A related energy difference can be calculated using expression 2.4 as follows

$$\Delta E_k(i) = E(k_i) - E(k), \tag{3.19}$$

where k_i denotes a state of the BM obtained from a configuration k by changing the state of the i–th neuron.

The system temperature T is defined by the cooling schedule. During the optimization process the parameter T is decreased which is analogue to the annealing process. It is required to let the BM reach the thermal equilibrium (maximum of energy function E) at each temperature T (each iteration of the annealing process). If the T is decreased slowly the BM mostly reaches its thermal equilibrium (TE) and the final BM state corresponds to the optimum solution to the problem. The BM asymptotically settles in the global optimum of the E once the temperature satisfies $T = 0$.

A cooling schedule is defined by: the starting temperature T_0, the lowest system temperature T_{min}, the cooling function, and the time step T_s. All of the parameters and the optimal cooling function have to be defined in order to guarantee BM convergence to the desired sub-optimal (close to optimal) solution.

Let us scope an asymptotic convergence of the Boltzmann machine. The convergence proof of the BM is similar to the convergence proof for the SA (Simulated Annealing) method. The main idea, how to formalize SA and BM, was invented by Aarts and Korst in [15]. Considering the stochastic nature of the BM, one can operate with the probability distribution of solutions k during the optimization process. The goal of the convergence proof is to express that only the optimal solutions exhibit non-zero probabilities at the end of the optimization process. Aarts and Korst proposed to use Markov chains for the formal expression of SA and BM.

3.2 Asymptotic Convergence

As mentioned in previous section, if the optimization problem is mapped properly onto the BM, then the energy function $E(k)$ equals the negative of the objective function $O(k)$. From the analogy of statistical mechanics, a stationary distribution $\boldsymbol{q}(T)$, also called Boltzmann distribution, serves that the probability of the k–th BM state can be expressed as follows [15]

$$q_k(T) = \frac{1}{N_0(T)} exp\left(\frac{-O(k)}{T}\right), \tag{3.20}$$

where

$$N_0(T) = \sum_{l \in \Omega} exp\left(\frac{-O(l)}{T}\right) \tag{3.21}$$

This the is so called *partition function.* The partition function is the enumeration of the Lagrange multiplier used in deriving the Boltzmann distribution in statistical mechanics. Here Ω denotes the configuration space of the BM (set of all possible BM states).

The convergence proof lies in two main points:

- It has to be derived that the Boltzmann machine equipped by stochastic neurons (see equation (1.2)) settles in a stationary distribution $\boldsymbol{q}(T)$ after an infinite number of performed iterations. The stationary distribution has to be reached in each value of control parameter T and it must be independent of initial state of the BM.
- The probabilities in (3.20) have to be zero for non-optimal solutions if $T \to 0$ and have to be uniformly distributed over a set of optimal solutions.

The first point is satisfied if a Markov chain represented by the BM (Boltzmann Machine) is *irreducible*, *aperiodic* and the BM stationary distribution satisfies the so called *detailed balance equation.*

Irreducibility

A Markov chain with transition matrix $\mathbf{P}$ is *irreducible*, if for each pair of configurations $k, l \in \Omega$ there is a non-zero probability of transition from k to l in a finite number of iterations [15]:

$$\forall k, l\; \exists n \geq 1 : (\mathbf{P}^n)_{k,l} > 0. \tag{3.22}$$

In the case of the Boltzmann machine, elements of transition matrix **P** can be expressed as follows

$$\mathbf{P}_{k,l}(T) = \begin{cases} G(i)A_k(i,T) & \text{if } l = k_i \\ 1 - \sum_{i=1}^{L} \mathbf{P}_{k,k_i}(T) & \text{if } l = k \\ 0 & \text{otherwise} \end{cases} . \tag{3.23}$$

Here $\mathbf{P}_{k,l}(T)$ denotes the probability that BM will change its configuration k to configuration l under given temperature T. The symbol $G(i)$ denotes the probability that the i–th neuron is proposed to be switched (uniform distribution is usually used), k_i is the state of BM obtained by switching the i–th neuron and $A_k(i,T)$ is an the acceptance probability where the proposed transition will be provided.

The acceptance probability is determined by the probabilistic activation function of the BM neuron model

$$A_k(i,T) = \frac{1}{1 + exp(-\frac{\Delta E_k(i)}{T})} \tag{3.24}$$

Equations (3.24) and (3.19) implies that $A_k(i,T) > 0$ if $T > 0$, which means that if the temperature is not zero, BM can change its state with the non-zero probability. This serves to prove *irreducibility* of BM:

$$\begin{aligned} \mathbf{P}^n_{k,l}(T) &= \sum_{r_1 \in \Omega_k} \sum_{r_2 \in \Omega_{r_1}} \sum_{r_3 \in \Omega_{r_2}} \cdots \sum_{r_{n-1} \in \Omega_{r_{n-2}}} \mathbf{P}_{k,r_1}\mathbf{P}_{r_1,r_2}\mathbf{P}_{r_2,r_3} \ldots \mathbf{P}_{r_{n-1},l} \\ &\geq G(i_1)A_k(i_1,T)G(i_2)A_{s_1}(i_2,T)G(i_3)A_{s_2}(i_3,T) \ldots G(i_n)A_{s_{n-1}}(i_n,T) \\ &> 0, \end{aligned} \tag{3.25}$$

where Ω_{r_j} is a set of BM configurations (neighbourhood) which can be obtained by switching one neuron of BM in the state r_j, s_i is the BM configuration which is obtained from state s by switching the i–th neuron. Generation probability $G(i)$ is the same for all neurons and it is grater than zero.

Aperiodicity

An irreducible Markov chain with transition matrix **P** is *aperiodic* if [15]

$$\exists k \in \Omega : \mathbf{P}_{k,k} > 0. \tag{3.26}$$

Regarding to transition probability (3.23) the condition 3.26 can be rewritten as follows

$$\begin{aligned}\mathbf{P}_{k,k}(T) &= 1 - \sum_{k_i \in \Omega_k} G(i) A_k(i,T) \\ &= 1 - G(j)A_k(j,T) - \sum_{k_i \in \Omega_k, k_i \neq k_j} G(i)A_k(i,T) \\ &> 1 - \sum_{k_i \in \Omega_k} G(i) = 0.\end{aligned} \tag{3.27}$$

We can write this, since there exists acceptance probability $A_k(j,T) < 1$ if Ω_k is not the set of optimal solutions (see equations (3.24) and (3.19)).

Balance equation

A Markov chain with transition matrix $\mathbf{P}$ has stationary distribution $\mathbf{q}$ if such a distribution satisfies the following equation

$$\forall k, l \in \Omega : q_k P_{k,l} = q_l P_{l,k}. \tag{3.28}$$

The *detailed balance equation* above can be expressed as in [15]

$$\begin{aligned}q_k(T)G(i)A_k(i,T) &= q_{k_i}(T)G(i)A_{k_i}(i,T) \\ q_k(T)A_k(i,T) &= q_{k_i}(T)A_{k_i}(i,T),\end{aligned} \tag{3.29}$$

then equation (3.29) can be verified using equations (3.19), 3.20 and 3.23 as follows

$$\begin{aligned}q_k(T)A_k(i,T) &= \frac{1}{N_0(T)} exp\left(\frac{E(k)}{T}\right) \frac{1}{1 + exp(-\frac{\Delta E_k(i)}{T})} \\ q_k(T)A_k(i,T) &= \frac{1}{N_0(T)} exp\left(\frac{E(k_i) - \Delta E_k(i)}{T}\right) \frac{1}{1 + exp(-\frac{\Delta E_k(i)}{T})} \\ q_k(T)A_k(i,T) &= \frac{1}{N_0(T)} exp\left(\frac{E(k_i)}{T}\right) \frac{1}{exp(-\frac{\Delta E_k(i)}{T}) + exp(\frac{\Delta E_k(i) - \Delta E_k(i)}{T})} \\ q_k(T)A_k(i,T) &= \frac{1}{N_0(T)} exp\left(\frac{E(k_i)}{T}\right) \frac{1}{1 + exp(-\frac{\Delta E_{k_i}(i)}{T})} = q_{k_i}(T)A_{k_i}(i,T).\end{aligned}$$

Optimal distribution

The final step which ensures an asymptotic convergence of the Boltzmann machine is to prove that $q_{nonopt}(T) = 0$ if $T \to 0$. As mentioned above, the probability of BM in state k after an infinite number of the iterations is independent from its initial state. It is to be proven that stationary probabilities in (3.20) are greater than zero only for set of optimal solutions

$$\lim_{T \to 0} q_k(T) = \frac{1}{|\Omega_{opt}|} \Gamma_{\Omega_{opt}}(k), \tag{3.30}$$

where $\boldsymbol{q}$ is the stationary distribution of the BM. Function $\Gamma_{\Omega_{opt}}(k)$ equals 1 if $k \in \Omega_{opt}$ and 0 otherwise. The symbol $|\Omega_{opt}|$ denotes a number of optimal states. It is possible to express the condition stated above in terms of BM stationary probabilities (3.20) and objective function O as follows

$$\lim_{T\to 0} q_k(T) = \lim_{T\to 0} \frac{exp\left(\frac{-O(k)}{T}\right)}{\sum_{l\in\Omega} exp\left(\frac{-O(l)}{T}\right)} = \lim_{T\to 0} \frac{\overbrace{exp\left(\frac{O_{opt}-O(k)}{T}\right)}^{\text{equals to 1 if } k\in\Omega_{opt}}}{\underbrace{\sum_{l\in\Omega} exp\left(\frac{O_{opt}-O(l)}{T}\right)}_{\text{equals to } n \text{ if } |\Omega_{opt}|=n}} = \frac{1}{|\Omega_{opt}|}\Gamma_{\Omega_{opt}}(k).$$

This holds because

$$\forall k \notin \Omega_{opt} : \lim_{T\to 0} = exp\left(\frac{O_{opt}-O(k)}{T}\right) = 0,$$

where O_{opt} is a value of the objective function in the optimal solution and $O(k)$ is a value of the objective function related to the k–th state of BM.

The asymptotic convergence proof stated above ensures that if the Boltzmann machine is properly simulated, it will converge to set of optimal solutions. In practical words, given proof of convergence, BM is capable to simplify an impedance network if a simplified network exists.

Obviously, a simulation of BM over infinity of the iterations is impractical. Our aim is to employ the BM to simplify given IN in the shortest possible time. This is the reason for exploiting suitable asymptotic cooling approximation previously called "a cooling schedule".

3.3 Finite-Time approximation

In the previous section, convergence of the optimization process has been proven for an infinite number of Boltzmann machine transitions (iterations). In this section an approximation of asymptotic behaviour is to be employed without losing versatility of the method. The approximation does not depend on a particular combinatorial problem but it is based on approximating the stationary distribution described in the previous section. The idea lies in substituting a *quasi equilibrium state* to the exact *equilibrium state* characterized by the stationary distribution $\boldsymbol{q}(T)$. Quasi equilibrium is defined by the following expression:

$$\|\boldsymbol{a}(N_l, T_l) - \boldsymbol{q}(T_l)\| < \varepsilon, \qquad (3.31)$$

where $a(N_l, T_l)$ is a quasi-stationary distribution obtained after N_l iterations (transitions) of BM under the temperature T_l in the l-th iteration. As mentioned above, a cooling schedule is defined by three parameters and another parameter arises with finite-time approximation. All of the cooling schedule parameters can be summarized in the following four points:

- An *initial value* of temperature T_0 has to be defined.
- A suitable *decrement function* for cooling has to be found.
- A finite number of iterations N_l leading to quasi-equilibrium state has to be expressed.
- A *final value* of temperature has to be chosen as a stop criterion

All of the points defining the finite-time cooling schedule are addressed in the following part. A conceptually simple cooling schedule proposed by Aarts and Van Laarhoven is investigated for our simplification algorithm. The reason for choosing a simpler scheme lies in the fact that it can be easily modified for parallel implementation of the BM. Our approach is to modify the schedule proposed in [15] for simulated annealing onto the form suitable for the Boltzmann machine. As it will be clear from the following paragraphs, the conceptually simple schedule is represented by an exponential temperature profile.

A set of simulation results is provided inside each paragraph denoted to a particular parameter . The Boltzmann machine was assumed to operate sequentially. In order to illustrate influences of all parameters, the simulations are executed in the same initial conditions (an IN to be reduced is always the same). More simplification problem instances can be found in chapter 4.6.

Initial temperature

The only criterion for the starting temperature T_0 is that 50 percent of proposed transitions from an initial state k_0 should be accepted by the Boltzmann machine. This probability corresponds to the BM neuron model 3.24. In this way, we can avoid stacking the BM in a local minimum of E. Initial acceptance probability (see a neuron acceptance probability in (3.24)) is closely connected with the maximum energy difference caused by possible transition. The worst case (the lowest transition probability) occurs if a transition from the highest level of energy function E_{max} to the lowest value E_{min} is proposed. Our approach is based on calculating the maximal negative difference $\Delta E_{min,max} = E_{min} - E_{max}$ and incorporating expression 3.24 to determine T_0 for $A_{k_0}(i, T_0) \approx 0.5$. If we chose $T_0 = -10 \cdot \Delta E_{min,max}$ the resulting acceptance probability is:

$$A_{k_0}(i, T_0) = \frac{1}{1 + exp(-\frac{\Delta E_{min,max}}{-10\Delta E_{min,max}})} = 0.475 \tag{3.32}$$

Since the transition leading to a maximal negative difference $\Delta E_{min,max}$ occurs with a very low probability, we consider that $A_{k_0}(i, T_0) = 0.475$ is appropriate for a practical application. Obviously, the BM reaches its maximal energy E_{max} if all of the excitatory connections (positive weights) are activated while the minimal energy E_{min} can be measured if all of the inhibitory connections are activated (negative weights). A connection is called activated if two connected neurons are "on". Thus, the maximal negative energy difference can be expressed as:

$$\Delta E_{min,max} = \sum_{i=1}^{L} \sum_{j=1,\{i,j\}\in N}^{L} w_{i,j} - \sum_{i=1}^{L} \sum_{j=1,\{i,j\}\in P}^{L} w_{i,j}, \tag{3.33}$$

where N and P are sets of the index pairs corresponding to negative and positive weights respectively. Notice that, the $\Delta E_{min,max}$ is overestimated by 3.33 since we deal with fully connected BM. This means that there are neurons which can not be activated by respecting both the negative and positive weights. This is obvious since our simplification task is formulated as a multi-criterial optimization.

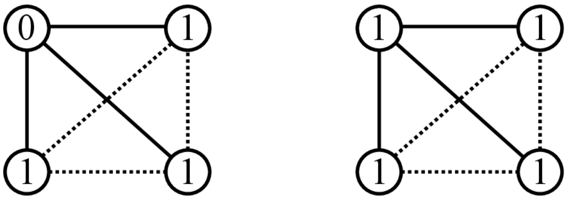

Figure 3.2 *A state of the BM corresponding to the first term in (3.33) (left); a state of the BM corresponding to the second term in 3.33 (right)*

In Figure 3.2, an exemplar BM is depicted. The dashed connections denote negative weights while the positive weights are represented by solid lines. The minimum energy state E_{min} can be reached in the case of our exemplar BM (see Figure 3.2 left). On the other hand, if we force the BM (Boltzmann Machine) to activate all of the positive weights, the maximal energy E_{max} does not appear since we also activate the inhibitory connections. This phenomenon implies that the maximal difference $\Delta E_{min,max}$ will be probably overestimated in the case of large BM. However, equation (3.33) is a good estimation to preserve a proper search of the entire solution space.

Figure 3.3 was obtained by a computer simulation of the BM simplifying an IN consisting of L = 100 elements. The reduction factor was set to $\alpha = 0.5$ and the IN was finally reduced by 39% . Please refer to chapter 4.6 dedicated to simulations for more detail. In figure 3.3, we demonstrate appropriate estimation of the initial value of parameter T_0. It can be clearly observed that the energy function defined in (2.4) (denoted by black thin line) has a large variance at high temperatures, thus BM can search the solution space properly. This corresponds to equation (3.20) for stationary distribution. As the BM is cooled down the variance approaches zero. This behaviour is illustrated in Figure 3.3 since actual values of the energy function approach their expected values (red thick line) at low temperatures. The expected values were estimated from sets of transitions proposed at each value of temperature T_l.

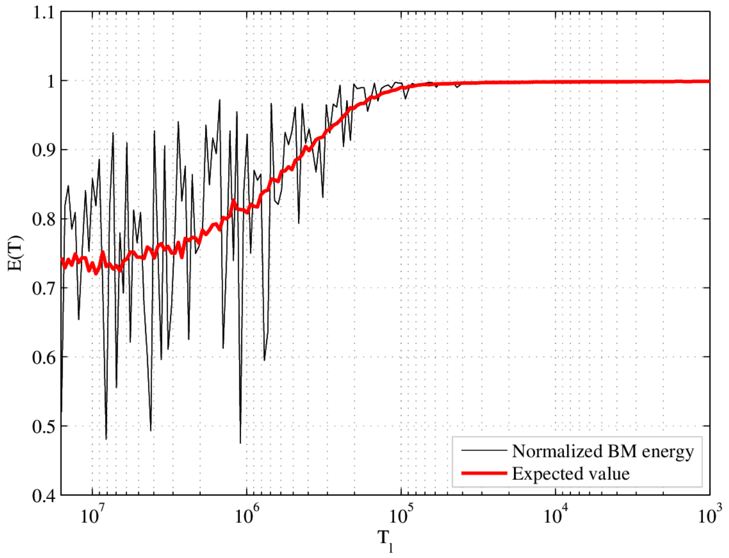

Figure 3.3 *Simulated BM convergence to the optimal distribution; an initial temperature T_0 was set according to 3.33*

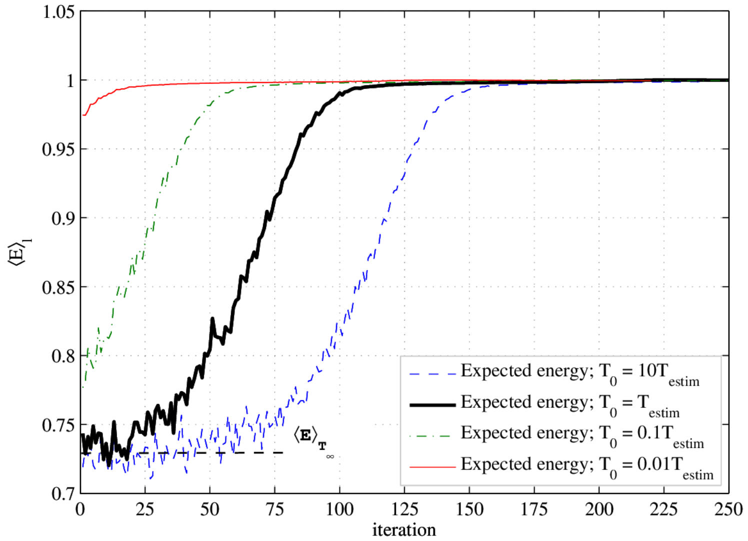

Figure 3.4 *A parametric study of initial temperature values; expected values of the energy function evolve in different ways; the BM run employing our estimation T_0 is depicted by black thick line*

Notice that, the energy function rather oscillates around its expected value than maximizes the consensus function over the iterations in Figure 3.3. It should be noted, that the BM doesn't search a solution space in the steepest descent manner. Convergence of the BM lies in the convergence of its stationary distribution as was mentioned above.

In Figure 3.4, four runs of the BM are depicted. Each simplification process was executed with respect to a different initial temperature: $T_0 = 10 \cdot T_{estim}$, $T_0 = T_{estim}$, $T_0 = 0.1 \cdot T_{estim}$ and $T_0 = 0.01 \cdot T_{estim}$. Here, T_{estim} is the value estimated in (3.33). In order to depict the parametric study in a transparent way, expected values of energy functions are depicted in Figure 3.4 rather than their actual values over the iterations. It is a sufficient approach providing that the energy state variances approach zero at low temperatures (see Figure 3.3). Since cooling was started in various melting temperatures $T_{0,i}$, the convergence curves are compared according to the number of iterations needed to reach the optimal distribution ($\langle E\rangle_{optim} = 1$).

