Contents

1 Overview

The first Dagstuhl Seminar on Neural Computing had been organized by Wolfgang Maass (Technische Univrsität Graz, Austria), Christoph von der Malsburg (Ruhr Universität Bochum), Eduardo Sontag (Rutgers University, USA) und Ingo Wegener (Universität Dortmund). It brought together 35 participants from 6 countries, among these 6 from overseas.

The seminar consisted of 28 plenary talks, 5 tutorials, a panel discussion, an open problem session, and numerous discussions as well as two extra talks in smaller groups. The panel discussion covered a number of common problems in theoretical and experimental research on neural networks, such as the proper choice of models and benchmark problems for the investigation of computing and learning on biological and artificial neural networks. Tutorials were given by Christoph von der Malsburg (”The binding problem of neural networks”), Wulfram Gerstner (”Models of spiking neurons”), Angus Macintyre (”The VC-dimension of neural networks”), Jehoshua Bruck/Thomas Hofmeister/Matthias Krause (”The computational complexity of threshold circuits”), and Peter Bartlett (”Learning on neural nets from the point of view of computational learning theory”).

A thorough understanding of neural networks requires methods and results from quite diverse disciplines such as computer science, biology, engineering, physics, statistics, and mathematics. Therefore we had invited to this Dagstuhl seminar besides computer scientists also experts for neural networks from these other disciplines. The five tutorials provided a substantial common knowledge background and sufficient mutual ”language-understanding”, so that the participants could follow the presentations of new research results from all of the here represented disciplines. Many interesting interdisciplinary discussions ensued, especially since many of the participants never had the opportunity before to attend a meeting of this type.

Apart from this interdisciplinary aspects, the meeting also provided a forum for the presentation of a number of exciting new research results about neural networks. As just one example one could mention the plenary talk, and two evening sessions with technical details, which were given by Angus Macintyre from Oxford University. He reported on the solution of a well-known open problem in theoretical research on neural networks, which he had recently achieved jointly with Marek Karpinski from the University of Bonn. Their seminal result, which was made public for the first time at this Dagstuhl

meeting, provides a **polynomial** upper bound for the Vapnik-Chervonenkis dimension of sigmoidal neural nets (previously only a double-exponential upper bound had been achieved).
This new result, as well as some other new research results which were presented at this seminar, will be published in a special issue of the Journal of Computer and System Sciences for the first Dagstuhl Seminar on Neural Computing.

Wolfgang Maass

Edited by Berthold Ruf

2 Abstracts

Tutorial: The Binding Problem of Neural Networks

by *Christoph von der Malsburg*

Neural Networks aspire to be a universal data structure, fit for a great variety of applications. In their context each neuron is an elementary proposition, and a situation is fully described by a vector of neural signals. This data structure is criticised for being too limited in its power. For instance, let a visual scene contain several objects, each described by a set of neurons encoding single features. This data structure is grossly ambiguous, not being explicit about the grouping of features in terms of objects (a red triangle and a green square would be confused with a green triangle and a red square). What is called upon is a device, as fundamental ingredient of the neural data structure, for expressing the grouping (or "binding") of neurons into separate structures. This is the binding problem. It can be solved with the help of temporally structured neural signals and signal correlations to express grouping. The ambiguity mentioned is demonstrated in psychophysical experiments under viewing time restrictions. Model simulations are presented which illustrate the problem and its solution by temporal coding. The significance for reducing the complexity of learning from examples is discussed and illustrated by a model.

Tutorial: Models of spiking neurons

by *Wulfram Gerstner*

In most models of neural networks, the output of a neuron is described by an analog (or binary) variable and is often called a firing 'rate'. The firing rate is, however, a somewhat unprecise notion, and a description of biological neurons on the level of single spikes seems more appropriate.

Presently, there exist a number of different models of spiking neurons which are reviewed in this tutorial. It is shown that many of these models, in particular different versions of the integrate-and-fire model, can be classified in terms of their linear response to incoming spikes. The response is described by a kernel $\kappa(s, s')$ where s and s' denote the time that has passed since the last spike of the postsynaptic and presynaptic neuron, respectively. Furthermore, there is a function $\eta(s)$ which describes the free evolution after a spike at $s = 0$. This approach naturally leads to the Spike Response Model which is the most general renewal model with linear inputs.