Regarding the work of Aarts and Korst we obtained very similar curves in Figure 3.4 like they did for simulated annealing. The criterion for the initial value of T_0 states that all of the proposed transitions should be accepted with the same probability. This means, that the expected value of energy $\langle E\rangle_{T_0}$ should approximately equals its expected value for the infinitesimal temperature $\langle E\rangle_{T_\infty}$ (see dashed horizontal line in Figure 3.4). As can be observed in Figure 3.4, the expected value $\langle E\rangle_{10 \cdot T_{estim}}$ approximately equals $\langle E\rangle_{T_{estim}}$. Thus, we can consider that $\langle E\rangle_{T_{estim}} \approx \langle E\rangle_{T_\infty}$ which implies correctness of our approach.

One could argue that we can reach the same value of energy function (objective function) after lower number of iterations by employing a much lower initial temperature $T_0 = 0.01 \cdot T_{estim}$ (see Figure 3.4 red solid curve). We can say, that it is possible to use a lower initial temperature than $T_0 = T_{estim}$ but the result will be less reliable since the condition for proper statistical convergence is not satisfied.

Cooling profile

In the previous chapter, the asymptotic behavior of the BM was investigated and the temperature T of the BM was assumed to decrease smoothly. Moreover, it was considered that the stationary distribution of each Markov chain was achieved for each value of T. Since the aim is to approximate the asymptotic behaviour in finite time, decreasing the temperature and settling a particular Markov chain have to be discretized. The discretized process of annealing can be understood as a series of homogeneous Markov chains. If a slow cooling schedule is proposed (low difference between T_l and T_{l+1}) then stationary distributions of neighbouring Markov chains will be similar. Thus, the propagation of the stationary distribution (3.20) over the iterations doesn't need a lot of transitions (settling of a particular Markov chain) at each temperature T_l. If the BM is annealed faster (with a high difference between T_l and T_{l+1}) the situation is opposite. Notice that the faster the cooling the faster the optimization algorithm.

The ideas mentioned above concludes that the steeper the chosen cooling the lager the number of transitions required to obtain a quasi equilibrium state. On the other hand, a slow cooling schedule leads to a slow convergence of the stationary distribution (3.20) to the optimal distribution (3.30) (see Figure 3.5). Thus, we deal with two trade-off criteria: speed of cooling and speed of settling. Obviously,

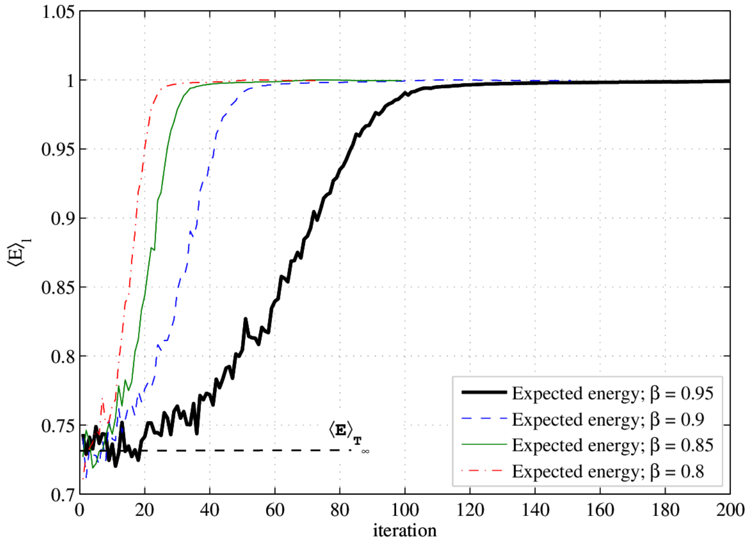

Figure 3.5 *A parametric study of speed of cooling; expected values of the energy function satisfy the initial criterion; the BM run related to the previous figure is depicted by black thick line*

both of the criteria influence the running time of the optimization process. For this reason both parameters β and N_l (see expressions (3.34) and (3.31)) are investigated in this paragraph. As mentioned above, the conceptually simple cooling schedule is appropriate for our purpose. In this case, a system is cooled down by an exponential profile, which can be described by the following recurrent equation:

$$T_{l+1} = \beta T_l, \tag{3.34}$$

where the coefficient β should vary between $\beta =< 0.8, 0.99 >$. The simulations provided in the previous paragraph were done with employing a slow cooling schedule $\beta = 0.95$ to ensure sufficiently accurate quasi equilibrium distributions over all iterations.

Four cooling profiles were considered to compose a parametric analysis depicted in Figure 3.5. Each curve was obtained by BM annealing which started with initial temperature $T_0 = T_{estim}$. If we compare Figures 3.4 and 3.5 we will obviously conclude that increasing the speed of cooling leads to faster convergence of the optimization process. However, in contrast with the first approach (lowering the initial temperature T_0), we obtain reliable results since the convergence criterion $E\rangle_{T_{estim}} \approx \langle E\rangle_{T_\infty}$ holds for each curve in Figure 3.5.

All of the simulations investigated above were provided with a sufficiently high number of transitions $N_l = 500 \cdot L$ to obtain an accurate estimation of quasi equilibrium distribution over all temperatures. Now it is needed to choose a lower N_l to increase the BM efficiency. Since the parameters β and N_l obviously influence each other, we first estimate an appropriate number of transitions N_l for each temperature T_l.

According to [15], the length of the l-th Markov chain can equal the number of BM neurons if the total number of neurons satisfies $L > 100$. This condition is satisfied since we are simplifying a very complex impedance network. Let us note, that the number of load impedances composing an IN equals the number of neurons L of the BM. Aarts found an approximation which returns the probability of selecting the i-th neuron for transition if N_l transitions are proposed:

$$P(i) = 1 - exp(-\frac{N_l}{L}). \tag{3.35}$$

If $N_l = L$ is chosen, then the probability of choosing the i-th neuron for transition equals to $P(i) \approx 2/3$. This means, that in 3 iterations of annealing process all of the neurons are asked for their transition, which is considered as sufficient in [15]. Note that, the distribution selecting neurons for transition remains uniform.

The Aarts estimation $N_l = L$ take into the account various problem complexities. In Figure 3.6 four simplification instances with increasing level of complexity are depicted: $L_1 = 100$, $L_2 = 200$, $L_3 = 500$ and $L_4 = 1000$. At first, each problem instance i was solved by BM with following the cooling schedule: $T_0 = T_{estim}$, $\beta = 0.95$ and $N_l = 500 \cdot L_i$. These solutions were taken as the reference. Then, much faster annealing was executed:$T_0 = T_{estim}$, $\beta = 0.8$ and $N_l = L_i$. Each i-th run of fast BM was related to the i-th reference solution. As we can see in Figure 3.6, our BMs converge properly to the reference solutions in all of the four cases. Let us note, that actual levels of energy are depicted in Figure 3.6 instead of their expected values.

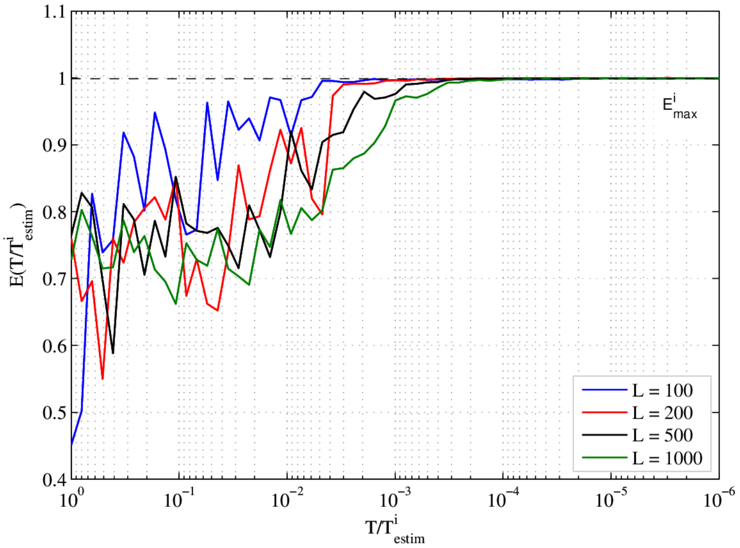

Figure 3.6 *Validation of cooling schedule parameters; $N_l = L$, $\beta = 0.8$; each curve is related to a particular run of the BM simplifying differently complex INs: L = 100, 200, 500 and 1000.*

Results depicted in Figure 3.6 show us that the estimation $N_l = L$ can be used for various problem complexities. Since all of the fast BMs converged to their reference solutions, we suggest the steepest cooling schedule $\beta = 0.8$ can be used for our simplifier.

Stopping criterion

In the previous paragraphs, we defined all of the three main parameters for a sufficient cooling schedule. Since the estimations of the cooling parameters taken into account characteristic of a particular simplification problem, we argue that they can be used for an arbitrary problem. The last parameter which acts as a stopping criterion is to be chosen experimentally. A proper quasi-optimal distribution can be recognized by one simple indicator. If the BM rejects all of the proposed transitions over m iterations, then its distribution is assumed to be close to the optimal one. As expressed above, the optimal distribution assigns zero probability to the states which are not optimal (see equation (3.30)). This means that the BM rejects all of the proposed transitions.

All of the experiments illustrated above employed the same stopping criterion $m = 5$. Thus, each run of the BM machine was stopped if all of the proposed transitions were not accepted in five consecutive temperature decrements. This condition was chosen according to Aarts and Korst's recommendation. Moreover, one can easily check the correctness of this approach by observing the saturated evolution of the energy function in Figures 3.4 - 3.6.

In each experiment (see Figures 3.4 - 3.6), we incorporated several trial impedance patterns representing serial and parallel resonate circuits and one higher order load circuit. These trial circuits were parametrized to compose a library of impedance patterns. Each IN used in a particular computer experiment above was composed by a random selection of patterns from the library of load impedances. Since the scope of this section was to analyse convergence of the BM in the case of finite-time approximation, a detailed description of simplified the INs was omitted. However, a more specific description of impedance patterns and comparison of BM with other optimization techniques will be given in chapter 4.6 which will be more practically oriented.

In this section, all of the parameters needed for sufficient BM execution were determined. The original contribution lies in formulating cooling schedule parameters proposed by Aarts and Korst in terms of the BM annealing. The simulations performed, showed good agreement with our expectations. The derived expressions can be applied in an arbitrary simplification case. Since BM annealing doesn't require computation of the objective function in each iteration, execution time of the BM simplifier is naturally smaller in comparison with the simulated annealing method even in the case of sequential BM implementation. This phenomenon will be addressed in the following section. The next section is focused on parallel execution of the BM. Probabilistic dynamics of parallel BM will be under the scope of the next section rather than analysing a particular parallel implementation.

3.4 Unlimited parallelism

Before we discuss the concept of parallel BM we should describe the simulation of sequential BM employed in the previous section more specifically. A transition process of sequential BM can be described in the following three points:

1. The i-th neuron is selected from a set of all neurons according to the generation probability $G(i)$ uniformly distributed over the set.

2. The calculation of the difference in energy caused by switching the i-th neuron if BM is in configuration k can be derived from (3.19):

   $$\Delta E_k(i) = (1 - 2k(i)) \left[\sum_{\{i,j\} \in Q_i}^{L-1} w_{i,j} k(j) + b_i \right], \tag{3.36}$$

   where $k(i)$ denotes the state of the i-th unit in configuration k and Q_i is the set of index pairs defining connections between the i-th neuron and its neighbouring units. This expression explains the efficiency of BM.

3. The i-th unit is switched with probability of transition given by the acceptance probability (activation function of stochastic neuron):

   $$A_k(i, T_l) = \frac{1}{1 + exp(-\frac{\Delta E_k(i)}{T_l})}. \tag{3.37}$$

   Here, we can clearly observe, how the temperature T_l influences the acceptance probability of each transition. High values of T_l can compensate a high negative difference in energy (3.36) which leads to the higher probability of the proposed transition. Low temperatures preserve transitions with positive difference in energy.

The routine described above is provided N_l times for each temperature T_l which is lowered according to the cooling schedule (see equation (3.34)). The efficiency of the BM lies in calculating the energy difference (3.36) which depends on the local feature of the BM configuration. For instance, the simulated annealing method or the genetic algorithm requires enumerating the entire objective function (2.7) to evaluate a solution which is more computationally demanding than equation (3.36).

Figure 3.7 depicts a fully connected BM trial with its connections. The connections of the i-th neuron with its neighbouring units used for the energy difference calculation in (3.36) are depicted by solid lines. The dashed connections don't influence the difference in energy. Obviously, this behaviour is essential for large BMs ($L > 100$).

A natural step from sequential BM towards its parallel execution is to sort BM units into independent sets without any direct connections between them. This

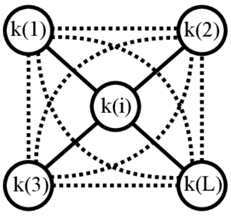

Figure 3.7 *An illustrative BM in configuration k, the i-th unit is to be switched*

approach allows us to provide transitions inside a particular group of units independently in parallel, since the differences in energy are independent of each other. Aarts and Korst called this approach limited parallelism. The advantage of this limited form of parallelism is that its asymptotic convergence can be proven [15]. On the other hand, in order to exploit this approach one has to find the minimum number of the independent sets. Unfortunately, this task is formulated as an **NP**-complete problem [15]. Moreover, our simplification task is represented by a full-connected BM and the limited form of parallelism can not be employed.

The following part of this section will be aimed at inspecting of the unlimited version of parallelism applied on our simplification problem. First, let us summarize the transition routine for unlimited parallelism as we did for sequential BM:

1. Random generation of a set of units which are to be proposed for transition simultaneously. In contrast with sequential BM, more than one neuron can update its state in one iteration. We will follow recommendations in [**?**] to fix the number of simultaneously selected units to $q = 2/3L$. This means that in each iteration q number of neurons is proposed for transition which is either accepted or not according to the following two criteria.

2. Calculation of the difference in energy is performed in the same way as was done for the sequential machine. The differences are independently calculated for all of the selected units and the actual BM configuration. Since two neighbouring units can be generated for transition, their energy differences could not be valid, due to equation (3.36) holding for one neuron transition in the set of neighbouring units. This is what Aarts called erroneously calculated differences and this is the reason, why rigorous proof of convergences doesn't exist for an unlimited version of parallelism. However, intuitive prove of asymptotic convergence was provided in [15] and supported by a large number of simulations.

3. Finally, the acceptance probability of the proposed transition is based on the same neuron model (3.37) as in the case of the sequential machine. The only difference here is that each i-th unit has its own parameter $T_{l,i}$. Basically, this is the reason why we have used the conceptually simple cooling schedule.

All of the three steps pointed above are iterated in the following way. Each unit has its own counter for the trials $N_{l,i}$, temperature $T_{l,i}$, and stopping criteria m_i.

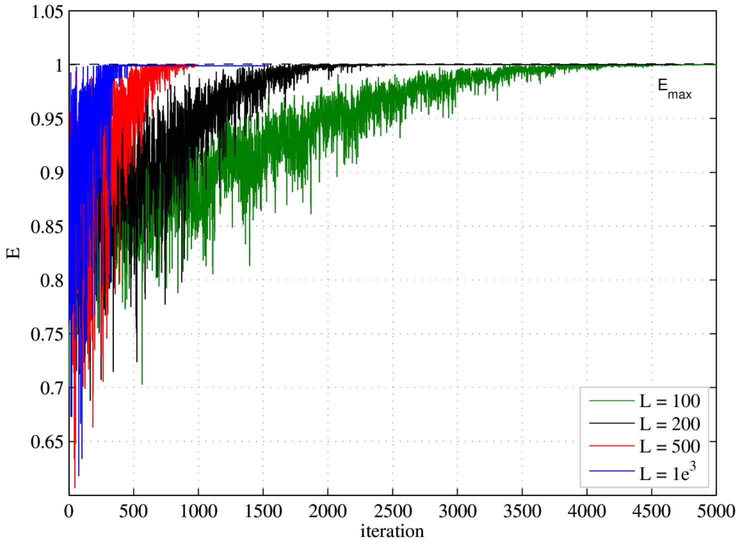

Figure 3.8 *Convergence of stationary distributions in parallel emulation of the BM related to the sequential simulations*

BM is annealed until all of the units are deactivated - all of the stopping counters m_i are zero.

Figure 3.8 depicts convergences of the parallel Boltzmann machines to their maximal energy states. The BMs were employed to solve four simplification instances which have been used in the previous tests (see Figure 3.7). One can clearly observe convergences of the stationary distributions in Figure 3.8: variance decreases to zero, and the expected value reaches the maximal energy. The results obtained using BM parallel emulation show us that we can exploit an unlimited version of BM parallelism even in the case of a fully-connected model which is the worst case for erroneously calculated differences. Thus, our application confirms Aarts and Korst experimental statements. Notice that rigorous convergence proof for parallel BM doesn't exist.

In the case of parallel BM, each unit has its own starting temperature, which is updated according to (3.34). The slope of the temperature decrements $\beta = 0.8$ was kept the same as for the sequential machine. It was the reason why the convergence curves were depicted over iterations in Figure 3.8 instead of temperature dependency in Figure 3.7. The i-th unit temperature is decreases if the unit was proposed for transition $N_l = L/4$ times. If the i-th unit is not updated for five consequent temperature decrements $m_i = 5$ (all of the transitions are rejected) the unit is blocked and its state remains fixed. Annealing is stopped if all of the units are blocked.

Obviously, parallel implementation can reduce a computation time significantly. On the other hand, settling time of the BM can increase due to erroneously calculated differences. This means that the reduction of execution time is not clearly proportional to the number of mutually activated units q. Both the sequential and parallel machines are compared from a time consumption point of view in the chapter 4.6 which dedicated to numerical simulations.

3.5 Conclusions on neural dynamics

Proof of BM asymptotic convergence based on the Aarts and Korst approach was derived in detail. A finite-time approximation of asymptotic behaviour was then examined along with estimating time schedule parameters. The initial temperature value strongly effects computational time of the annealing process and can prevent (or cause) settling the BM in a local minimum of the energy function. The equation estimating an initial temperature value was derived, the BM was simulated, and excellent agreement with Aarts proposal for simulated annealing was obtained. The remainder of the hyper-parameters were estimated according to recommendations in [15] and parameter analyses were performed to ensure the convergence of approximated behaviour. Finally, the BM was executed to solve our simplification problem in various problem instances and complexities. All simulations converged closely to the optimal solutions.

Furthermore, unlimited parallelism was investigated to explore the possibility of parallel simplification and speed up the combinatorial search. Parallel Boltzmann machines were emulated and various problem instances and complexities were solved. Convergence curves settled in the same solutions as we obtained using sequential BMs. On the other hand, the parallel execution is not clearly the faster approach to annealing since erroneously calculated energy differences slow down BM settling to its stationary distribution. A more detailed comparison of the sequential and parallel BMs is given in the last chapter of this work.

4 Probabilistic neural networks

4.1 PDF estimation

Dealing with a probabilistic approach to machine learning, PDF estimation is the key procedure which has to be provided in classification and approximation tasks. For instance, concerning a two class classification problem, PDFs are required for the Bayesian decision. Each point $\mathbf{X}_{bound}$ lying on the boundary between two classes A and B has to satisfy the following equation:

$$f_A(\mathbf{X}_{bound}) = \kappa f_B(\mathbf{X}_{bound}). \tag{4.38}$$

Here the PDF $f_A(\mathbf{X})$ gives the probability of occurrence of a vector $\mathbf{X} = \{x_1, x_2, \ldots, x_p\}$ in the case of vector belonging to class A while $f_B(\mathbf{X})$ denotes the PDF if vector $\mathbf{X}$ belongs to the class B. The symbol κ involves loss functions related to the decisions $\mathbf{X}_A$ and $\mathbf{X}_B$. This is the key concept of the PNN (Probabilistic Neural Network) proposed in [31].

Dealing with an approximation problem, paper [44] gives proof that a regression, also called a conditional mean value, is the best predictor for least-squares based learning. A neural network can be generally formulated as a regression estimator. Regression of y on $\mathbf{X}$ can be expressed as a conditional mean value using the Bayesian rule as follows

$$\begin{aligned} y(\mathbf{X}) &= E[y|\mathbf{X}] = \int y f(y|\mathbf{X}) \mathrm{d}y \\ &= \int y \frac{f(y,\mathbf{X})}{f(\mathbf{X})} \mathrm{d}y = \frac{\int y f(y,\mathbf{X}) \mathrm{d}y}{\int f(y,\mathbf{X}) \mathrm{d}y} \\ &= \frac{\int y f(y) f(\mathbf{X}) \mathrm{d}y}{\int f(y) f(\mathbf{X}) \mathrm{d}y} \end{aligned} \tag{4.39}$$

Here $f(\mathbf{X}, y)$ denotes joint PDF of the input vectors $\mathbf{X}$ and scalar variable measurements y. The $f(y)$ and $f(\mathbf{X})$ are marginal densities which are not generally known and have to be estimated from a training set. This is the key concept of the GRNN (General Regression Neural Network) invented in [43]. The kernel density estimation (KDE) is a non-parametric estimation method capable of modelling an arbitrary PDF based on a training set. Moreover, the Parzen window is consistent, which means that it returns a true PDF if a sufficiently high number of training samples is presented. Basically, the Parzen window is the subject of what we are focusing on, since it is the base of both PNN and GRNN.

It is obvious from equations (4.38) and (4.39) that PDF estimation plays an important role in classification and approximation problems. Moreover, a classification task can be expressed in terms of regression. Assuming the previously mentioned two class problem, we can define the value $y_A = 0$ as a class A identifier and $y_B = 1$ as a class B identifier. Hence, regression of y on $\mathbf{X}$ can be written as follows [44]

$$\begin{aligned} E[y|\mathbf{X}] &= \sum_i y_i P(y_i|\mathbf{X}) \\ &= y_A P(y_A|\mathbf{X}) + y_B P(y_B|\mathbf{X}) \qquad (4.40) \\ &= P(ClassB|\mathbf{X}) \end{aligned}$$

Thus, in the case of two a class problem, regression is a conditional probability of class B. The probability of class A can be easily obtained by $1 - P(ClassB|\mathbf{X})$. Then a decision boundary expressed in (4.38) can be found.

Coming back to neural networks, the PNN architecture implements the Parzen estimator for PDF extraction from the input data [31]:

$$f(\mathbf{X}) = \frac{1}{(2\pi)^{n/2}\sigma^n} \frac{1}{m} \sum_{i=1}^{m} \cdot \exp\left[-\frac{(\mathbf{X}-\mathbf{P}_i)^T(\mathbf{X}-\mathbf{P}_i)}{2\sigma^2} \right], \qquad (4.41)$$

where $\mathbf{X}$ is the n–dimensional input vector, $\mathbf{P_i}$ denotes the i–th input pattern, the kernel variance (smoothing parameter) is represented by σ^2, and m is the number of training patterns. The Parzen window belongs to the class of kernel density estimators and it is usually equipped with a multidimensional Gaussian kernel function. In this case, the PNN directly employs kernel density estimation (KDE) method.

In [43] the GRNN structure was derived by extending the KDE expression, defining PNN structure, by one dimension corresponding to the distribution of targets Y:

$$\begin{aligned} f(\mathbf{X}, Y) &= \frac{k}{m} \sum_{i=1}^{m} \overbrace{\cdot \exp\left[-\frac{(\mathbf{X}-\mathbf{P}_i)^T(\mathbf{X}-\mathbf{P}_i)}{2\sigma^2} \right]}^{\text{PNN term}} \qquad (4.42) \\ & \underbrace{\cdot \exp\left[-\frac{(Y-Y_i)^2}{2\sigma^2} \right]}_{\text{GRNN term}}, \end{aligned}$$

where $\mathbf{X} \in \mathbb{R}^n$ is the input vector, $Y \in \mathbb{R}$ denotes the output scalar value, $\{\mathbf{P}_i, Y_i\} \in \mathcal{D}$ is the i-th training pair from the set $\mathcal{D}$, m denotes the number of patterns, and

$k = (2\pi)^{-(n+1)/2}\sigma^{-(n+1)}$ normalizes the estimated $(n+1)$-dimensional distribution. Equation (4.42) implies that it is possible to derive a general criterion for PNN and GRNN if we are able to construct an objective function for assessing multivariate KDE.

The main disadvantage of the original PNN discussed in section 1.1.2 is clear from expression 4.41. The number of training patterns m equals the number of kernel functions required for PDF estimation. Notice that, due to this fact, the PNN and GRNN computational complexity can be high in cases of the multi-dimensional problems or dense training sets. It is essential to employ some kind of structural simplification (reduction of the number of kernels) to increase computational efficiency.

We argue that a clustering technique has to be connected with a sufficient kernel width estimator and vice-versa since both tasks deal with model complexity and contributes to the model bias and variance [44] in a similar way. For such a complex approach, we can see a lack of suitable criterion which can evaluate models having various numbers of neurons and different kernel widths.

Furthermore, it should be clear now that phenomena occurring in regression estimation problems are common for both approximation and classification problems. We will exploit this generalization in the next section where the "*bias* vs. *variance* dilemma" will be formulated. Kernel width (smoothing parameter) will be considered as regularization valve which sets the amount of the bias and variance in the model.

Finally, if a neural network estimates PDF well, a PNN classifier exhibiting minimal expected errors over a testing set or the best GRNN approximation will be obtained. We will show in the next section that adjusting kernel width (searching of optimal smoothing) σ is not a part of the training procedure but it is the model selection problem. The model selection problem is strongly tied with the over-fitting phenomenon.

4.2 Bias vs. Variance

Probabilistic neural networks can be formulated as non-parametric regression estimators. We will use this interpretation to express the over-fitting phenomenon in the sense of expected squared error. Then, the bias vs. variance dilemma will be demonstrated on the probabilistic neural networks PNN and GRNN which are under the scope of this chapter. In order to make all of our ideas transparent, terms form the area of machine learning and their counterparts from modern statistics are listed in the following table.

The bias vs. variance dilemma says that even a well trained neural network can exhibit very high error over the testing set. In the case of noisy data, such a network can snap on noise and it doesn't estimate the true regression function. Unfortunately, it is difficult to distinguish between contributions of a true original

Machine learning	**Modern statistics**
neural network	*non-parametric regression estimator*
synaptic weights	*free parameters*
training	*least-squares solution*
over-fitting	*bias vs. variance dilemma*
generalization	*approximation*
regularization	*restricting of hypothesis set*

Table 4.1 *Terms definition*

function and noise. This is the reason for the dilemma. The same phenomenon can be observed on estimators used in PNN and GRNN structures and this is the reason for investigating this question.

As will be shown, the dilemma is naturally involved in the mean-squared error (MSE) formulation of the neural network training criterion. The MSE expression can be split into two parts (bias and variance) each one contributing to the overall error.

As demonstrated in the previous section, the classification problem is a special case of regression. In order to express bias and variance terms from the MSE formulation, let us consider an approximation problem. Training data will be denoted by symbol $\mathcal{D}$, the feed forward neural network will be represented by function $g(\mathbf{X};\mathcal{D})$, and the true value of an original measurement will be denoted by y. Then, we can investigate the mean squared error $MSE = E[(y-g(\mathbf{X};\mathcal{D}))^2|\mathbf{X}]$ where the expected error is computed with respect to the conditional probability distribution $f(y|\mathbf{X})$ as follows [44]

$$\begin{aligned}
MSE &= E\Big[\Big(\big(y-E[y|\mathbf{X}]\big)+\big(E[y|\mathbf{X}]-g(\mathbf{X};\mathcal{D})\big)\Big)^2|\mathbf{X}\Big] \\
&= E\big[\big(y-E[y|\mathbf{X}]\big)^2|\mathbf{X}\big]+\big(E[y|\mathbf{X}]-g(\mathbf{X};\mathcal{D})\big)^2 \\
&\quad +2E\big[\big(y-E[y|\mathbf{X}]\big)|\mathbf{X}\big]\cdot\big(E[y|\mathbf{X}]-g(\mathbf{X};\mathcal{D})\big) \\
&= E\big[\big(y-E[y|\mathbf{X}]\big)^2|\mathbf{X}\big]+\big(E[y|\mathbf{X}]-g(\mathbf{X};\mathcal{D})\big)^2 \\
&\quad +\underbrace{2\big(E[y|\mathbf{X}]-E[y|\mathbf{X}]\big)\cdot\big(E[y|\mathbf{X}]-g(\mathbf{X};\mathcal{D})\big)}_{\text{equals to zero}} \\
&= E\big[\big(y-E[y|\mathbf{X}]\big)^2|\mathbf{X}\big]+\big(E[y|\mathbf{X}]-g(\mathbf{X};\mathcal{D})\big)^2,
\end{aligned} \tag{4.43}$$

where $E[y|\mathbf{X}]$ is the true regression function. Equation (4.43) splits the MSE criterion onto two parts (detailed derivation can be found in A.1). The first term the of decomposed expected error is data and model independent. Thus, the only term influencing the MSE which depends on the neural network is the second right part $(E[y|\mathbf{X}]-g(\mathbf{X};\mathcal{D}))^2$ of equation (4.43). This means that a measure of neural network performance can be formulated as a network deviation from the true regression. The following derivation (4.44), expresses an expected error $MSE_{\mathcal{D}}$ over an ensemble of the training sets $\mathcal{D}_i$ which is simply an average over all sets since the probability of each set $\mathcal{D}_i$ is the same.

$$\begin{aligned}
MSE_{\mathcal{D}} &= E_{\mathcal{D}}\Big[\Big(\big(g(\mathbf{X};\mathcal{D}) - E_{\mathcal{D}}[g(\mathbf{X};\mathcal{D})]\big) + \big(E_{\mathcal{D}}[g(\mathbf{X};\mathcal{D})] - E[y|\mathbf{X}]\big)\Big)^2\Big] \\
&= E_{\mathcal{D}}\big[\big(g(\mathbf{X};\mathcal{D}) - E_{\mathcal{D}}[g(\mathbf{X};\mathcal{D})]\big)^2\big] + \big(E_{\mathcal{D}}[g(\mathbf{X};\mathcal{D})] - E[y|\mathbf{X}]\big)^2 \\
&\quad +2E_{\mathcal{D}}\big[g(\mathbf{X};\mathcal{D}) - E_{\mathcal{D}}[g(\mathbf{X};\mathcal{D})]\big] \cdot \big(E_{\mathcal{D}}[g(\mathbf{X};\mathcal{D})] - E[y|\mathbf{X}]\big) \\
&= \underbrace{\big(E_{\mathcal{D}}[g(\mathbf{X};\mathcal{D})] - E[y|\mathbf{X}]\big)^2}_{\text{bias}} + \underbrace{E_{\mathcal{D}}\big[\big(g(\mathbf{X};\mathcal{D}) - E_{\mathcal{D}}[g(\mathbf{X};\mathcal{D})]\big)^2\big]}_{\text{variance}}
\end{aligned} \tag{4.44}$$

Considering that many various training sets $\mathcal{D}$ are presented to the network, the trained neural network should always result in a good estimation of regression. In other words, an optimal neural network should be independent of any change of training samples if the number of samples in each training set $\mathcal{D}$ is sufficiently high. Notice that, the optimal neural network would be noise independent in this case.

It is clear from (4.44) that the bias measures a deviation of averaged neural network output from the true regression, while the variance is a measure of neural network sensitivity on a change of training set or noise. These two terms are conflicting in the sense of expected error minimization.

Generally, the more free parameters a neural network has, the higher variance and lower bias is observed. This is due to high flexibility of the complex neural networks - it can learn a huge variety of behaviours. A high number of hypotheses is considered for training. Such a model will behave like a high order polynomial. On the other hand, if a neural network employs a lower number of free parameters the bias will increase and variance decrease. This is an analogy with a lower degree polynomial (e.g. linear function) which cannot model very complex (non-linear) behaviour.

In order to give a clear picture about bias and variance, two examples of neural estimations are depicted in Figures 4.9 and 4.10. Each particular example considers four training sets and two networks: a network with high variance and a biased network. The first experiment deals with the approximation task while the second one deals with PDF estimation required by the classification task.

The result of the approximation experiment can be observed in Figure 4.9. Four training sets were obtained by sampling a one-dimensional function $y = 20 - 10/[1 + (x)^2]$. Each training set was corrupted by Gaussian noise with standard deviation $\sigma_{noise} = 2$. A smoothing parameter (the width of the Parzen window) σ was varied to force the network to have a high bias (for $\sigma = 5$) or high variance (for $\sigma = 0.5$).

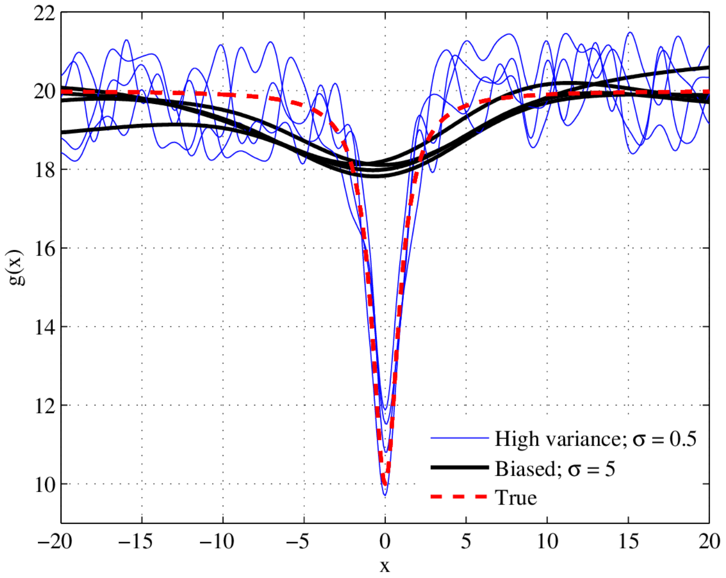

Figure 4.9 *Bias vs. variance demonstrated on mixed kernel density estimator used by GRNN; approximation task*

In the case of high bias (black thick curves), the neural network loses its flexibility. On the other hand the model is not very sensitive to noise. In the case of high variance model (blue thin curves), we can observe a high variance of estimations. The model better fits the sharp peak of the original function, but it is very sensitive to noise. The original function $y(x)$ is depicted by a red dashed line.

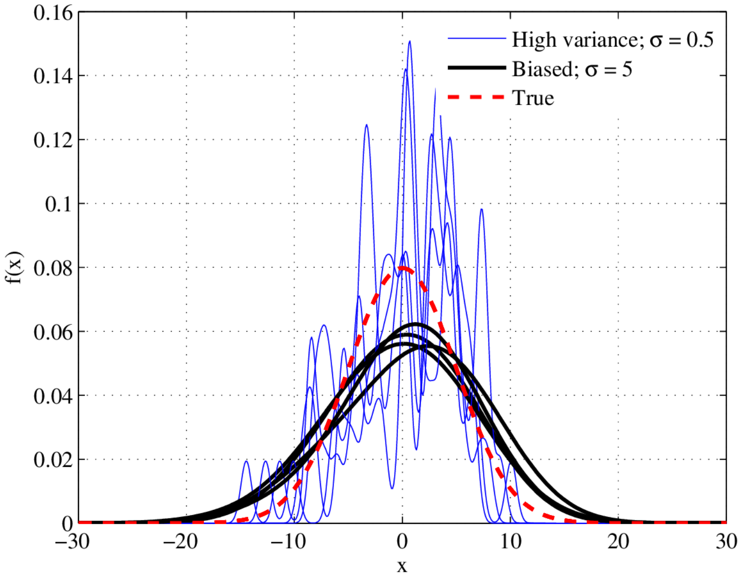

Figure 4.10 *Bias vs. variance demonstrated on the kernel density estimator employed by the PNN; classification task*

The second experiment dealing with the classification task was provided in a similar way as the first one. Four training sets were drawn from a Gaussian distribution with standard deviation $\sigma_{original} = 5$. This original distribution (depicted by a red dashed line in Figure 4.10) was estimated by KDE employed by PNN. The estimator was forced to have a high bias (for $\sigma = 5$) or high variance (for $\sigma = 0.5$).

In the case of biased estimator (black thick curves) the deviation from the original distribution is relatively high and independent of randomly drawn training sets. On the other hand, the high variance (blue thin curves) causes very unstable PDF estimation.

It should be clear from the two examples, that the bias vs. variance dilemma is common for both the approximation and classification tasks. Let us conclude this section with a general postulate. If we force a network to employ lower free parameters, we will systematically over-smooth data. Thus, the bias can be understood as a systematic error. On the other hand, if we let a model adjust a large number of free parameters, the network will fit noise in data and error will depend on this noise. Thus, the variance is viewed as a random error.

Selection of the optimal number of free parameters employed by a neural network can be understood as a *model selection problem*. The framework addressing these problems is called *regularization*. As we could observe, the PNN and GRNN model complexity is driven via the width of the Gaussian kernel σ. It generally holds that the higher σ the higher bias (lower variance) and vice-versa. The question is, how to find the optimal σ? This problem will be addressed in detail in the next sections.

4.3 Likelihood criterion

In the previous section, neural network performance was formulated in the sense of the MSE over various training sets $\mathcal{D}_i$. Let us inspect how the likelihood function corresponds to the bias vs. variance dilemma. Outputs of the Parzen window employed in PNN and GRNN have the form of probability densities. The likelihood $p(\mathcal{D}|\mathcal{E}, \sigma, \mathcal{H})$ is a joint probability of all data samples $\mathcal{D} = \{\mathbf{P}_i\}_{i=1}^{m}$ assigned by a model $\mathcal{H}$ (Parzen window). A set of free parameters $\mathcal{E} = \{\mathbf{Q}_j\}_{j=1}^{m}$ defines coordinates of Gaussian kernels (see (4.41)) and σ is a well known smoothing parameter:

$$p(\mathcal{D}|\mathcal{E}, \sigma, \mathcal{H}) = \frac{1}{(2\pi)^{(m+p)/2}\sigma^{(m+p)}m^m} \prod_{i=1}^{m} \sum_{j=1}^{m} exp\left[-\frac{(\mathbf{Q}_j - \mathbf{P}_i)^T(\mathbf{Q}_j - \mathbf{P}_i)}{2\sigma^2} \right]. \quad (4.45)$$

Notice that we let the window have various Gaussian kernel coordinates $\mathbf{Q}_j$, since it is considered to be a learning machine and the coordinates can be understood as another sample drawn from the original PDF. Obviously, a model $\mathcal{H}$ with maximal likelihood is not the most probable model for observed data $\mathcal{D}$. The model selection problem can be formulated as an estimation of optimal σ value which minimizes divergence between the true PDF and estimated one:

$$\sigma_{opt} = ArgMin \left\{ \int_{-\infty}^{\infty} PDF(x) ln \frac{PDF(x)}{PNN(x,\sigma)} \mathrm{d}x \right\}. \tag{4.46}$$

Here, the $PDF(x)$ is the original (unknown) PDF and $PNN(x, \sigma)$ is the density estimated by the neural network. Equation (4.46) is a commonly used criterion called Kullback-Leibler divergence or relative entropy. The neural network based on the Parzen window maximizes (4.45) if $\mathcal{D} = \mathcal{E}$ and $\sigma \to 0$ which is an unacceptable result as we will see in the next example. A model obtained by maximizing likelihood (a probability of the training data) has minimized bias but exhibits very high variance and criterion (4.46) is not minimized. This is the case depicted by a blue thin line in Figures 4.9 and 4.10.

Let us consider a three class problem to be solved by the PNN. Each of the classes A-C is represented by two samples: $\mathcal{D}_A = \{0, 1\}$, $\mathcal{D}_B = \{1.5, 3\}$ and $\mathcal{D}_C = \{4, 6\}$. Then, equation (4.45) can be decomposed into the following form

$$\begin{aligned} p(\mathcal{D}_2|\mathcal{E}, \sigma, \mathcal{H}) &= exp\left[-\frac{(P_1 - Q_1)^2 + (P_2 - Q_2)^2}{2\sigma^2}\right] + exp\left[-\frac{(P_1 - Q_2)^2 + (P_2 - Q_1)^2}{2\sigma^2}\right] \\ &+ exp\left[-\frac{(P_1 - Q_1)^2 + (P_2 - Q_1)^2}{2\sigma^2}\right] + exp\left[-\frac{(P_1 - Q_2)^2 + (P_2 - Q_2)^2}{2\sigma^2}\right] \end{aligned} \tag{4.47}$$

The third and fourth products in equation (4.47) approach zero if $\sigma \to 0$ since the arguments of the exponentials cannot be zeroed. These components act as Gaussian mixtures allowing the Parzen window to produce smooth densities and we will call them the inter-products. This is the reason why the only two Gaussians (the first two terms in (4.47)) can be observed in Figure (4.11) and PDF estimations in Figure 4.13 are strongly under-smoothed.