From single spiking neurons to global network behavior: The 'Spike Response Model'

by *Wulfram Gerstner*

The 'Spike Response Model' is a simple but powerful phenomenological model of a single neuron. Spike generation is induced by a combination of threshold and refractoriness as described by a function $\eta(s)$ where s is the time since spike emission. Spike reception evokes a postsynaptic potential, modeled by another response function $\kappa(s, s')$ where s' or s is the time since the last presynaptic or postsynaptic spike, respectively. Within the mathematics of stochastic point processes the spiking statistics of a single neuron can be calculated for arbitrary input. The global dynamics of a fully connected network of N SRM-neurons is described by a dynamic mean-field equation which is exact in the limit of $N \to \infty$. The network can be either homogeneous or it may consist of several pools of equivalent neurons. We discuss three types of solution, i.e., incoherent, coherent, and spatio-temporal firing patterns. In the case of incoherent firing, the mean firing rate is the only relevant parameter. But there are also coherent *oscillatory* and more complex *spatio-temporal* states. The latter allow information processing on a time-scale of a few *ms*. The potential relevance of these states for coding by single spikes is discussed.

The Spike Response Model and its Applications

by *Raphael Ritz*

Feature linking and pattern separation are shown to be performed as *simultaneous* processes by a highly connected auto–associative network of *spiking* neurons (the Spike Response Model). In principle, many patterns can be separated, but with a biological set of parameters the number is limited to *four*. The patterns have been learned by an asymmetric Hebbian rule that can handle a low activity which may vary from pattern to pattern. Spikes are generated by a threshold process and – with some delay – transmitted to postsynaptic neurons. There they evoke an excitatory or inhibitory postsynaptic potential (EPSP or IPSP). Spike emission is followed by an absolute refractory period and activates an inhibitory delay loop that prevents continuous firing.

Three different network topologies are discussed, i.e., a structureless fully connected system, a hierarchical network with four subsystems that represent different 'functions' and interact via feedforward *and* feedback connections.

Functional feedback turns out to be essential for context-sensitive binding. Finally, a locally connected net is studied as a simple model of a cortical sheet. Depending on the synaptic efficacy, *four* different scenarios evolve spontaneously, viz., stripes, spirals, rings, and complex pulsating patterns. These results can be related to experimental observations of drug–induced hallucinations.

Neural Nets and Statistics

by *Kurt Hornik*

A great number of Neural Network application consist in using them, in particular multilayer perceptrons (MLPs) with sigmoidal actication functions, to "learn" unknown input-output maps f, usually by minizing MSE. From a statistician's perspective, this amounts to approximating the conditional expectation of the target y given the input x by "estimating" f in the regression model $y = f(x) + e$.

This can either be done nonparametrically by smoothing the data, or by fitting certain families $\mathcal{M}$ of parametric models, such as polynomials, radial basis functions, MLPs, etc., to the data. Hence, it is obviously of fundamental interest to carefully investigate up to what extent MLPs can outperform their competitors at this task, if at all.

Whereas the basic universality of approximation results for MLPs were obtained more than 5 years ago, the more important questions regarding *rates of approximation*, i.e. how the approximation accuracy scales with the "complexity" of both the model class $\mathcal{M}$ (typically MLPs with bounds on the number of hidden units) and the underlying f, are still far from being well-understood. We present a few important open problems in this direction. In particular, we ask what "good" or "interesting" complexity measures for continuous-valued functions are.

Polynomial Bounds for VC-Dimensions of Sigmoidal Neural Networks

by *Angus Macintyre* (joint work with Marek Karpinski)

By using methods from differential topology (Sard's Theorem and Morse Theory), and Finiteness Theorems of Hovanskii, we give bounds for the VC-dimension for neural nets with activation function $\sigma(x) = \frac{1}{1+e^{\Gamma\S}}$, in which the dominant term is w^4, where w is the number of weights. The method can be adapted to other activation functions such as $\arctan(x)$. The method also gives bounds for sparse polynomial activation functions.

Invariance in Feedforward Networks

by *John Shawe-Taylor*

The paper describes a framework for addressing the training problem of multi-layer perceptrons by a principled introduction of weight sharing. The technique not only reduces the size of the class from which the learning algorithm must select its hypothesis but also reduces the number of examples required for a given level of generalization. The required sample size is analysed in the Probably Approximately Correct (PAC) model of learning and is shown to be proportional to the number of parameters times the logarithm of the number of computational nodes. The question of assessing the functionality of the weight sharing network is addressed, with a view to ensuring that the weight constraints introduced have not excluded the target functions of the learning task. A general theorem is given characterising when a sigmoid network with a given class of symmetries can distinguish two inputs.