Figure 4.12 illustrates the evolution of likelihood function (4.45) for class C depending on the smoothing parameter σ_c, where the maximum value of likelihood is obtained when $\sigma \to 0$. In Figure 4.11, a smoothing parameter value is set close to zero $\sigma = 0.1$ in order to closely maximize (4.45) and for practical reasons to visualize non-zero components. Obviously, if we set $\sigma = 0$ only Dirac pulses will be observed at $\mathcal{D}_C = \mathcal{E} = \{4, 6\}$. Finally, Figure 4.13 shows the result of density estimation and pattern classification. It is clear, that densities estimated by the PNN don't represent the classes very well even if the classification result seems to be right. One would expect rather smooth density over the entire region of each class.

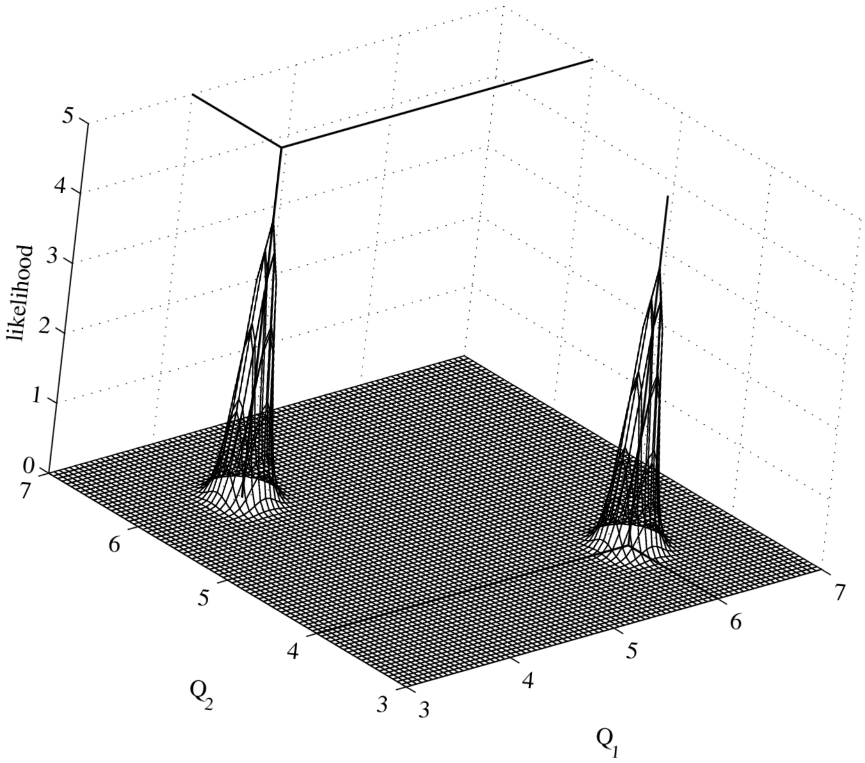

Figure 4.11 *Likelihood function for $\mathcal{D}_C$ set depending on PNN free parameters Q; $\sigma_c = 0.1$*

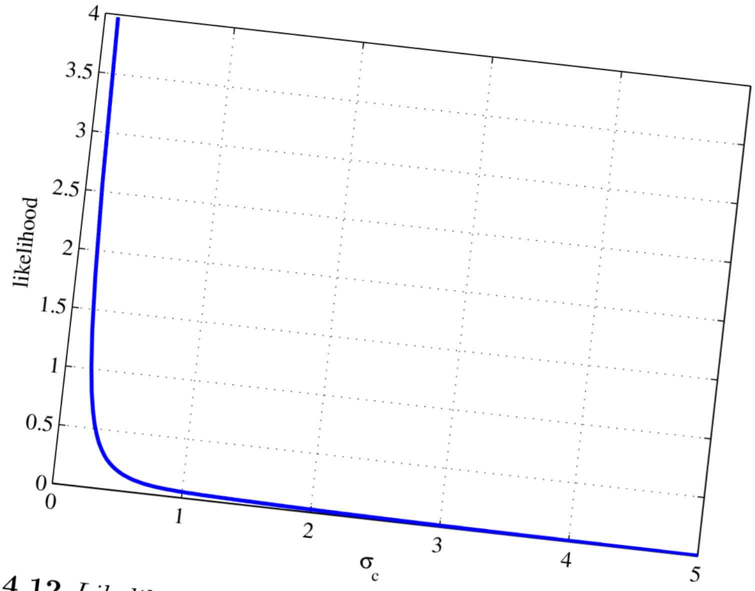

Figure 4.12 *Likelihood function for $\mathcal{D}_C$ set depending on smoothing parameter σ_c*

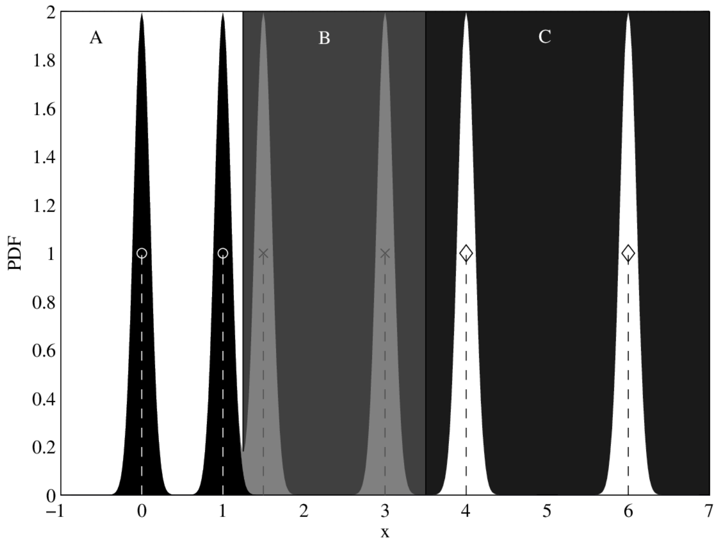

Figure 4.13 *Three class problem solved by PNN network (classes are depicted by filled boxes) maximizing likelihood function; PDF functions estimated by PNN are illustrated as sharp peaks inside the class regions A-C; training data are represented by markers*

Due to the previous arguments the ordinary likelihood function is not suitable for the model selection procedure and an alternative likelihood cross-validation (CV) function used to be usually employed [46]. In the next section we will exploit Bayesian statistic to derive a more accurate, comprehensive and intuitive model selection criterion for searching the optimal multivariate KDE kernel width, setting PNN and GRNN bias/variance equilibrium, and determining a sufficient number of radial neurons.

4.4 Bayesian strategy

The Bayesian approach to machine learning was comprehensively studied by D. McKay in his doctoral thesis [47]. The main idea lies in the suggestion that all of the quantities connected with a learning machine (eq. free parameters $\mathcal{E}$, smoothing parameter σ, or machine structure $\mathcal{H}$) are understood as random quantities. In this way, all of the characteristic properties of a particular neural model (eg. fitting, complexity, sensitivity) can be mapped onto the characteristic probability distribution - the posterior ($p(\mathcal{E}|\mathcal{D},\sigma)$ in our case). Based on the posterior distribution, a particular neural model can be selected -an equilibrium between bias and variance can be found.

Basic concept

The posterior distribution for our model $\mathcal{H}$ can be written in the following way:

$$\begin{aligned} p(\mathcal{E}|\mathcal{D},\sigma,\mathcal{H}) &= \frac{p(\mathcal{D},\mathcal{E},\sigma,\mathcal{H})}{p(\mathcal{D},\sigma,\mathcal{H})} = \frac{p(\mathcal{D}|\mathcal{E},\sigma,\mathcal{H})\cdot p(\mathcal{E}|\sigma,\mathcal{H})\cdot p(\sigma,\mathcal{H})}{p(\mathcal{D},\sigma,\mathcal{H})} \\ &= \frac{p(\mathcal{D}|\mathcal{E},\sigma,\mathcal{H})\cdot p(\mathcal{E})}{p(\mathcal{D}|\sigma,\mathcal{H})} = \frac{\text{likelihood x prior}}{\text{evidence}}. \end{aligned} \tag{4.48}$$

Here, the likelihood function has the form of equation (4.45) and the prior distribution of network free parameters $p(\mathcal{E}) = p(\mathcal{E}|\sigma,\mathcal{H})$ is independent of the chosen model and smoothing parameter σ. The prior distribution $p(\mathcal{E})$ is our prior knowledge or assumption about the original PDF since free parameters of the Parzen window $\mathcal{E} = \{\mathbf{Q}_j\}_{j=1}^{m}$ can be understood as samples drawn from the unknown density. Finally, the evidence (EV) $p(\mathcal{D}|\sigma,\mathcal{H})$ is the degree of belief that our model $\mathcal{H}$ with defined smoothing σ generates training data $\mathcal{D}$. Notice that the EV doesn't depend on a particular configuration of the free parameters $\mathcal{E}$. This means that the EV incorporates all of the possible configurations and can be understood as a measure of general model performance for a given value of σ. As we will see, the EV can act as a criterion for the model selection task. Firstly, let us explain the connection between the EV and bias vs. variance dilemma.

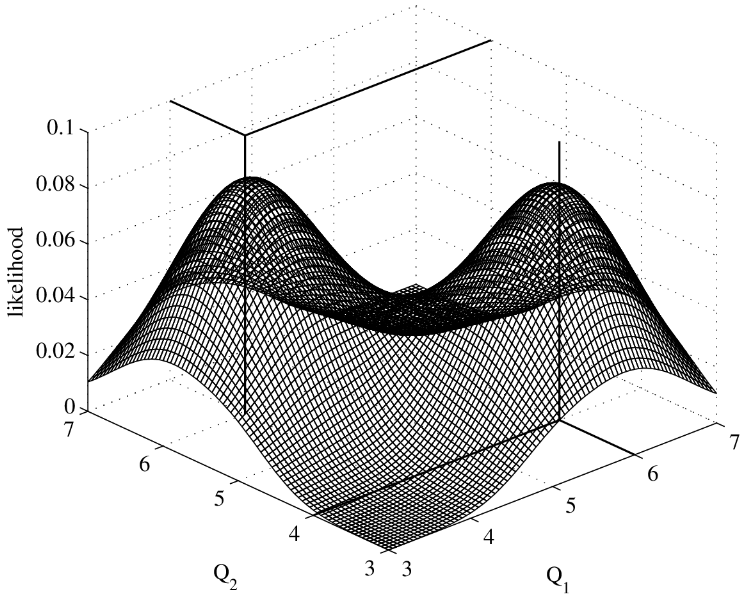

Figure 4.14 *Likelihood function for $\mathcal{D}_C$ set depending on PNN free parameters Q; $\sigma_c = 0.7$*

As was shown, likelihood has its maximum at $\sigma = 0$ (see Figure 4.12). The second effect of minimizing σ is that posterior density $p(\mathcal{E}|\mathcal{D},\sigma,\mathcal{H})$ becomes higher and narrower which can be observed in Figure 4.11. Notice that the likelihood functions depicted in Figures 4.11 and 4.14 exhibits the same shape as the posterior distributions if uniform prior distribution $p(\mathcal{E})$ is assumed. The narrower posterior density is observed, the more complex and sensitive model we have. This means that if we slightly change the configuration of free parameters $\mathcal{E}$, the model will produce very different estimations. Since the Parzen window free parameters can be understood as samples of the original PDF, the model with narrow posterior distribution will exhibit high MSE variance.

On the other hand, if the parameter σ takes higher values, the likelihood decreases and posterior distribution becomes to be flat (see Figure 4.14) which indicates a less complex model. In this case, a neural network is less sensitive on its parameters so variance decreases. Unfortunately, as the posterior further expands, the model becomes over-smoothed and the bias increases significantly which is indicated by the likelihood function decreasing.

In other words, the EV can increase even if the likelihood decreases (fitting gets worst) since the posterior distribution simultaneously expands (model gets simpler). This is due to the following formula rewritten from equation (4.48)

$$p(\mathcal{D}|\sigma,\mathcal{H}) = \frac{p(\mathcal{D}|\mathcal{E},\sigma,\mathcal{H}) \cdot p(\mathcal{E})}{p(\mathcal{E}|\mathcal{D},\sigma,\mathcal{H})}, \tag{4.49}$$

which implies that the EV is a quantity incorporating Occam's razor stating that less sensitive models with a lower number of the degrees of freedom (DOF) are more probable than very complex ones having many of the DOF.

In order to summarize just the obtained results, we can say the following. Smoothing parameter σ drives the expansion of posterior density which simultaneously results in decreasing likelihood. The EV is the ratio of likelihood to posterior densities. So, searching the maximum of the EV function equals searching the equilibrium between model fitting (likelihood) and Occam's razor (posterior accessible volume). This is a naturally formulated equilibrium between bias and variance.

Maximizing the EV, the posterior distribution of smoothing parameter $p(\sigma|\mathcal{D},\mathcal{H})$ is also maximized which proves the correctness of the idea:

$$p(\sigma|\mathcal{D},\mathcal{H}) = \frac{\overbrace{p(\mathcal{D}|\sigma,\mathcal{H})}^{\text{model evidence}} \cdot \overbrace{p(\sigma|\mathcal{H}))}^{\text{prior}}}{\underbrace{p(\mathcal{D}|\mathcal{H})}_{\text{evidence of Parzen window}}}. \tag{4.50}$$

A fundamental example

The developed neural model estimating probabilities in Figure 4.16 was obtained by maximizing the EV for each class A–C independently: $\sigma_A = 0.5$, $\sigma_B = 0.75$, $\sigma_C = 1$ (see Figure 4.15). The first difference from the PNN maximizing likelihood is that we obtained smooth class densities. In Figure 4.16, classification areas depicted by filled boxes are kept the same as in Figure 4.13. A second difference can be observed - decision boundaries between classes (see equation (4.38)) are shifted. This is caused by different widths and heights of particular densities A-C. This is an intuitive result since one would expect narrower distribution for data $\mathcal{D}_A$ than for the set $\mathcal{D}_C$.

Our exemplar PNN (Probabilistic Neural Network) consists of three independent Parzen windows. This is the difference from the original PNN structure, which operates with one global smoothing parameter. We simply exploit the independence of Parzen windows in the PNN and optimized each σ_i separately. This idea naturally increases flexibility of the original network on the one hand and simplifies model selection problem on the other hand. The model selection strategy lies in the efficient calculation of the EV, which is to be maximized for each Parzen window from a neural network. This issue will be addressed in the next section.

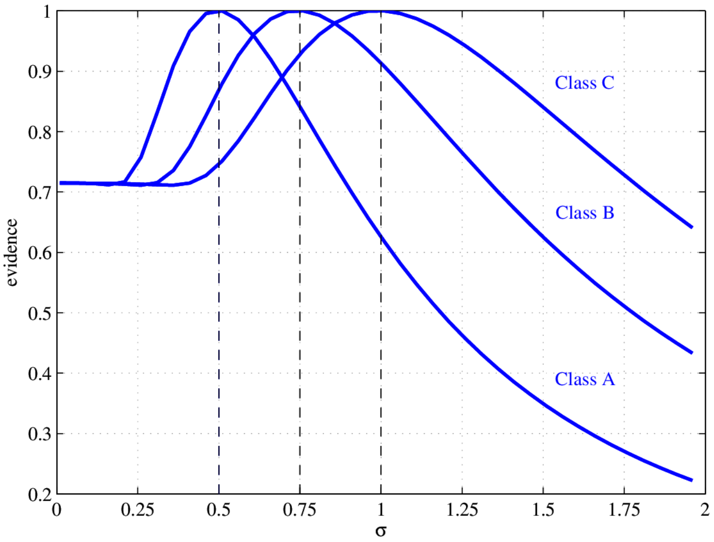

Figure 4.15 *Evolution of the evidence criterion for each training set $\mathcal{D}_A$–$\mathcal{D}_C$ depending on the PNN smoothing parameter*

4.5 Bayesian procedure

Obviously, the posterior distribution $p(\mathcal{E}|\mathcal{D}, \sigma, \mathcal{H})$ isn't known and the EV has to be estimated by marginalizing the likelihood distribution:

$$p(\mathcal{D}|\sigma, \mathcal{H}) = \int_{\mathcal{E}} p(\mathcal{D}, \mathcal{E}|\sigma, \mathcal{H})\mathrm{d}\mathcal{E} = \int_{\mathcal{E}} p(\mathcal{D}|\mathcal{E}, \sigma, \mathcal{H}) \cdot p(\mathcal{E})\mathrm{d}\mathcal{E}, \qquad (4.51)$$

where the prior distribution $p(\mathcal{E})$ has to be chosen and we have to integrate over the entire space of free parameters. Exemplary results depicted in Figure 4.15 illustrating evolution of the EV for our classification were obtained by a numerical quadrature integration technique. As we have already mentioned, once the integral

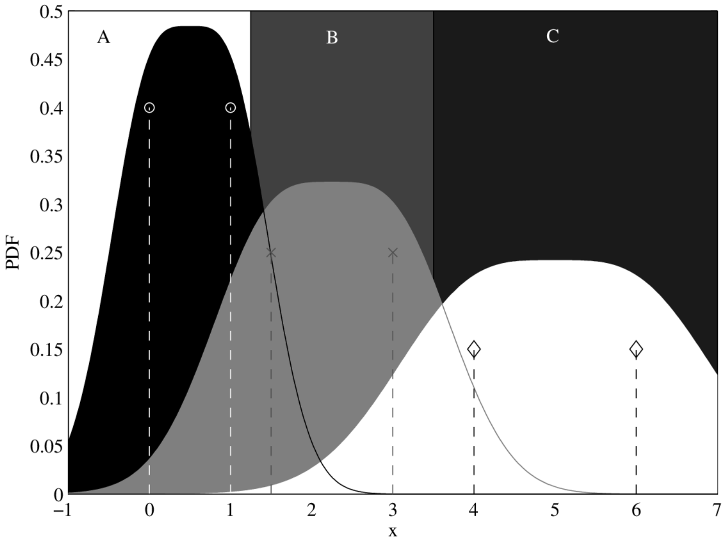

Figure 4.16 *A three class problem solved using the PNN network maximizing the EV; training data are represented by markers*

(4.51) is solved we can find optimal smoothing σ (bias and variance equilibrium) according to the highest EV of the model. The question is, how to calculate the marginalization integral more efficiently.

Generally, the marginalizing procedure is a crucial step in most of the Bayesian approaches. Firstly, equation (4.51) cannot be performed analytically due to the form of likelihood (see (4.45)). In open literature, there are several commonly used approximative integration routines, a brief overview can be found in [48]. Unfortunately, none of those approaches is suitable for our purpose due to high dimensionality of the marginalization integral which destroys the applicability of both the Monte Carlo or quadrature approaches. Finally, the posterior distribution exhibits multi-modal character (for small σ) so that the approach from [47] cannot be used. However, a unique efficient integration procedure can be obtained if all of the aspects are taken into the account.

The following part of this section is dedicated to the design of an analytical approximative integration method specialized to our marginalization problem. A uniform prior distribution $p(\mathcal{E})$ is chosen since we have no prior knowledge of shape of the original PDF. This is, as we will see, a mathematically convenient option. Then, the likelihood function is decomposed as the sum of the products and each product is locally approximated by a polynomial. In this way we obtain an analytically integrable approximative form of the likelihood. A combinatorial simplification technique is designed to decrease the computational complexity of the problem. First, a one dimensional PDF estimator is considered to illustrate the ideas in a transparent way. Then, the approach is generalized for multivariate cases.

Prior distribution

Prior distribution $p(\mathcal{E})$ occurring in (4.51) can be considered as a uniform distribution of the samples $\mathcal{E}$. Since we have no prior knowledge about the original distribution, the choice of uniform distribution is an intuitive option which is, furthermore, mathematically convenient. If a uniform $p(\mathcal{E})$ is employed, it acts as a window defining limits of the integral in equation (4.51) and we can write:

$$p_{(unif)}(\mathcal{D}|\sigma, \mathcal{H}) = \alpha \cdot \int_{\hat{\mathcal{E}}} p(\mathcal{D}|\mathcal{E}, \sigma, \mathcal{H}) \cdot \mathrm{d}\mathcal{E} \tag{4.52}$$

where $\hat{\mathcal{E}}$ is a subspace of the free parameters whose volume is defined by the uniform distribution. The constant α maintains unity volume of the prior distribution. Notice that the α can be omitted if the dataset is scaled so that it lies within the interval $[0, 1]$.

This approach simplifies the EV estimation since the prior distribution is naturally incorporated within the integral limits and the problem is simplified to marginalizing of the likelihood function over a finite subspace of the free parameters. The following three paragraphs will discuss the effect of the prior assumption.

Firstly, in Figure 4.17, normalized EV estimations considering Gaussian (blue curve) and uniform (red curve) prior distributions are compared in the example given in the previous section. As can be observe, both approaches lead to the same EV extrema-the same neural models are selected. Moreover, it generally holds that the more training samples, the smaller effect of the prior assumption.

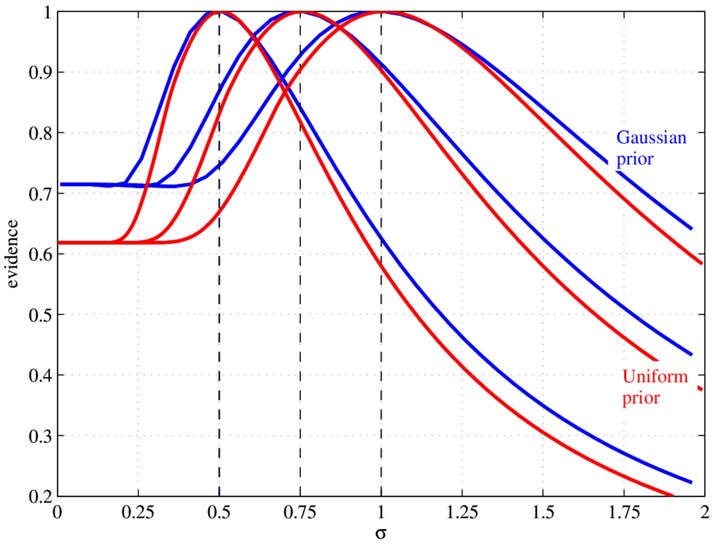

Figure 4.17 *Comparison of the two types of prior distributions; uniform distribution exhibit the same coordinates of the maximal EVs as it was in the case of Gaussian prior distribution*

The evidence is not very sensitive to choice of the prior distribution width since most of the likelihood components usually lie far enough from the limits of integration which is demonstrated in Figure 4.18, where the Gaussian PDF was to be estimated by the Parzen window based on twenty kernels. As will be seen later, the inter-products (see (4.43)) are more likely placed in between the integral limits and simultaneously exhibit narrower bandwidth σ_{inter}. Due to these reasons, we consider the simplest prior distribution definition: limits are given by the boundary samples as in Figure 4.18).

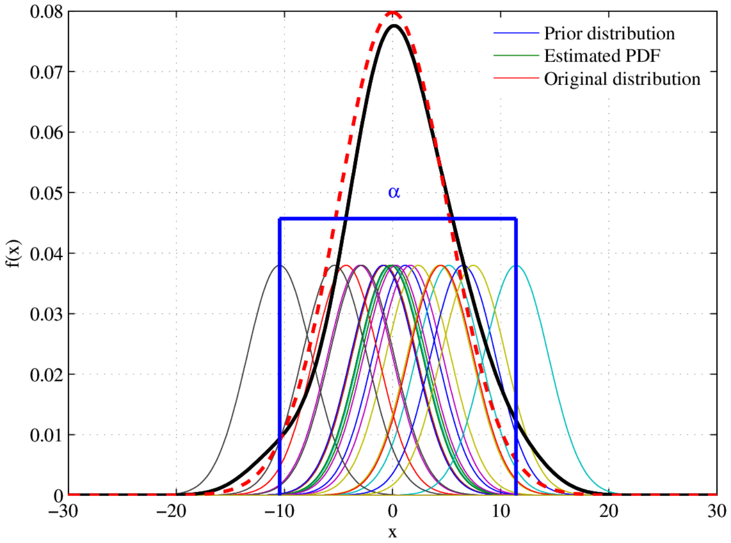

Figure 4.18 *Prior distribution acting as a window for marginalization integral; Gaussian kernels are more likely placed in between the integral limits: an intuitive illustration*

The last illustrative example deals with two mode density estimation from a sparse dataset (for more detail see chapter 4.6). In this case, the prior assumption (black thick line) deviates significantly from the original distribution (thick dashed line) and even from the estimated one (thin solid line). However, the PDF estimation is very close to the reference result obtained by cross-validation (CV) criterion. In this way, we want to demonstrate robustness of the EV against the prior assumption. The example will be further discussed in section 5.3.

Likelihood decomposition

The likelihood function to be integrated consists of the product of the sums over the entire training set $\mathcal{D}$:

$$p(\mathcal{D}|\mathcal{E},\sigma,\mathcal{H}) = \frac{k^m}{m^m}\prod_{i=1}^{m}\sum_{j=1}^{m} exp\left[-\frac{(\mathbf{Q}_j-\mathbf{P}_i)^T(\mathbf{Q}_j-\mathbf{P}_i)}{2\sigma^2}\right], \tag{4.53}$$

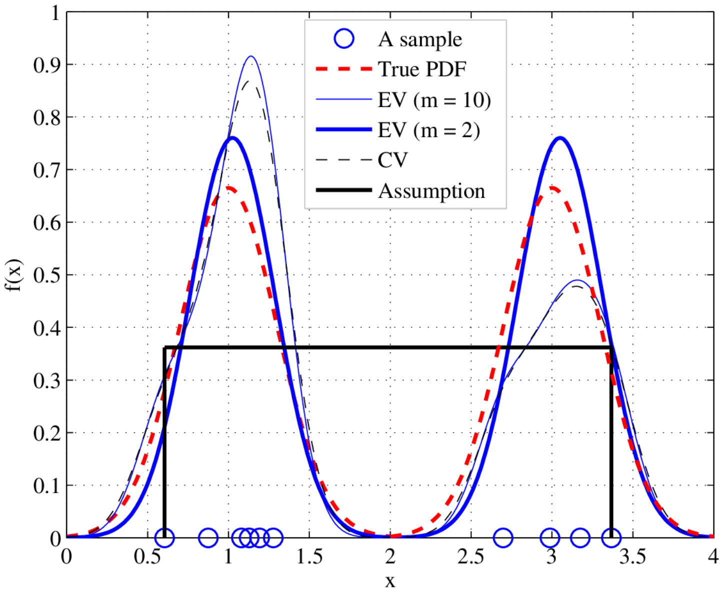

Figure 4.19 *Kernel density estimation driven by the EV criterion; a uniform prior distribution (black thick solid line) is very rough assumption on the original PDF (red thick dashed line); the EV criterion leads to the similar generalization (thin solid line) like the CV criterion (black thin dashed line) in the case of all kernels activated (m = 10).*

where each $\mathbf{Q}_j \in \mathcal{E}$ is to be marginalized out, m denotes the number of the patterns, and $k = (2\pi)^{-n/2}\sigma^{-n}$ normalizes the estimated n-dimensional distribution. Equation (4.53) can be rewritten as the sum of the products so that the marginalization integral from equation (4.52) is partitioned as the sum of the product integrals:

$$p(\mathcal{D}|\sigma,\mathcal{H}) = \sum_{k=1}^{np} \int_{\hat{\mathcal{E}}} G_k(\mathcal{E})\mathrm{d}\mathcal{E} \tag{4.54}$$

Each product $G_k(\mathcal{E})$ is in the form of multivariate Gaussian distribution since the k-th product $G_k(\mathcal{E})$ can be expressed in the following way

$$G_k(\mathcal{E}) = g_{n_1}(Q_1)g_{n_2}(Q_2)\cdots g_{n_s}(Q_s), \tag{4.55}$$

and each particular $g_{n_i}(Q_i)$ has the following form (see A.2)

$$\begin{aligned} g_{n_i}(Q_i) &= exp\left[-\frac{(P_1-Q_i)^2+(P_2-Q_i)^2+\cdots+(P_n-Q_i)^2}{2\sigma^2}\right] \\ &= \beta_{n_i}\cdot exp\left[-\frac{n_i\cdot(\Phi_n-Q_i)^2}{2\sigma^2}\right]. \end{aligned} \tag{4.56}$$

Here, Φ_n is an average value over n training samples appearing in a particular inter-product. While Φ_n denotes a coordinate, the constant β_n denotes the height of particular impulse and can be easily calculated:

$$\beta_n = exp\left[-\frac{(P_1 - \Phi_n)^2 + (P_2 - \Phi_n)^2 + \cdots + (P_n - \Phi_n)^2}{2\sigma^2}\right]. \tag{4.57}$$

Two important facts are to be considered. The first one, we are able to express all of the products resulting from equation (4.53) by scaled exponentials in standard form. Thus, we are able to find a suitable analytical approximation for each inter-product simply by scaling the approximative integrable form of a standard exponential (see the next section). The second note, since a high number of training samples m leads to an extremely high number of products np, it is essential to neglect the non-contributing ones. We can assume that a lot of the likelihood components do not contribute to the entire integral significantly due to a high variance between the samples (see equation (4.57)) or small width of the high order inter-products (see (4.56)).

Analytical approximation

The *Local approximation* approach can be employed since each integrand of the decomposed likelihood function is an impulse which contributes to the entire integral only in the vicinity of its coordinates. Then it can be locally approximated by a polynomial and analytically integrated over the specific area of a particular impulse where the polynomial approximation is always valid.

In the previous section, we proposed expressing all of the likelihood components separately as Gaussians. Thus, the task of this paragraph is to find an efficient local approximation of the Gaussian function. Obviously, we can define two conflicting criteria: the highest accuracy and the lowest order of an approximating polynomial.

First, the Taylor series was employed to approximate the Gaussian. This first attempt is illustrated in Figure 4.20. The order of the Taylor polynomial was increased up to 12 which significantly improved its accuracy. On the other hand, since the products can be represented by relatively wide Gaussians (depending on the smoothing parameter σ) the Taylor approach is not suitable for our purpose.

The second attempt was based on Lagrange polynomials. Since we know the exact position and width of each particular product, we can sample the local area in the sense of a smoothing parameter. Thus, we sampled the original product within the interval $< -3\sigma; +3\sigma >$ and observed the precision. We argue that the main contribution of each pulse lies within this interval. However, the method is not restricted by any size of the local area.

Notice that we compare the estimators relating to our purpose - to approximate the Gaussian within a specific interval. As can be seen, the Lagrange approach has better precision and lower order polynomials in our case. Still, there are oscillations

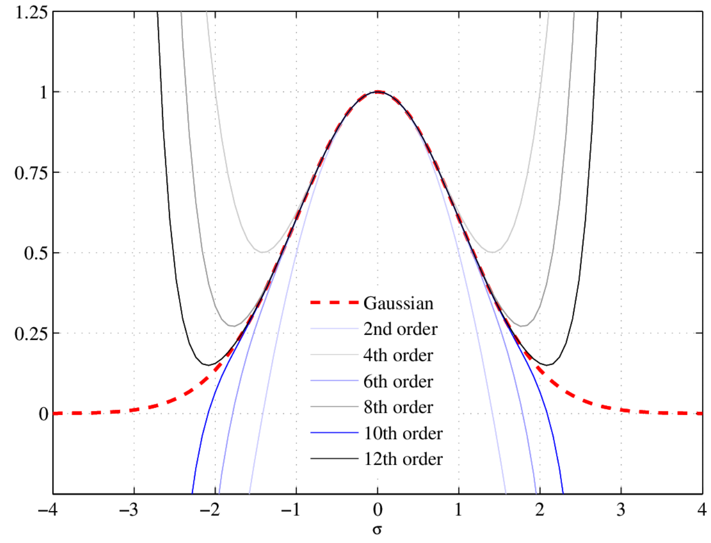

Figure 4.20 *On the precision of Taylor series approximation*

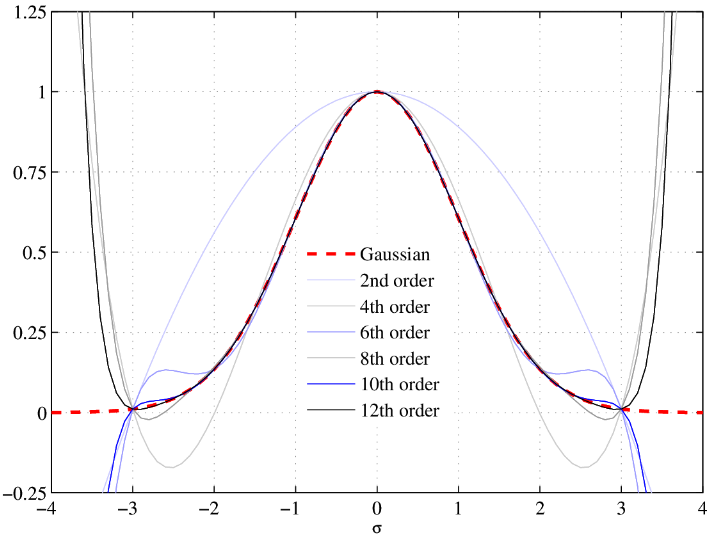

Figure 4.21 *On the precision of Lagrange approximation*

recognized for sparse sampling (low order). To suppress these oscillations and simultaneously to decrease the order of the polynomial, the Chebyshev approximation can be applied. The last parametric study is dictated to the best approximation approach which can be justified by observing Figure 4.22.

Finally, as discussed above, most of the products are distributed within the interval defined by the prior. This is due to the mean value Φ_n in equation (4.56) defining coordinates of the products rather to the center of the prior. Thus, it is worthwhile to substitute each particular inter-product integral by the Gaussian

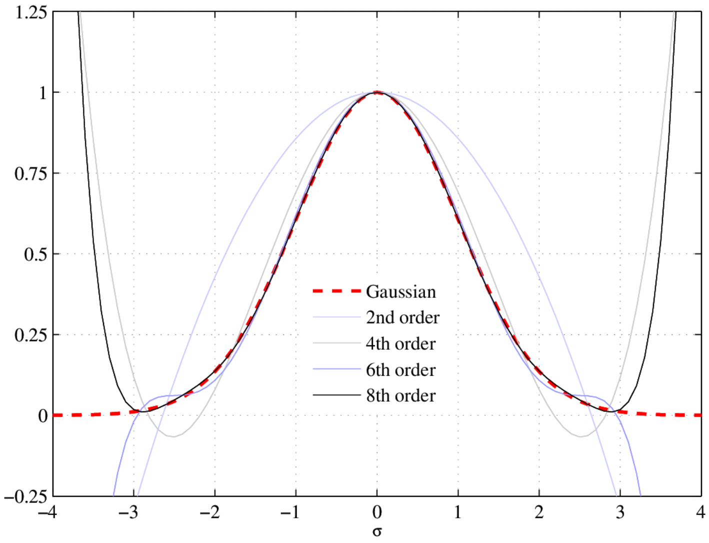

Figure 4.22 *On the precision of Chebyshev approximation*

integral:

$$\int_{Q_a}^{Q_b} g_n(Q_i)\mathrm{d}Q_i = \sqrt{\frac{2\pi}{n}}\sigma\beta_n, \tag{4.58}$$

where Q_a and Q_b are the integration limits defined by the prior. This approximative approach avoids the need of the local approximation step and decreases the computational demands discussed in the next section.

Computational issue

As has been already mentioned, the crucial problem of the proposed approach is the high number of the products $G_k(\mathcal{E})$ which are to be integrated. Basically, the total number of products $np = m^m$ to be considered is computationally intractable. This is the original problem complexity which will be called *Level 0*. Fortunately, a lot of the possibilities how to reduce the computational demands can be found.

The first approach employs combinatorial symmetry in the entire set of the products obtained by likelihood decomposition. Due to the same integral limits in (4.54) for all of the parameters, a smaller number of subsets $ns << np$ consisting of the products with the same contribution to the EV is needed to enumerate. For instance, in the case of two samples, the likelihood function can be decomposed into four products (see (4.47)). Then, it is required to enumerate only two product integrals thanks to the following expression

$$\begin{aligned}\int_{\hat{\mathcal{E}}} p(\mathcal{D}_2|\mathcal{E},\sigma,\mathcal{H})\cdot \mathrm{d}\mathcal{E} &= 2\cdot\int_{Q_{1,a}}^{Q_{1,b}}\int_{Q_{2,a}}^{Q_{2,b}} exp\left[-\frac{(P_1-Q_1)^2+(P_2-Q_2)^2}{2\sigma^2}\right]\mathrm{d}Q_1\mathrm{d}Q_2\\ &+ 2\cdot\int_{Q_{1,a}}^{Q_{1,b}}\int_{Q_{2,a}}^{Q_{2,b}} exp\left[-\frac{(P_1-Q_1)^2+(P_2-Q_1)^2}{2\sigma^2}\right]\mathrm{d}Q_1\mathrm{d}Q_2.\end{aligned} \tag{4.59}$$

The number of identical products in the j-th sub-set $\#G_j$ can be calculated:

$$\#G_j = \frac{m!}{(\sum_{i=1}^{s}(n_i-1))!}, \tag{4.60}$$

where n_i is the number of merged training samples assigned to the i-th free parameter Q_i and s is the number of the inter-products (see equation (4.55)). All of the simulations given in section 4.6 exploit the mentioned combinatorial symmetry of likelihood. This technique (*Level 1*) leads to an exact simplification of *Level 0.* The unique subsets can be found by backtrack algorithm [49], which is required to be executed once for all training sets with fixed m and for all values of smoothing parameter σ. Then, the EV (evidence) can be calculated according to the following expression

$$p(\mathcal{D}|\sigma,\mathcal{H}) = \sum_{k=1}^{ns}\#G_k\cdot\int_{\hat{\mathcal{E}}} G_k(\mathcal{E})\mathrm{d}\mathcal{E}, \tag{4.61}$$

where $G_k(\mathcal{E})$ is the product belonging to the k-th unique sub-set.

Further simplification (*Level 2*) can be achieved by restricting the number of merged samples n_i in a product. According to equation (4.56), an inter-product incorporating a high number of samples doesn't contribute significantly to the EV so that a particular unique class of products can be omitted in equation (4.61).

Finally, it is possible to substitute each particular inter-product integral by the Gaussian integral which avoids the need of the local approximation step as has been mentioned above. Incorporating this approximation, we reach computational simplification of *Level 3.*

Figure 4.23 depicts, how the described simplification influences EV estimation of the model illustrated in Figure 5.33. The EV was computed for all of the ns classes first. Then, the number of the unique subsets was reduced by restricting the number of merged samples up to $max(n_i) = 2$ according to *Level 2.* Simultaneously, all of the inter-product integrals were approximated by the Gaussian integral (*Level 3*) and the computed normalized EV was compared with the CV criterion. As can be clearly observed in Figure 4.23, the EV form a smooth function exhibiting one global maximum. Furthermore, the maxima for cross-validation and EV criteria leads to the same generalization of the KDE (Kernel Density Estimator) model (see Figure 5.33) even in the case of the highest level of likelihood simplification (*Level 3*).

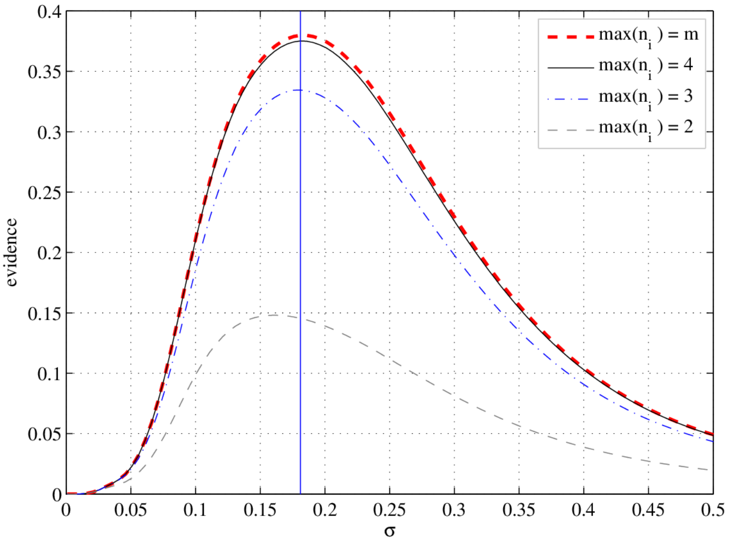

Figure 4.23 *The EV criterion calculated by simplified marginalization integral (Level 3) considering various degrees of Level 2 approximation; the EV criteria are depicted by thin lines, reference cross-validation is denoted by thick dashed line.*

To conclude this subsection, let us give a clear picture about the simplification efficiency by listing computation time of the EV in Figure 4.23 for all of the levels: t_0-intractable (*Level 0*), $t_1 = 243$ s (*Level 1*), $t_2 = 25$ s (*Level 2*), $t_3 = 2$ s (*Level 3*).