On Bridging the Gap between Theory and Practice of Neural Networks

by *Georg Dorffner*

In this talk I present an approach, pursued at our Institute, toward a more thorough theoretical underpinning of neural networks from the view of a practitioner. By doing this we aim at partially filling the apparent gap between common practice of neural nets and existing theoretical results. In particular, our approach is based on a general feedforward network model in the virtual room spanned by the dimensions propagation rule (”net input”), transfer function and learning rule. We show that, besides the well-known models like MLP or RBFN, this virtual room contains novel, largely unexplored network types, sometimes even with a continuum between types. Each of these network types is appropriate for certain data distributions and problem domains. Thus they can be viewed like pieces of a jigsaw puzzle yielding an overall picture of an application-driven selection of appropriate architectures for a given problem. Further discussion will be devoted to approaches for narrowing the number of degrees-of-freedom of networks in the light of some rather pessimistic results on learnability from computational learning theory. This view will be highlighted with several examples - mainly from the medical domain - including cases of initialization based on data analysis, adaptive decision boundaries and neural network units functioning as hyperplane regressors.

Tutorial: The Computational Complexity of Threshold Circuits
by *Thomas Hofmeister* and *Matthias Krause*

Threshold circuits may be viewed as discrete feedforward neural nets. In this tutorial, we try to give a survey of recent results which characterize the computational power of threshold circuits.

There is a close connection to analog neurons and we mention how sigmoidal circuits can be simulated by threshold circuits efficiently.

We then discuss the influence of the size of the weights upon the power of threshold circuits. E.g., a recent result showed that exponential-size weights can be simulated by polynomial-size weights by increasing the depth only by one.

Some of the complexity classes which capture the power of threshold circuits are defined, namely LT_d and $\hat{LT}_d = TC_d^0$. In particular, TC_d^0 stands for the class of all functions which can be computed in threshold circuits of polynomial size and depth d. Many basic functions are now known to be contained in TC_d^0 for some small d. In fact, in many cases, even depth 2 or depth 3 is enough. Some of the functions are listed in the talk.

Another result included is the fact that ACC-functions have subexponential size depth 3 threshold circuits. On the lower bound side, we mention the function "Inner Product modulo 2" which is known not to be computable in TC_2^0.

A short section of the talk is devoted to depth 2 circuits with a threshold output gate and gates from some set of basic operations as its input gates. One example is the set of all parity functions, which leads us to the notion of Voting Polynomials.

We then go into detail as far as the upper bounds are concerned. One very successful technique which helped in designing small-depth threshold circuits is given by the notion of approximability. We will define it and sketch its usefulness by explaining how it can be used to reduce the depth of threshold circuits.

Since circuits for more complex functions are designed using decomposition into less complex functions, it is useful to know that some of these "less complex" functions are 1-approximable. We show that some of the earlier methods which proved some function to be contained in TC_2^0 also yield as

a byproduct the proof that the function is even 1-approximable. Examples of 1-approximable functions which are used as submodules are the "multiple addition" or every symmetric function. We then show another depth-saving trick which consists of the observation that constant fan-in can be exploited when input functions are d-approximable.
These tricks can be applied to show that basic functions like "division", "multiplication" or "sorting" can be computed in TC_3^0. We also show some lower bound results which use projection reductions and which show that the circuits constructed above are depth-optimal if we require polynomial size.

Fourier Transforms and Threshold (Neural) Circuit Complexity

by *Jehoshua Bruck*

There exists a large gap between the empirical evidence of the computational capabilities of neural networks and our ability to systematically analyze and design those networks. While it is well known that classical Fourier Analysis is a very effective mathematical tool for the design and the analysis of linear systems, such a tool was not available for artificial neural networks which are inherently nonlinear. Recently, the spectral analysis tool was introduced in the domain of discrete neural networks. The application of the spectral technique led to a number of new insights and results, including lower and upper bounds on the complexity of computing with neural networks as well as methods for constructing optimal (in terms of performance) feedforward networks for computing various arithmetic functions. The focus of the presentation is on an elementary description of the basic techniques of Fourier analysis and its applications in threshold circuit complexity.

Realizing AC_0–functions by Real Polynomials

by *Matthias Krause*

The task of realizing an n–argument Boolean function f by one threshold gate is equivalent to constructing a hyperplane (i.e., a linear functional $l = l(x_1, \ldots, x_n)$) of the n–dimensional Euclidian space which separates the positive inputs from the negative inputs. We study the more general problem of representing f by a polynomial $p = p(x_1, \ldots, x_n)$ in the sense that for all positive inputs x it holds $p(x) > 0$ and for all negative inputs x it holds $p(x) < 0$. We are interested in estimating the *length* of f which is defined to be the minimal number of monomials a polynomial must have for representing a given function in the above sense. Observe that $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$

can be represented by k monomials iff f has a depth–two circuit consisting of k parity gates at the bottom connected with one threshold gate at the top. There was known a close relation between the representability of a given function and properties of its spectral coefficients, the length of f is bounded from above by the sum of the absolute values all spectral coefficients, and from below by the inverse of the maximal coefficient (*Bruck, Smolensky 1990*). We study representability of AC_0–functions. We show by probabilistic arguments that on the one hand $AC_{0,2}$–functions always have small length, but that one the other hand there are $AC_{0,3}$–functions of exponential length. Both results can not be obtained by spectral–theoretic arguments as by a result of *Linial, Mansour,* and *Nisan* (1990) AC_0–functions always have big spectral coeffieceints, where also the sum of the absolute values all spectral coefficients may be exponential.

Computing Sparse Approximations Deterministically

by *Thomas Hofmeister* and *Hanno Lefmann*

It is known that for every $n \times m$-matrix A with entries taken from the interval $[0,1]$ and for every probability vector $\underline{p}$, there is a sparse probability vector $\underline{q}$ with only $O(\ln n/\varepsilon^2)$ non-zero entries such that every component of the vector $A \cdot \underline{q}$ differs from every component of $A \cdot \underline{p}$ in absolute value by at most ε.

The existence of such a vector is proved by a probabilistic argument. It was an open problem whether there is an efficient, i.e. polynomial-time, deterministic algorithm which actually constructs such a vector $\underline{q}$.

In this paper, we provide an algorithm which computes such a vector $\underline{q}$ and which takes time polynomial in n,m, and ε. The algorithm is based on the method of "pessimistic estimators".

The approximation result was crucial in some applications to matrix games. In the talk, we also sketch why this result could have some implications for threshold circuits.

On the complexity of analog circuits: computing Boolean functions with analog circuits of bounded fan–in

by *György Turán* (joint work with Farrokh Vatan)

An arithmetic threshold circuit is built of bounded fan–in addition, subtraction, multiplication and sign gates, and the real constant inputs. It is related to the other models of analog computation such as feedforward neural nets and real random access machines. We consider the computational power of analog threshold circuits for computing Boolean functions. It was shown by Gashkov that almost all n–variable Boolean functions require arithmetic circuits (i.e. arithmetic threshold circuits without sign gates) of size $\Omega(2^{n/2})$. We show that this bound can be extended to arithmetic threshold circuits. On the other hand, there is a size–depth trade–off in the sense that for every polynomial $p(n)$, for almost all Boolean functions f, every arithmetic circuit of depth $p(n)$ computing f has size $\Omega(2^{n-O(\log n)})$. We also prove a superlinear lower bound for the arithmetic threshold formula size of an explicitly defined Boolean finction. It is shown that the arithmetic threshold formula size of the Element Distinctness function is $\Omega(n^{3/2}/\log n)$. This implies a lower bound for 'standard' threshold circuits: every depth d threshold circuit with arbitrary weights computing this function has size $\Omega(n^{1/2(d-1)}/(\log n)^{1/d-1})$.

Computational Models Using Recurrent Neural Nets

by *Eduardo Sontag* (joint work with Hava Siegelmann)

We pursue a particular approach to analog computation, based on dynamical systems of the type used in neural networks research. Our systems have a fixed structure, invariant in time, corresponding to an unchanging number of "neurons". If allowed exponential time for computation, they turn out to have unbounded power. However, under polynomial-time constraints there are limits on their capabilities, though being more powerful than Turing Machines. (For rational weights, a similar but more restricted model is shown to be polynomial-time equivalent to classical digital computation.) Moreover, there is a precise correspondence between nets and standard non-uniform circuits with equivalent resources, and as a consequence one has lower bound constraints on what they can compute. This relationship is perhaps surprising since our analog devices do not change in any manner with input size. We note that these networks are not likely to solve polynomially NP-hard problems, as the equality "P = NP" in our model implies the almost complete collapse of the standard polynomial hierarchy. In contrast to classical computational models, the models studied here exhibit at least some robustness with respect to noise and implementation errors.