Dimensionality issue

One of the advantages of the proposed method is the simplicity of multi-dimensional estimation. The ideas mentioned above can be directly applied to general r-dimensional case since we can rewrite equation (4.54) as follows:

$$\begin{aligned} p(\mathcal{D}|\sigma,\mathcal{H}) \;=\; & \sum_{k=1}^{ns} \#G_k \cdot \int_{\hat{\mathcal{E}}_1} G_{k,1}(\mathcal{E}_1,\sigma_1) \int_{\hat{\mathcal{E}}_2} G_{k,2}(\mathcal{E}_2,\sigma_2) \\ & \cdots \int_{\hat{\mathcal{E}}_r} G_{k,r}(\mathcal{E}_r,\sigma_r) \cdot \mathrm{d}\mathcal{E}_1 \mathrm{d}\mathcal{E}_2 \cdots \mathrm{d}\mathcal{E}_r, \end{aligned} \tag{4.62}$$

where $\hat{\mathcal{E}}_i$ is the i-th subspace corresponding to the i-th dimension of parameter vector $\mathbf{Q}$ and σ_i denotes the i-th width of the Gaussian kernel corresponding to the i-th dimension of the PDF to be estimated.

It is possible to integrate each dimension separately so that we can employ our strategy derived above. Then, the entire integral is obtained by multiplying all of the particular one-dimensional integrals. Notice that we still need to decompose the likelihood function first, process each product separately, and finally sum the product integrals. Moreover, within the proposed method, it is possible to consider various kernel widths over the dimensions. Finally, computational complexity grows linearly with increasing problem dimensionality due to equation (4.62).

4.6 Conclusions on probabilistic neural networks

The previous chapter described a novel comprehensive criterion for evaluating probabilistic neural networks employing the Parzen window. The key concept is based on the Bayesian framework which allows us to develop a complex objective incorporating model fitting, generalization, regularization, and structural change in a neural model.

The procedure of obtaining the EV criterion can be introduced in five fundamental points:

- Selection and the effect of the prior distribution - EV is not very sensitive to the prior
- Likelihood decomposition and its approximation - a proposed approach to solving the marginalization integral
- Local approximation - likelihood can be locally approximated by the Chebyshev polynomial or the Gaussian integral can be employed
- Computational complexity - a crucial point of the Bayesian approach, it can be significantly reduced by the presented combinatorial and likelihood approximations
- Multi-dimensional issue - developed criterion naturally tackles an arbitrary dimensional problem.

Three numerical examples (see 5.3), characterized by different kinds of generalized model selection problem will be further presented to judge the EV criterion. In all of the cases, the developed criterion outperforms the approximated MISE criterion called cross-validation. Neural models selected by the EV exhibit better generalization from both the subjective and objective points of the view.

Although the initial computational complexity is unacceptable, efficient simplistic procedures can be found. The author can see the main contribution of the chapter lying in versatility and performance of the method, original integration procedure, combinatorial approach to the reduction of the computational complexity, and local analytical approximation of the decomposed likelihood.

All of the results were submitted for publication in [5] and convince the author for further research of this particular kind of approach to neural inference.

5 Numerical validation

5.1 Validation strategy

In this section, validation strategies for both the Boltzmann machine and the EV (evidence) approaches are to be introduced. Described validation approaches will be further discussed in sections 5.2 and 5.3, which will be covered by numerical simulations and practical examples.

Boltzmann machine

Section 5.2 inspects performance and reliability of the IN simplification based on the Boltzmann machine. Generally, the goal is to validate all ideas presented in chapters 1.2 and 2.5. In our case, a validation test-case should fulfil three fundamental criteria:

- The validation test-case should correspond to a real-world problem to inspect applicability of a proposed method. Thus, a bank of various impedance patterns is compiled in order to synthesize a versatile set of various impedance networks.
- More various problem instances should be introduced to ensure reliability and robustness of the proposed approach. Impedance patterns within the bank are randomly combined so that different impedance networks to be simplified are synthesized.
- A reference conventional method should be also applied to compare practical performance of the novel method. The proposed Boltzmann machine is compared with the simulated annealing technique and genetic algorithms. Unlimited parallelism of the BM (Boltzmann Machine) is emulated and computational time is discussed.

Evidence criterion

Section 5.3 demonstrates versatility of the developed EV criterion for probabilistic and general regression neural networks. Three different computer experiments are presented. Each particular task listed below is represented by a specific kind of the model selection problem which was solved via the proposed EV criterion:

- The first example deals with probability density estimation and structural change of the KDE (Kernel Density Estimator). The model EV is employed to search both the optimal smoothing and optimal number of radial neurons (kernels).
- The second problem lies in separation of intertwined spirals using the PNN (Probabilistic Neural Network) structure. Here, proposed criterion indicates the optimal density estimation incorporated in PNN structure forming the optimal class boundaries between intertwined spirals.
- The last example illustrates generalization capability of the GRNN (General Regression Neural Network) in noisy training data. The GRNN complexity is driven via the EV criterion.

5.2 Boltzmann machine simplifier

5.2.1 Validation test-case

As mentioned, the aim of this section is to validate the proposed simplification method in the most possible objective way. Thus, the bank of the various access impedances is compiled using three fundamental electronic circuits: simple connector model, serial and parallel L-C resonators. Parameters of the elementary circuits are varied so that a set of impedance patterns is obtained. Notice that the BM operates with measured or simulated impedance samples. This is the reason for using term pattern. Then a diverse set of impedance networks can be synthesized using combination of various impedance patterns and versatile set of simplification instances can be obtained.

Each impedance pattern consists of magnitude and phase sampled over a specific frequency interval. Three used elementary circuits are summarized in figure 5.24 and followed by analytical expressions for each particular input impedance. The bank of impedance patterns used for the testing procedure is illustrated in figures 5.25 - 5.27. In figure 5.28 equivalent impedance pattern of an exemplar test IN is depicted. Here, all methods were tuned to return a reduce IN consisting of the same number of the elements.

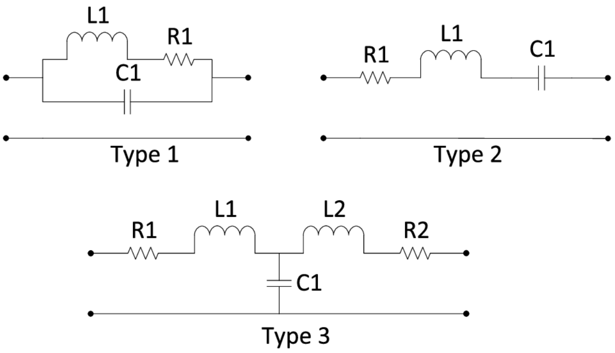

Figure 5.24 *Fundamental impedance circuits for an impedance network synthesis*

Type 1:

$$Z_{T_1}(s) = \frac{R_1 + s \cdot L_1}{1 + s \cdot (C_1 R_1) + s^2 \cdot (C_1 L_1)} \tag{5.63}$$

Type 2:

$$Z_{T_2}(s) = \frac{1 + s \cdot (C_1 R_1) + s^2 \cdot (C_1 L_1)}{s \cdot C_1} \tag{5.64}$$

Type 3:

$$\begin{aligned} Z_{T_3}(s) \quad &= \quad \frac{R_1 + R_2 + s \cdot (L_1 + L_2 + C_1 R_1 R_2)}{1 + s \cdot (C_1 R_2) + s^2 \cdot (C_1 L_2)} \\ &+ \quad \frac{s^2 \cdot (C_1 L_2 R_1 + C_1 L_1 R_2) + s^3 \cdot (C_1 L_1 L_2)}{1 + s \cdot (C_1 R_2) + s^2 \cdot (C_1 L_2)} \end{aligned} \tag{5.65}$$

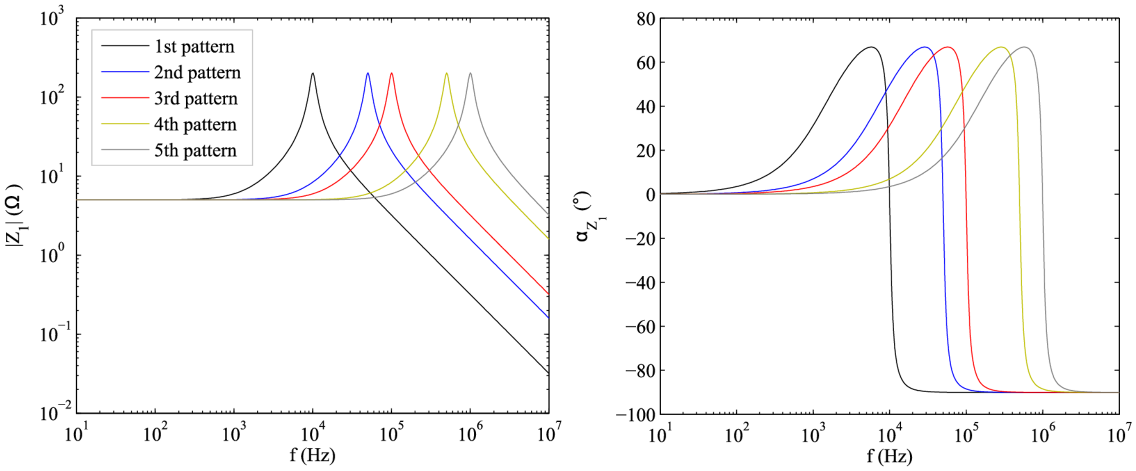

Figure 5.25 *Type 1 impedance pattern*

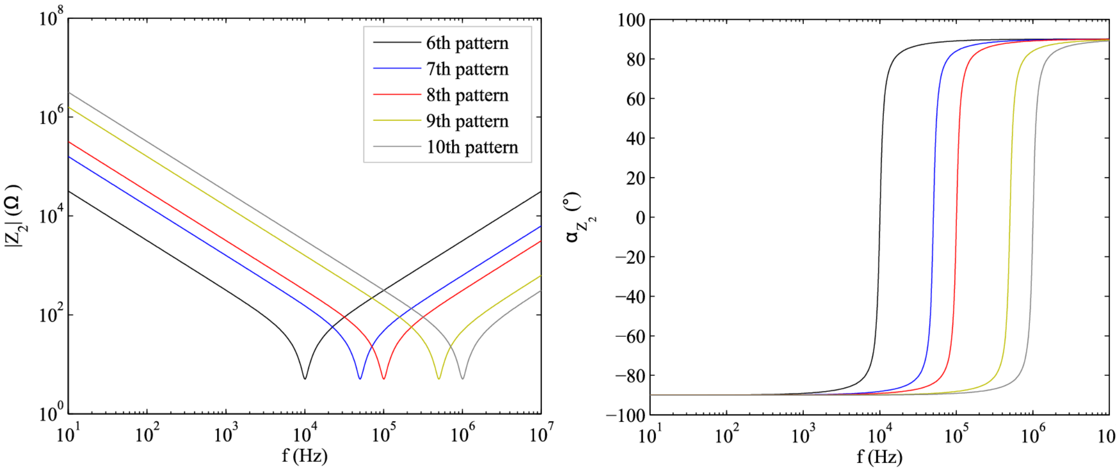

Figure 5.26 *Type 2 impedance pattern*

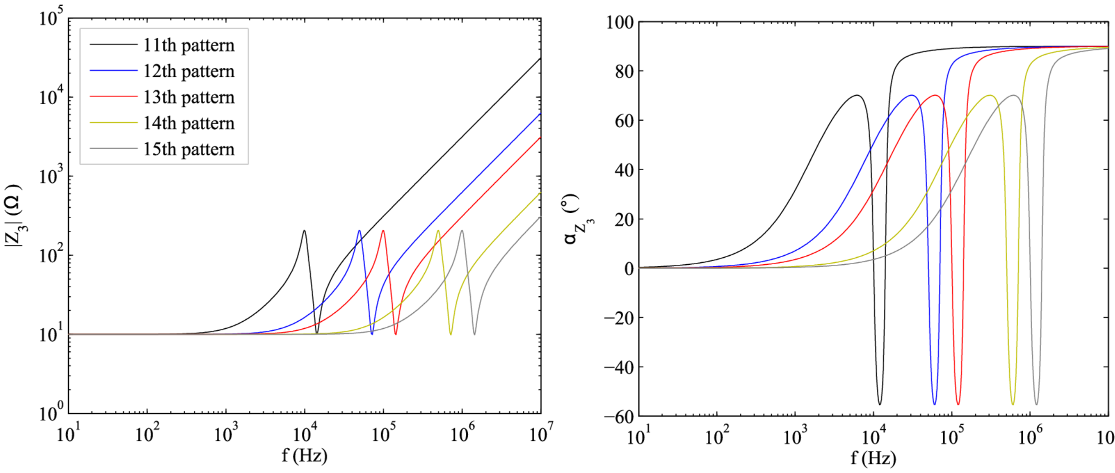

Figure 5.27 *Type 3 impedance pattern*

5.2.2 Performance

Performance of the developed simplifier was compared with the simulated annealing (SA), and genetic algorithm (GA) techniques. In order to evaluate and compare all methods, five INs with increasing level of complexity where examined:$L = 100$, $L = 200$, $L = 300$, $L = 500$, $L = 1000$. Each particular algorithm was executed ten times to measure averaged running time $\bar{t}$ and maximal/minimal reached cost function O_{max}/O_{min}. Notice that it is not possible to compare the methods by a number of objective function enumerations, since the BM evaluates its unique function (energy difference) (see (3.19)). Unfortunately, its ambiguous to compare various optimisation methods due to their hyper-parameters. An expert knowledge is required to configure each algorithm in the best way. However, it is possible to

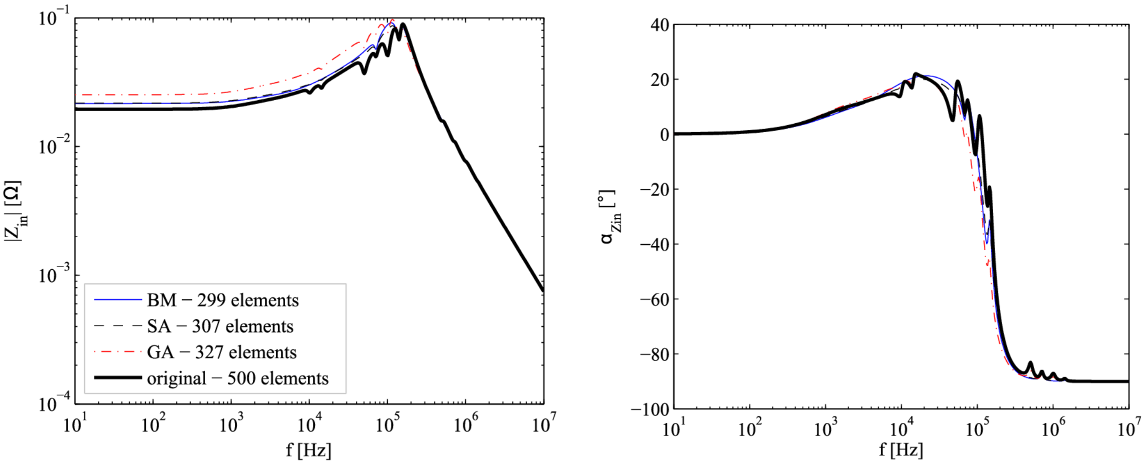

Figure 5.28 *A comparison of the three simplifiers under the test. The original IN consisted of 500 elements.*

tune the algorithms for a particular level of simplification and observe the deviation of a reduced IN from the original one. Table 5.2 summarizes obtained results over the various problem instances.

	BM ($\beta = 0.8$)			**SA** ($\beta = 0.8$)			**GA**		
L	$\bar{t}$ (s)	O_{min}	O_{max}	$\bar{t}$ (s)	O_{min}	O_{max}	$\bar{t}$ (s)	O_{min}	O_{max}
100	**0.1**	26.6	**26.6**	**4.5**	26.6	**70.6**	12.8	30.8	39.5
200	**0.3**	57.0	**57.0**	**23.6**	56.5	**92.6**	37.9	70.7	225.4
300	**0.6**	86.7	**86.7**	**37.2**	86.0	**88.6**	55.0	138.2	222.7
500	**1.4**	164.7	**164.7**	**160.3**	167.4	**208.2**	157.1	417.8	583.1
1000	**4.6**	348.8	**348.8**	**1216.1**	354.8	**535.7**	1367.5	1747.7	4058.2

Table 5.2 *A comparative study of three combinatorial simplifiers*

Dealing with hyper-parameters, the cooling schedule for the BM was defined according to results from chapter 3.3. Then, the schedule for the SA method was derived such that, similar semi-optimal solution to the BM could be found. Speed of cooling $\beta = 0.8$ was held the same for both the BM and SA routines. Comparison of the BM and SA is essential due to the similar principals occurring within both approaches. However, the same procedure was performed by the GA method. A number of individuals was systematically increased from 80 to 300 according to increasing the number of elements L. Decimation of population was set as selection procedure due to its more reliable behaviour (see section 5.2.3).

As can be seen in table 5.2, the fastest and most accurate method is the developed BM simplifier. Efficiency of proposed BM lies in very simple function (energy difference) to be enumerated within the annealing process in comparison with the objective function required by the other methods. Computational time corresponding to the BM incorporates the mapping procedure expressed in (2.14). Moreover, the BM always converges very close to optimal solution which proves the correctness of the proposed cooling schedule.

In Figure 5.29, computational time required by the BM to converge versus a number of the IN elements is depicted. A cubic polynomial fits the measured values well. Thus, we can say that the BM annealed according to the proposed cooling schedule exhibit approximately $O(L^2)$ computational complexity.

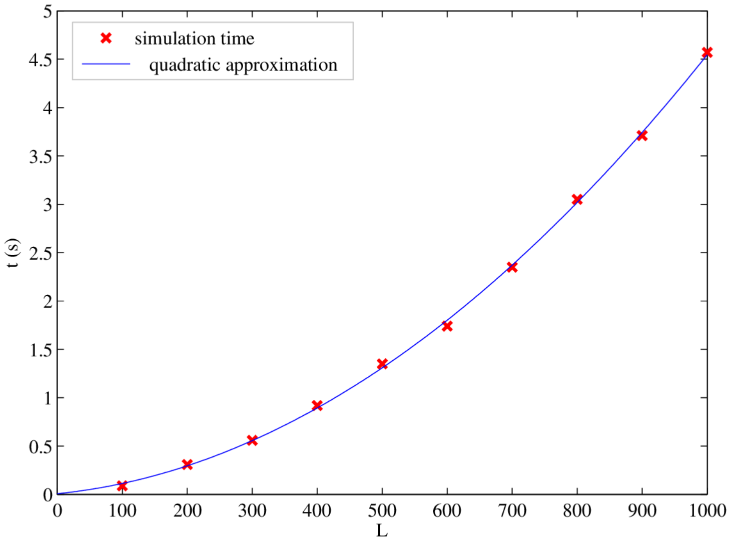

Figure 5.29 *Computational time depending on the number of IN elements*

5.2.3 Reliability

In order to compare efficiency and robustness of all algorithms, simple statistical evaluation was provided. First, all methods were executed 1000 times to simplify the same IN. Then, it was possible to estimate a probability distribution of the results (see Figure 5.30). It is obvious, the narrower distribution was obtained by a more reliable algorithm and the lower mean value was obtained by a more efficient algorithm. The statistical experiment was performed on the simple trial IN consisting of the $L = 100$ elements. Each particular simulation was executed under random initial conditions.

While the first test was performed in one particular IN simplified 1000 times, the second test dealt with 25 various problem instances (different INs). Each particular IN was simplified using all three algorithms under the test. In Figure 5.31, the most probable results estimated using each algorithm can be observed over the 25 problem instances. Performance is represented by median values denoted by lines, reliability is represented by intervals covering 99% of possible results. In Figure 5.31, each particular errorbar indicates 99% of observed results obtained by the corresponding algorithm.