On the Computational Complexity of Networks of Spiking Neurons

by *Wolfgang Maass*

We investigate the computational power of a formal model for networks of spiking neurons. It is shown that simple operations on phase-differences between spike-trains provide a very powerful computational tool that can in principle be used to carry out highly complex computations on a small network of spiking neurons. We construct networks of spiking neurons that simulate arbitrary threshold circuits, Turing machines, and a certain type of random access machines with real valued inputs. We also show that relatively weak basic assumptions about the response- and threshold-functions of the spiking neurons are sufficient in order to employ them for such computations. Furthermore we prove upper bounds for the computational power of networks of spiking neurons with arbitrary piecewise linear response- and threshold-functions, and show that they are with regard to real-time simulations computationally equivalent to a certain type of random access machine, and to recurrent analog neural nets with piecewise linear activation functions. In addition we give corresponding results for networks of spiking neurons with a limited timing precision, and we prove upper and lower bounds for the VC-dimension and pseudo-dimension of networks of spiking neurons.

Subsymbolic-Symbolic Cooperative Learning

by *Giancarlo Mauri*

Today, there is a strong evidence that a lot of advantages can be obtained by integrating a symbolic approach to learning with a subsymbolic (i.e., neural) one, so as they can cooperate in a more powerful paradigm. Using this kind of integration, we obtained very good results in control problems (control of a flexible arm, control of the attitude angles of a geostationary satellite), in face recognition and in natural language parsing, better than with classical symbolic approaches. After showing these examples, a formal model of learning machine is proposed where the usual components of the so called Probably Approximately Correct (PAC) learning model interact with a neural network which behaves as a noisy, but realistic, Oracle. The ideal framework is that a broad "intuitive" knowledge about a concept c, achieved subsymbolically through the neural network, is employed to enlarge, randomly or by demand, the set of examples available for giving a symbolic representation of c within a concept class C. In case a neural network is trainable with no errors on the training set and no ”malicious” errors in generalization, we exhibit the success of two PAC-style algorithms for learning the two classes k-CNF and k-term DNF.

Data Clustering and Computer Vision: An Approach to Adaptive Vision

by *J.M. Buhmann*

Partitioning a set of data points which are characterized by their mutual dissimilarities instead of an explicit coordinate representation is a difficult, NP-hard combinatorial optimization problem. We formulate this optimization problem of a pairwise clustering cost function in the maximum entropy framework using a variational principle to derive corresponding data partitionings in a d-dimensional Euclidian space. This approximation solves the embedding problem and the grouping of these data into clusters simultaneously and in a selfconsistent fashion.

The algorithm implements a new strategy for nonlinear dimension reduction and visualization. To yield a clustering solution of predefined quality, active data selection is employed to considerably reduce the number of required dissimilarities.

Neural Computation for Robot Vision

by *Helge Ritter*

The talk focuses on a class of neural learning algorithms that are derived from the Self-organizing maps, which model the structuring of neural layers in the brain. The talk presents two extensions of the basic self-organizing map approach, namely (i) LLM-networks, in which a self-organizing map is used to adaptively tesselate the input space for a collection of locally linear maps or “linear experts”, and (ii) parametrized self-organizing maps (PSOMs), in which the map is represented parametrically, using a set of basis manifolds or “basis maps”. In contrast to most spin-glass-type networks, which are limited to the storage of point attractors, PSOMs can be viewed as recurrent nets with a more general, smooth attractor manifold and thus can provide a “continuous associative memory”. Several case studies of these approaches to learning problems in robotics and computer vision are discussed. In addition, a hierarchical combination of multiple PSOMs is presented, in which higher level PSOMs use pre-trained lower-level PSOMs as their "building blocks". This approach allows a hierarchical specialization and can lead to extremely rapid learning at the upper levels.

Segmentation of Optical Flow Fields

by *Hans-Helmut Nagel*

While estimating both components of optical flow based on the postulated validity of the Optical Flow Constraint Equation, it has been tacitly assumed so far that the partial derivatives of the gray value distribution - which are required for this approach at the pixel positions involved - are independent from each other. It has been shown in a theoretical investigation (Nagel 94) how dropping this assumption affects the estimation procedure. The advantage of such a more rigorous approach consists in the possibility to replace heuristic tests for the local detection of discontinuities on optical flow fields by well known stochastic tests. First results from various experiments with this new approach are presented and discussed. The question has then been raised of how these results might be related to neural computation.