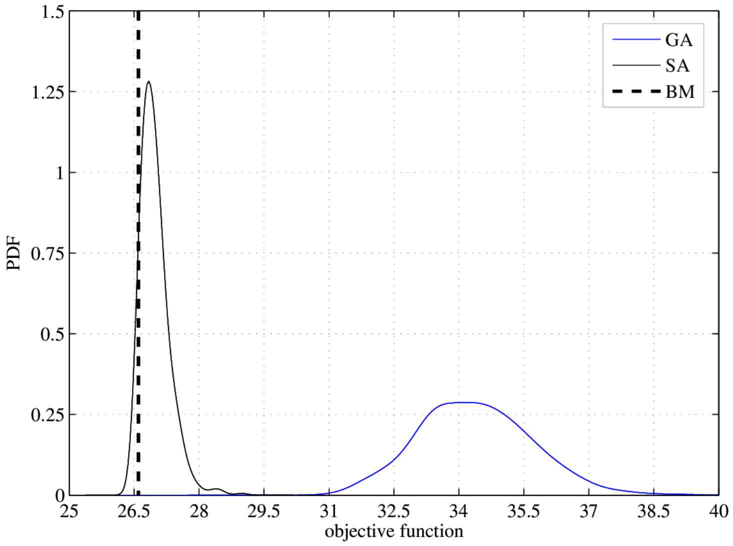

Figure 5.30 *Probability distribution of the results provided by the GA, SA, and BM; the BM always converges to the same solution which is indicated by the vertical dashed line*

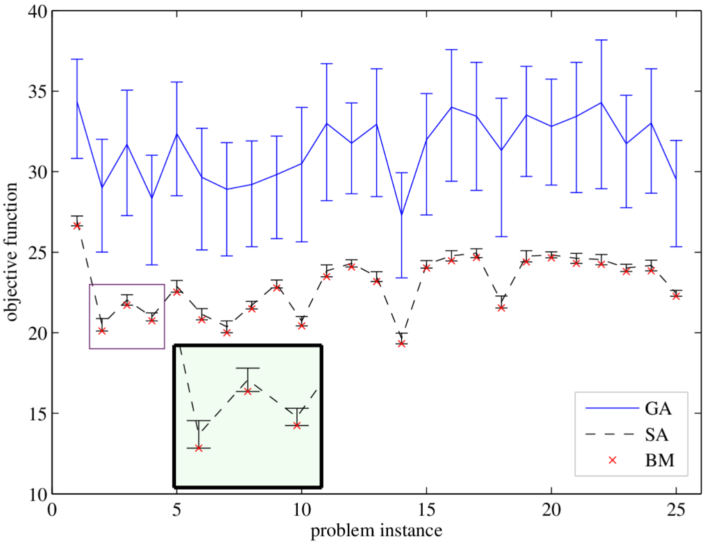

Figure 5.31 *Most probable values of the objective function reached by a particular method (lines); the errorbars indicates 99% of observed results obtained by the corresponding algorithm; the results obtained by the BM are denoted by cross markers*

Figure 5.31 illustrates both the performance and reliability of the algorithms under the test over 25 different problem instances. Generally, the GA algorithm exhibits low quality in comparison with both the SA and BM methods. As can be observed in Figure 5.31, the BM provides solutions exhibiting the same quality as 0.5% of the best results provided by the SA method, which is convincing proof of the BM efficiency.

5.2.4 Parallel emulation

In section 3.4 parallel execution of the BM was discussed from the theoretical point of view. If the BM is executed in parallel, more stochastic neurons can simultaneously change their states, which may lead to erroneously calculated difference in the energy function. Unfortunately, the simplification problem is represented by the full connected pattern (see Figure 2.1). Thus, the limited version of parallelism can not be applied since no independent set of neurons can be found. In the case of the full connected BM, full parallelism lead to significant deceleration in convergence due to high occurrence of the erroneous calculations. The parallel BM was simulated using the same test-case as used in section 5.2.1. Computational time was measured over the annealing process and the mapping procedure was excluded to emphasize the time of convergence. Both the parallel and sequential machines retuned the same solution to the problem as listed in table 5.2. Thus, table 5.3 compares execution time of the sequential and parallel BM. The total simulation time measured for parallel BM was divided by $L2/3$ to take into account the parallel activation of the $L2/3$ neurons and judge efficiency of the parallel approach.

	sequential	parallel
L	$\bar{t}$ (ms)	$\bar{t}$ (ms)
100	12.4	36.0
200	29.9	2100.1
300	45.8	415.0
500	90.0	1236.1
1000	214.7	2604.0

Table 5.3 *A comparative study of the parallel and sequential execution of the BM; the computational time of sequential execution is listed in milliseconds.*

Measured execution time is much higher in the case of the parallel machine. Unfortunately, we have to conclude that any parallelism is not suitable for our purpose due to the full connected type of the BM and erroneously calculated energy differences.

5.3 Probabilistic neural networks

5.3.1 A structural change of KDE

A set consisting of ten samples (circles in figure 5.33) was drawn from a two mode probability distribution formed by two Gaussian distributions $\sigma_{orig} = 0.3$. The

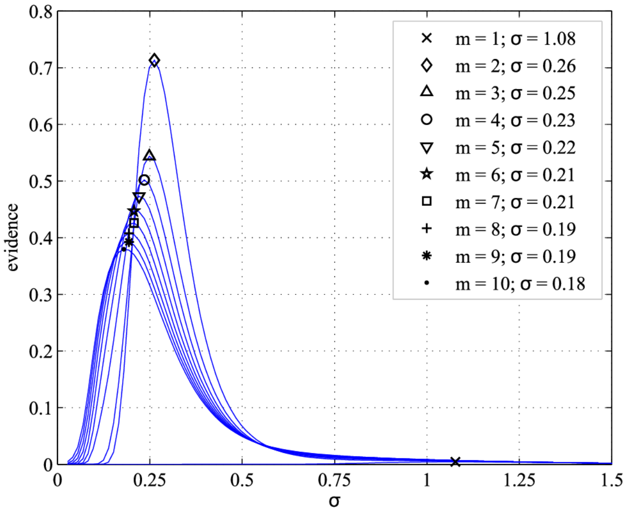

Figure 5.32 *Evolution of the EV criterion for various numbers of the kernels* $m = \{1, 2, \cdots 10\}$*; EV maxima are denoted by markers, each one corresponding to a particular KDE.*

task was to simultaneously tune the smoothing parameter and adjust the sufficient number of the kernels (radial neurons) so that the bias/variance equilibrium would be found. Figure 5.32 illustrates evolution of the EV according to a particular number of the kernels. Obviously, a single kernel $m = 1$ can not fit the data well due to a two modal character of the sample. This is also the reason for significantly higher value of the smoothing parameter $\sigma_{m=1} = 1.08$ resulting in the model with high bias. On the other hand, as the number of neurons increases $m > 2$ the maximum of EV decreases since the model exhibit higher variance as observed in Figure 5.33. The highest EV was observed in the case of the two kernels $m = 2$, which is convincing result since the original distribution consists of two Gaussians. If the number of the kernels increases the estimated σ decreases since more kernels cover the input space more easily.

Figure 5.33 summarizes three approaches to the kernel density estimation: evidence (EV) based selection, cross-validation (CV), and evidence based structural change. If the number of the kernels $m = 10$ equals the number of samples in the training set the EV selection criterion and cross-validation technique lead to a model with similar generalization. Notice that the *Level 3* simplification was employed in this simulation.

If structural change is incorporated, the proposed criterion selects the model with significantly better generalization: $\mathrm{MSE}_{CV} = 11.1 \cdot 10^{-3}$, $\mathrm{MSE}_{EV} = 2.9 \cdot 10^{-3}$. Here, the mean squared error was measured over the estimated and original PDFs in Figure 5.33. Furthermore, the selected model consist of five times less kernels in comparison with the original one.

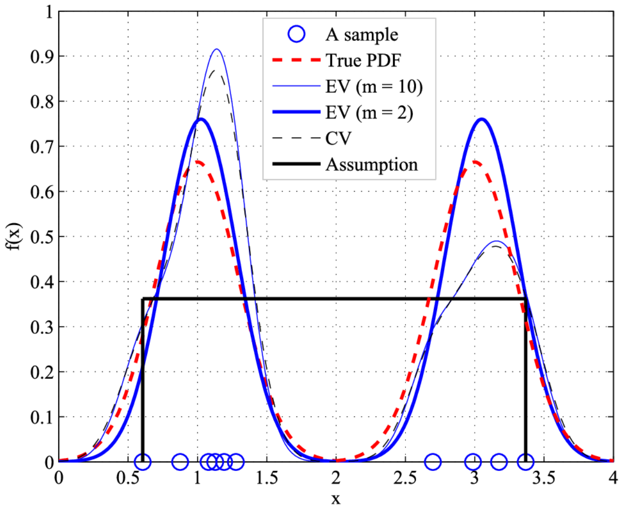

Figure 5.33 *Kernel density estimation driven by the EV criterion; a uniform prior distribution (black thick solid line) is very rough assumption on the original PDF (red thick dashed line); the EV criterion leads to the similar generalization (thin solid line) like the CV criterion (black thin dashed line) in the case of all kernels activated (m = 10); a model consisting of two kernels ($m = 2$) selected by the EV exhibits the best generalization (blue thick solid line).*

5.3.2 Classification task

The second experiment lies in separation of intertwined spirals which have been already used as an illustrative classification problem in [45]. The reason for addressing this particular task lies in its three characteristic points: strongly non-linearly separable problem, high variance in mutual distances of the neighbouring samples, clearly non-uniform distribution of the samples (strongly against the prior assumption in equation (4.52)).

A complete training set consisted of 4×10 training samples belonging to four classes. The training set for the first class is depicted in Figure 5.34 (left). The remaining three sets have been obtained rotating the initial spiral by $(i-1)\pi/2$, where i indicated a particular class. As mentioned, density of the training samples increases as we follow a spiral towards its center. Thus, the EV criterion was employed to find a compromise between precisely shaped density within area close to the center and smooth density within distant area, see Figure 5.34 (left). This equilibrium is strongly connected with shape of the decision boundaries formed by the PNN.

According to Figure 5.34 (right), one can argue that the obtained model will classify wrongly even the training samples within the spiral center. In this case, three points should be considered: a global smoothing parameter σ was considered, the EV approach incorporates uncertainty in training samples, and, generally, a model with worst fitting can exhibit better generalization.

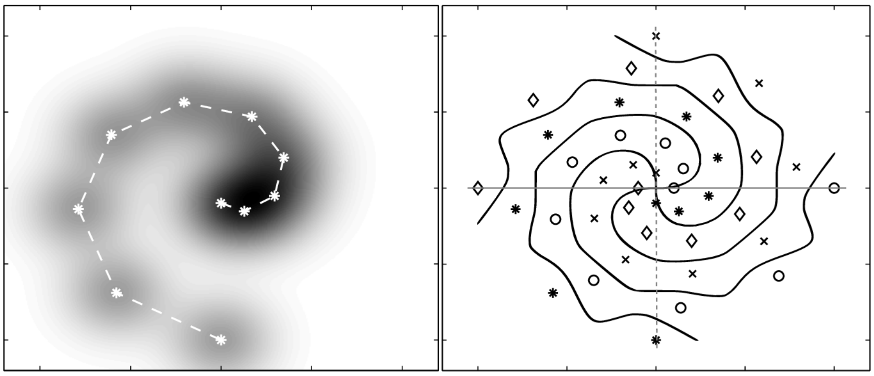

Figure 5.34 *A probability distribution estimated by PNN belonging to a particular spiral (left) and decision boundaries provided on the output of the PNN (right); each particular spiral training set is denoted by a unique type of marker; EV (evidence) based PNN model produces decision boundaries compromising between over-smoothed (CV) and over-fitted (NN) solutions.*

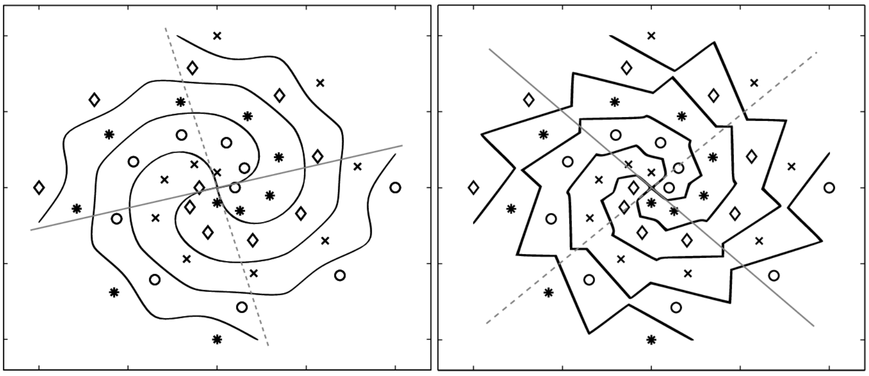

Figure 5.35 *Decision boundaries formed by two reference estimations: over-smoothed CV based PNN (left), over-fitted nearest neighbour (NN) estimate (right).*

In order to evaluate the proposed criterion, two more computations were provided. Figure 5.35 depicts decision boundaries formed by cross-validated PNN and by the nearest neighbour estimator. The nearest neighbour estimator exhibits very high complexity, precise fitting, and, consequently, very rough decision boundaries. On the other hand, if PNN complexity is driven via CV criterion, the obtained classifier leads to smooth boundaries but also to misclassification close to the center.

Two lines were created in parallel with decision boundaries in the center of each spiral so that the trend of the boundaries could be observed. As we can see in Figures 5.34 and 5.35, the EV solution ($\sigma_{ev} < \sigma_{cv}$) fits better the data than the CV approach and simultaneously exhibits worst fit than the nearest neighbour estimator. This is important conclusion, since one would expect rather smoother model in the case of EV criterion due to the effect of prior assumption (an uniform distribution). Despite of the prior, proposed EV criterion selected an equilibrium between over-fitting (too high complexity) and over-smoothing (too low complexity).

5.3.3 Approximation task

A training set for approximation test case consists of 50 samples obtained by sampling an original function depicted in Figure 5.36 (thick dashed line). The target is a noisy ($\sigma_{noise} = 0.75$) composite function including both of the smooth and rough parts, which is crucial for all model selection criteria. More specifically, an original function can be expressed: $f(x) = -20/(1 + x^2)$ on the interval $[-12, 3)$, $f(x) = sin[0.4\pi \cdot (x - 3)]$ within the range $[3, 10.5)$, and $f(x) = -20/[1 + (x - 13.5)^2]$ within $[10.5, 13.5]$.

The training set was partitioned onto five neighbouring sections and sufficient smoothing for each particular section was estimated. Basically, this approach to the GRNN simulation is similar to the one used in [45]. The EV was employed to find all smoothing parameters for the GRNN estimator while the cross-validation criterion was used as a reference method. Figure 5.36 shows that both criteria lead to compromise between fitting the rough parts of the original function and suppressing noise. The EV approach applies less smoothing than the CV criterion. Notice that this statement holds for all numerical examples given above and it always leads to better generalization.

Figure 5.37 depicts amount of bias and variance in the EV based and CV based models. Ten training sets were randomly generated to depict model complexities in Figure 5.37. As can can be observed, the CV criterion systematically selects smoother models exhibiting higher bias, which is not desirable if a function to be approximated exhibits complex behaviour.

In order to evaluate the EV criterion for the GRNN in a rigorous way, one hundred noisy training sets were generated and average MSE over all sets was measured for both of the EV and CV criteria. Each particular MSE measured deviance between the original function $f(x)$ and particular GRNN estimate over the entire approximated interval and for dense testing set, see Figure 5.37. The resulting errors clearly indicates that EV criterion selects more probable models: $\text{MSE}_{CV} = 1.73$, $\text{MSE}_{EV} = 1.02$.

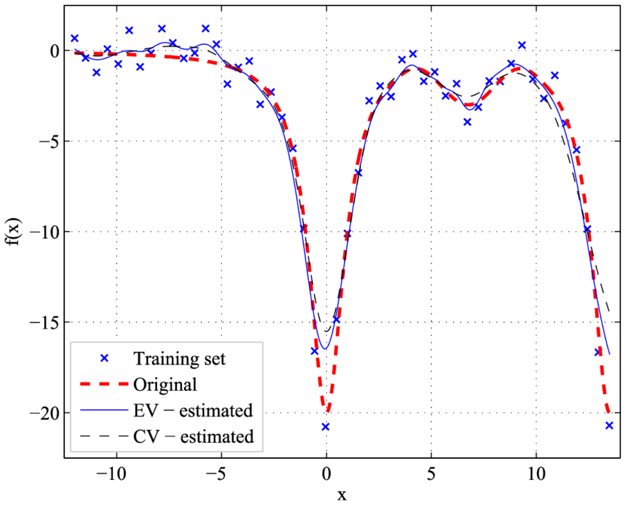

Figure 5.36 *Comparison of the EV based (solid thin line) and the CV based (dashed thin line) GRNN estimators recovering the original function (dashed thick line) from a training set corrupted by Gaussian noise (crosses).*

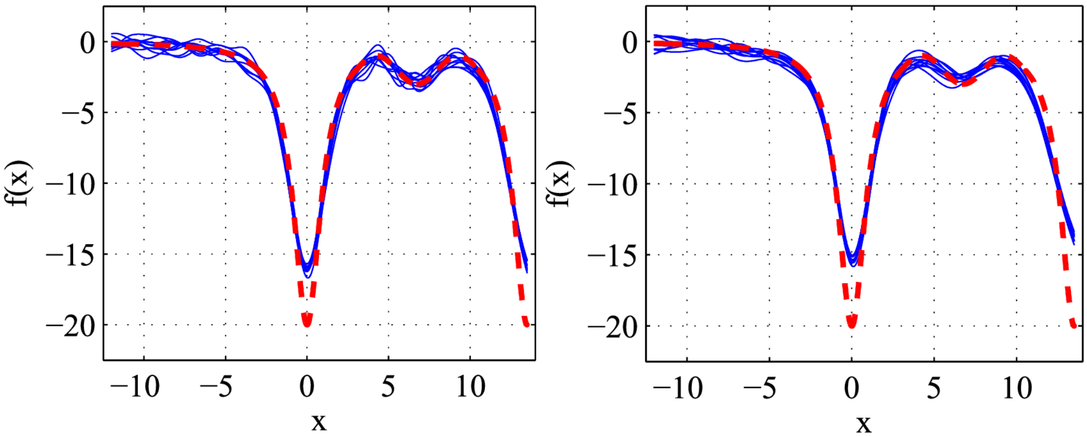

Figure 5.37 *Illustration of the bias and variance via ten training sets; the EV based estimation (left) results in lower bias (better fit of the rough parts) and more variance (more sensitive to noise), the CV based model (right) exhibits exactly the opposite behaviour.*

5.4 Discussion on numerical results

Let us to briefly discuss the obtained numerical results. Dealing with the BM combinatorial simplifier, experimental observation can be summarized:

- The mapping procedure was provided appropriately since the BM searched the minimum of cost function defined by the simplification task (see Figure 5.28).
- Execution time needed for convergence of the BM is approximately 200 times lower than the SA. Notice that the main portion of execution time is consumed by the mapping procedure (see Table 5.2).
- Computational complexity of the BM is approximately $O(L^2)$. This means that execution time rises with a square of the problem complexity and the developed BM can solve the simplification problem in polynomial time (see Figure 5.29).
- The BM always returns a solution better than 0.5% of the results provided by the SA method. The BM is the most reliable algorithm in comparison with SA and GA techniques (see Figure 5.31). This also proves correctness of the proposed cooling schedule.
- Parallel emulation of the BM exhibits much lower performance than the sequential form. It is due to fully connected pattern causing high occurrence of erroneously calculated difference in energy (see Table 5.3).

The EV criterion was tested on three different task incorporating the model selection problem:

- First, all kernels were used for the kernel density estimation. In this case the EV criterion leads to the similar model as the CV (cross-validation) criterion. Then, the kernels were iteratively removed and the highest level of the EV was estimated. The reduced KDE estimator leads to significant improvement in generalization in comparison with CV decision: $\text{MSE}_{CV} = 11.1 \cdot 10^{-3}$, $\text{MSE}_{EV} = 2.9 \cdot 10^{-3}$.
- Referring to the classification task, the EV criterion selected the classifier equilibrating between over-fitting (nearest neighbour criterion) and over-smoothing (cross-validation criterion) (see 5.34).
- The last experiment dealt with the approximation task. The EV and CV criteria were compared in terms of the mean squared error over multiple training sets. It is the criterion expressed in (4.44). The proposed EV criterion leads objectively to better generalization in comparison with CV criterion: $\text{MSE}_{CV} = 1.73$, $\text{MSE}_{EV} = 1.02$.

6 Conclusion

Even though all objectives and results have been independently discussed above, it is worthwhile to present a general point of the view. The author finds interesting to remark some positive and promising features of the ideas as well as their limitations and disadvantages.

First, its obvious that the simplification task can be solved very efficiently using the mapping procedure and the Boltzmann machine. Estimated computational complexity promises high efficiency of the developed method in solving large problems. The BM incorporates measured values of the impedance elements, which is an important practical feature. On the other hand, the proposed method was demonstrated on a simple circuit topology which makes the mapping not so complicated. The future research should be mainly focused on generalizing the mapping procedure to enable simplifying an arbitrary topology of an equivalent circuit. This step will probably require some change in the energy function of the BM and some structural change of the machine itself. However, if the generalized mapping problem was solved a robust synthesiser of equivalent circuits could be obtained. Such a method would incorporate both the equivalent circuit model performance (fitting) and the model complexity (generalization).

The main motivation for developing the evidence criterion was given by need of some comprehensive approach to the model selection problem. This problem is not usually addressed by the behavioural modellers but its crucial from the model generalization point of the view. As demonstrated above, the proposed criterion can address a wide spectrum of machine learning problems if its applied on the Parzen estimator. As mentioned above, the most crucial problem of the proposed method is its computational complexity. Even if some simplistic methods were proposed and discussed, the computational time rises to rapidly with extent of training data. This should be the first step in future research. A possible way is to find an alternative form of the kernel leading to analytically integrable likelihood. The positive result is the robustness of the criterion against the prior assumption, which was verified by numerical simulations. This aspect makes possible to use the evidence for regularizing solutions to the inverse tasks (e.g de-blurring problem, linearised inverse scattering problem) since the same phenomena occur within the field of these problems. Moreover, optimal and efficient selection of the KDE smoothing parameter is still open question in statistics and the evidence is a possible solution to the problem.

Bibliography

[1] Koudelka, V., Raida, Z.: 'Evaluation of Electromagnetic Immunity of Layered Structures by Neural Networks', IET Antennas, Microwaves & Propagation, vol. 5, pp. 482-489, 2011.

[2] Koudelka, V., Raida, Z., Tobola, P.:'Simple Electromagnetic Modelling of Small Airplanes: Neural Network Approach', Radio Eng., vol. 18, pp. 38-41, 2009.

[3] Koudelka, V., Svobodova, J., Raida, Z.: 'Impedance Network Simplification: A Combinatorial Optimization Approach', In Proceedings of ICEAA conference on Electromagnetics in Advanced Applications, Torino-Italy, September 2011.

[4] Koudelka, V., del Rio Bocio, C., Raida, Z.: 'Diffracted Image Restoration: A Machine Learning Approach', In Proceedings of ICEAA conference on Electromagnetics in Advanced Applications, pp. 931-934, Torino-Italy, September 2013.

[5] Koudelka, V., Raida, Z.: 'Evidence Based Selection Criterion for Probabilistic and General Regression Neural Networks', Neural Networks - ELSEVIER, submitted for publication.

---

[6] Goksu, H., Pommerenke, D.J., and Wunsch, D.C.: 'FDTD data extrapolation using multilayer perceptron (MLP)', In proceedings of IEEE International Symposium on Electromagnetic Compatibility, vol.2, pp. 735-737 vol.2, 18-22, August 2003.

[7] Goksu, H., and Donald, C.: 'Artificial Neural Networks and Neural Information Processing', Istanbul, Turkey: Springer, 2003.

[8] Hacib, T., Mekideche, M., and Ferkha, N: 'Computational Investigation on the Use of FEM and RBF Neural Network in the Inverse Electromagnetic Problem Identification', IAENG International Journal of Computer Science. Available: http://www.iaeng.org/

[9] Maris, T., Ekonomou, L., Fotis , G., Nakulas, A., and Zoulias, E.: 'Electromagnetic Field Identification Using Artificial Neural Networks', In proceedings of 8th WSEAS International Conference on Neural Networks, pp.84-89, June 19-21, 2007.

[10] Cook, S.: 'The millennium prize problems: The P Versus NP Problem', pp. 87-104, Clay Math. Inst.,2006.

[11] Aardal, K. et al: 'A Decade of Combinatorial Optimization', University Utrecht, 1997. Available: http://igitur-archive.library.uu.nl/.

[12] Michalewicz, Z., Fogel, D., B. 'How to Solve It: Modern Heuristic.', Berlin: Springer-verlag, 2000.

[13] Hochbaum, D., S. (ed.): 'Approximation Algorithms for NP-Hard problems', Boston: PWS Publishing Company, 1996.

[14] DEB, K. 'Multi-Objective Optimization using Evolutionary Algorithms', UK, Chichester: Wiley, 2001.

[15] Aarts, E., H., L. and Korst, J., H., M.: Simulated Annealing and Boltzmann Machines. New York: Wiley, 1988.

[16] Hopfield, J. and Tank, D.: Neural Computation of Deci-emarginE rgsions in Optimization Problems, Biol. Cybern. 52, pp.141-152, 1985

[17] Wilson, G., V., and Pawley, G., S.: 'On the Stability of the Travelling Salesman Problem algorithm of Hopfield and Tank', Byological cybernetics, vol. 58, no. 1, pp. 63-70, 1988.

[18] YUE, T., FU, L.: 'Ineffectiveness in Solving Combinatorial Optimization Problems Using a Hopfield Network: A New Perspective from Aliasing Effect', In proceedings of IEEE International joint conference on neural networks, San Diego, 1990.

[19] Asai, H., Onodera, K., and Ninomiya, H.: 'A Study of Hopfield Neural Networks with External Noises', In Proceedings of IEEE International Conference on Neural Networks, vol. 4, Perth, WA, 1995.

[20] Mandziuk, J.: 'Pulsed noise-based stochastic optimization with the Hopfield model', In Proceedings of International Conference on Neural Networks, Huston, TX, USA, 1997.

[21] Yamada, Y., Aihara, K., and Kotani, M.: 'Chaotic Neural Networks and The Traveling Saleman Problem', In Proceedings of International Join Conference on Neural Networks, Nagoya, Japan, 1993.

[22] Haykin, S.: Neural Networks: A Comprehensive Foundation - 2nd ed., Upper Saddle River, New Jersey: Prentice Hall, 1999.

[23] Aarts, E., H., L. and Korst, J., H., M.: 'Boltzmann machines as a model for parallel annealing', Algorithmica, vol. 6, no. 6, pp. 437-465, 1991.

[24] Kirkpatrick, S., Gelatt, C., D., Jr., and Vecchi, M., P.: 'Optimization by Simulated Annealing', Science 13, vol. 220, no. 4598, pp. 671-680, May, 1983.

[25] Aarts, E., H., L. and Korst, J., H., M.: 'Boltzmann machines for travelling salesman problems', European Journal of Operational Research, vol. 39, no. 1, pp. 79-95, March, 1989.

[26] De Gloria, A., Faraboschi, P., Olivieri, M.: 'Block placement with a Boltzmann Machine', IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 13, no. 6, pp. 694-701, June, 1994.

[27] d'Anjou, A. at al: 'Solving satisfiability via Boltzmann machines', IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, no. 5, pp. 514-521, May, 1993.

[28] Cook, S., A. and Mitchell, D., G.:'Finding Hard Instances of the Satisfiability Problem: A Survey',In Satisfiability Problem: Theory and Applications, Providence, RI: Amer, 1997.

[29] Ramanujam, J., Sadayappan, P.: 'Mapping combinatorial optimization problems onto neural networks', Information Sciences, vol. 82, no. 2, pp. 239-255, January, 1995.

[30] Tesar, B., B., Kapenga, J., Trenary, R.: 'A Boltzmann machine solution of the traveling salesperson problem: a study for parallel implementation', In Proceedings of IEEE Region 10 International Conference TENCON, Bombay, India, November, 1989.

[31] Specht, D., F.: 'Probabilistic Neural Networks and the Polynomial Adaline as Complementary Techniques for Classification', IEEE Transactions on Neural Networks, vol. 1, pp. 111-121, March, 1990.

[32] Tripathy, M., Maheshwari, R.P., and Verma, H.K.: 'Probabilistic neural-network-based protection of power transformer', IET Electric Power Applications, vol. 1, no. 5, pp. 793-798, September, 2007.

[33] Mishra, S., Bhende, C.N., and Panigrahi, B.K.: 'Detection and Classification of Power Quality Disturbances Using S-Transform and Probabilistic Neural Network', IEEE Transactions on Power Delivery, vol. 23, no. 1, pp. 280-287, January, 2008.

[34] Samantaray, S., R., Panigrahi, B., K., and Dash, P., K.: 'High impedance fault detection in power distribution networks using time-frequency transform and probabilistic neural network', IET Generation, Transmission & Distribution, vol. 2, no. 2, pp. 261-270, March 2008.

[35] Tripathy, M., Maheshwari, R.P., and Verma, H.K.: 'Power Transformer Differential Protection Based on Optimal Probabilistic Neural Network', IEEE Transactions on Power Delivery, vol. 25, no. 1, pp. 102-112, January, 2010.

[36] Gerbec, D., Gasperic, S., Smon, I., Gubina, F.: 'Allocation of the Load Profiles to Consumers Using Probabilistic Neural Networks', IEEE Transactions on Power Systems, vol. 20, no. 2, pp. 548-555, May, 2005.

[37] Perera, N., Rajapakse, A., D.: 'Recognition of Fault Transients Using a Probabilistic Neural-Network Classifier', IEEE Transaction on Power Delivery, vol. 26, no. 1, pp. 410-419, January, 2011.

[38] Burrascano, P.: 'Learning vector quantization for the probabilistic neural network', IEEE Transactions on Neural Networks, vol. 2, no. 4, pp. 458-461, July, 1991.

[39] Mao, K., Z., Tan, K.-C., Ser, W.: 'Probabilistic Neural-Network Structure Determination for Pattern Classification', IEEE Transactions on Neural Networks, vol. 11, no. 4, pp. 1009-1016, July, 2000.

[40] Xin, J., Srinivasan, D., Ruey , L., C.: 'Classification of Freeway Traffic Patterns for Incident Detection Using Constructive Probabilistic Neural Networks', IEEE Transactions on Neural Networks, vol. 12, no. 5, pp. 1173-1187, September, 2001.

[41] Berthold, M., R., Diamond, J.: 'Constructive Training of Probabilistic Neural Networks', Neurocomputing, vol. 19, no. 3, pp. 167-183, March, 1998.

[42] Specht, D., F.: 'Enhancements to Probabilistic Neural Networks', In proceedings of IEEE Conf. on Neural Networks, New York, June, 1992.

[43] Specht, D., F.: 'A General Regression Neural Network', IEEE Transactions on Neural Networks, vol. 2, no. 6, pp. 568-576, November 1991.

[44] Geman, S., Bienenstock, E., Doursat, R.: 'Neural networks and the Bias/Variance Dilemma', Neural Computation, vol. 4, no. 1, pp. 1-58, January, 1992.

[45] Tomandl, D., Schober,A.:A Modified General Regression Neural Network (MGRNN) with New, Efficient Training Algorithmsas as a Robust Black Box-Tool for Data Analysis, Neural Networks, no. 14, pp. 023-1034, 2001.

[46] Silverman, B., W.:'Density Estimation for Statistics and Data Analysis', London: Chapman and Hall, 1986.

[47] MacKay, D., J., C.:'Bayesian Methods for Adaptive Models', Doctoral thesis, California Institute of Technology Pasadena, California, 1992.

[48] Minka, T., P., A Family of Algorithms for Approximate Bayesian Inference, Doctoral thesis, Massachusetts Institute of Technology, 2001.

[49] Bitner, J.R., Reingold, E.M., Backtrack programming techniques. Commun. ACM 18, pp. 651656. 1975.

[50] Prechelt, L.:'Proben 1 - A Set of Neural Network Benchmark Problems and Benchmarking Rules', Technical Report, Universitat Karlsruhe, September, 1994.

Appendix A

Probabilistic neural networks

A.1 Mean squared error decomposition

A detailed decomposition of MSE is based on the assumption that conditional mean value of every function of y is computed with respect a conditional probability density function $f(y|\mathbf{X})$

$$E[\phi(y,x)|x] = \int \phi(y,x) f(y|\mathbf{X})\mathrm{d}y, \tag{A.1}$$

then we can rewrite expression 4.43 as follows

$$\begin{aligned} E[(y-g(\mathbf{X};\mathcal{D}))^2|\mathbf{X}] &= E\big[\big(y-E[y|\mathbf{X}]\big)^2|\mathbf{X}\big] + \big(E[y|\mathbf{X}] - g(\mathbf{X};\mathcal{D})\big)^2 \\ &\quad +2E\big[\big(y-E[y|\mathbf{X}]\big)|\mathbf{X}\big]\cdot\big(E[y|\mathbf{X}]-g(\mathbf{X};\mathcal{D})\big), \end{aligned}$$

which can be decomposed in the sum of three following integrations:

a)

$$\int (y-E[y|\mathbf{X}])^2 \cdot f(y|\mathbf{X})\mathrm{d}y = E\big[(y-E[y|\mathbf{X}])^2|\mathbf{X}\big]$$

b)

$$\begin{aligned} \int \big(E[y|\mathbf{X}]-g(\mathbf{X};\mathcal{D})\big)^2 \cdot f(y|\mathbf{X})\mathrm{d}y &= \big(E[y|\mathbf{X}]-g(\mathbf{X};\mathcal{D})\big)^2 \int f(y|\mathbf{X})\mathrm{d}y \\ &= \big(E[y|\mathbf{X}]-g(\mathbf{X};\mathcal{D})\big)^2 \end{aligned}$$

since $E[y|\mathbf{X}]$ is deterministic function (see expression 4.39) as well as $g(\mathbf{X};\mathcal{D})$

c)

$$\int 2(y-E[y|\mathbf{X}])\cdot\big(E[y|\mathbf{X}]-g(\mathbf{X};\mathcal{D})\big)\cdot f(y|\mathbf{X})\mathrm{d}y = 2\big(E[y|\mathbf{X}]-g(\mathbf{X};\mathcal{D})\big) \cdot \underbrace{\int (y-E[y|\mathbf{X}])\cdot f(y|\mathbf{X})\mathrm{d}y}_{\text{equals to zero}}$$

since

$$\begin{aligned}\int (y - E[y|\mathbf{X}]) \cdot f(y|\mathbf{X})\mathrm{d}y &= \int y f(y|\mathbf{X})\mathrm{d}y - E[y|\mathbf{X}] \int f(y|\mathbf{X})\mathrm{d}y \\ &= E[y|\mathbf{X}] - E[y|\mathbf{X}] \\ &= 0\end{aligned}$$

Now it is clear that if we take all of the right sides from expressions a) - c) we will obtain equation 4.43.

A.2 Likelihood decomposition

Form of the likelihood function (see 4.45) to be marginalized lets grow a lot of various components which are not, basically, in the standard exponential form (Gaussian). The following idea makes each component possible to be expressed in the standard form. Let us to consider a function $f(x)$ in the form:

$$f(x) = exp\left[-\frac{(a-x)^2}{2\sigma^2}\right] = \underbrace{exp\left(-\frac{a^2}{2\sigma^2}\right) exp\left(\frac{2ax}{2\sigma^2}\right)}_{\text{shift and scale}} \underbrace{exp\left(-\frac{x^2}{2\sigma^2}\right)}_{\text{pure exponential}} . \tag{A.2}$$

Then, we can express an inter-product $g_2(x)$ in the same way:

$$\begin{aligned} g_2(x) &= exp\left[-\frac{(a-x)^2 + (b-x)^2}{2\sigma^2}\right] \\ &= \underbrace{exp\left(-\frac{a^2}{2\sigma^2}\right) exp\left(-\frac{b^2}{2\sigma^2}\right) exp\left(\frac{2ax}{2\sigma^2}\right) exp\left(\frac{2bx}{2\sigma^2}\right)}_{\text{shift and scale}} \underbrace{exp\left(-\frac{2\cdot x^2}{2\sigma^2}\right)}_{\text{pure exponential}} \\ &= h_2(x) \cdot exp\left(-\frac{2\cdot x^2}{2\sigma^2}\right) = \beta_2 \cdot exp\left(-\frac{2(\Phi_2 - x)^2}{2\sigma^2}\right) \end{aligned} \tag{A.3}$$

Following this idea, an arbitrary complex inter-product can be represented by the Gaussian exponential:

$$\begin{aligned} g_n(x) &= exp\left[-\frac{(a_1-x)^2 + (a_2-x)^2 + \cdots + (a_n-x)^2}{2\sigma^2}\right] \\ &= h_n(x) \cdot exp\left(-\frac{n\cdot x^2}{2\sigma^2}\right) = \beta_n \cdot exp\left(-\frac{n\cdot(\Phi_n - x)^2}{2\sigma^2}\right). \end{aligned} \tag{A.4}$$

In the following part we will proof our statements by ensuring that

$$g_2(x) = h_2(x) \cdot exp\left(-\frac{2\cdot x^2}{2\sigma^2}\right) = \beta_2 \cdot exp\left(-\frac{2(\Phi_2 - x)^2}{2\sigma^2}\right). \tag{A.5}$$

In according to expression 4.57 we can write

$$\beta_2 = exp\left[-\frac{(a-\Phi_2)^2+(b-\Phi_2)^2}{2\sigma^2}\right] \tag{A.6}$$

and

$$\Phi_2 = \frac{a+b}{2}, \tag{A.7}$$

then

$$-a^2-b^2+2ax+2bx-2x^2 = -(a-\Phi_2)^2-(b-\Phi_2)^2-2(x-\Phi_2)^2 \tag{A.8}$$

is to be proven, which is trivial.

VLASTIMIL KOUDELKA

PERMANENT ADDRESS

Klisska 81
Usti nad Labem
400 01
The Czech Republic

PERSONAL INFORMATION

Date of bird: 30.04.1985
Phone number: +420 723 403 392
Email: vlasta.koudelka@gmail.com

EDUCATION

Faculty of Electrical Engineering and Communication 2004-2007
Brno University of Technology, The Czech Republic
Course: Radio electronics, bachelor's degree
Bachelor's thesis: Neural Design of Wideband Antenna

Faculty of Electrical Engineering and Communication 2007-2009
Brno University of Technology, The Czech Republic
Course: Radio electronics, master's degree
Master's thesis: Neural Networks for EMC Modelling of Small Airplanes

EXPERIENCE

Scientific mission Deblurring techniques 2013

Short term scientific mission UPNA Antenna group, Pamplona, Spain

Junior researcher Imaging techniques, regularization 2012 - 2013

Millimeter wave EM structures for biomedical research

Junior researcher Behavioural modelling 2009 - 2011

High Intensity Radiated Field Synthetic Environment HIRF SE

Junior researcher Transfer function of aircraft fuselage 2008

Analytic Research of Threats in Electromagnetically Integrated Systems ARTEMIS

SELECTED PUBLICATIONS

1. Koudelka, V., Raida, Z.: 'Evaluation of Electromagnetic Immunity of Layered Structures by Neural Networks', IET Antennas, Microwaves & Propagation, vol. 5, pp. 482-489, 2011.
2. Koudelka, V., Raida, Z., Tobola, P.:'Simple Electromagnetic Modelling of Small Airplanes: Neural Network Approach', Radio Eng., vol. 18, pp. 38-41, 2009.
3. Koudelka, V., Svobodova, J., Raida, Z.: 'Impedance Network Simplification: A Combinatorial Optimization Approach', In Proceedings of ICEAA conference on Electromagnetics in Advanced Applications, Torino-Italy, September 2011.
4. Koudelka, V., del Rio Bocio, C., Raida, Z.: 'Diffracted Image Restoration: A Machine Learning Approach', In Proceedings of ICEAA conference on Electromagnetics in Advanced Applications, pp. 931-934, Torino-Italy, September 2013.
5. Koudelka, V., Raida, Z.: 'Evidence Based Selection Criterion for Probabilistic and General Regression Neural Networks', Neural Networks - ELSEVIER, submitted for publication.