Tutorial: Computational Learning Theory

by *Peter Bartlett*

This tutorial provides an introduction to computational learning theory and reviews results relevant to the problem of learning in neural networks.

We review Valiant's probably approximately correct model, which gives a probabilistic framework for learning classification functions. In this model, the sample size necessary to learn using a set of classification functions (such as a class of feed-forward neural networks) depends on a combinatorial measure of complexity of the function class known as its Vapnik-Chervonenkis dimension. The computational complexity of learning is equivalent to that of a certain optimization problem (finding a function in the class consistent with the data). We discuss the implications of these results for neural network learning, concentrating on feed-forward networks. We describe extensions of these results to the problem of learning real-valued functions.

Agnostic PAC-Learning Within Small Neural Nets

by *Hans Ulrich Simon*

We consider a variant of Wolfgang Maass' algorithm for agnostic pac-learning within small neural networks. Our learning algorithm runs on an architecture consisting of the input layer, one hidden layer, and the output layer. This architecture is augmented with an auxiliary data structure which represents an ordered partition P of the input space into p cells. The units in the hidden layer compute functions which are linear in the input variables, and whose weights are programmable. The output units compute piecewise linear functions, where there are p pieces (linear functions) per output unit whose weights are architectural (not programmable). In computation mode, the ordered partition controls which of these pieces is applied: if the input belongs to cell numbered i, then all output units use the respective piece number i. In learning mode, not only the programmble weights associated with the hidden units are programmable, but also the choice of the ordered partition P is adjustable: P may be selected from a given "polynomialy enumerable" system $\mathcal{P}$ of partitions. "Polynomially enumerable" means that given a sample S of m training examples, there are only $M = pol(m)$ many different ways to partition S by partitions from $\mathcal{P}$, and representations for such partitions $P_1, \ldots, P_M$ can be computed in polynomial time.

Our main results are as follows:

1. Agnostic pac-learning of real-valued functions can be perfomed on each architecture of this kind by solving $pol(\epsilon, \delta)$ many linear programs.

2. Our architecture can efficiently simulate any first order net with piecewise linear activation functions. Thus our learning algorithm can take first-order architectures of constant size as its touchstone class.

3. Although our architecture is computationally more powerful than the architecture (the first-order neural tree) used by Maass' algorithm, the linear programs that we must solve are considerably simpler, and the generalization capabilities of both algorithms are similar (because the pseudo-dimensions of the associated classes of loss-functions are closely related).

On the Complexity of Learning on Perceptrons with Binary Weights

by *Michael Schmitt*

We know that a lot of problems dealing with learning in neural networks are computationally not feasible. Commonly used methods to cope with complexity in practice try to incorporate knowledge about the functions being learned into the algorithm. Techniques to choose the architecture or predetermine weights may lead to more efficient algorithms, however, we do not know how to apply these methods optimally.

In our work we pursue a different approach. We try to bypass intractability of training a specific architecture by restricting the set of permitted training samples. To this end, we introduce two parameters to characterize what we think makes samples hard to train. The values of the first parameter, termed "heaviness", is the maximum dot product of an example with itself, the second parameter, termed "coincidence", is the maximum dot product of two different examples. For binary examples heaviness corresponds to the maximum number of non-zero components, coincidence corresponds to the maximum number of non-zero components that two examples have in common.

The results presented concern single neurons with binary weights and binary inputs. It is shown that the problem of achieving agreement with all examples remains NP-complete if the examples are allowed to have heaviness at least 4 and coincidence at least 1. For the problem of minimizing disagreement we obtain NP-completeness already for heaviness 2 and coincidence 1. For all complementary cases we are able to present linear time algorithms. Thus we have completely characterized all defined restrictions of learning problems for this architecture with respect to their complexity. Similar results have been obtained for other architectures as well.

Convergence of the back-propagation algorithm for time-delay and recurrent networks

by *Peter Bartlett*

We study local convergence properties of a gradient descent learning algorithm for two-layer time-delay and locally recurrent sigmoid networks. We assume that the observed data sequence is generated by a network of this type with a known structure, and consider the convergence of the estimated parameters to their true values. Under mild conditions on the true parameters (that are generically satisfied), for almost all input sequences the estimated parameters locally converge to these true values exponentially fast. Furthermore for periodic input sequences, almost all sequences with period at least N will suffice, where N is the number of parameters. Any shorter sequence will not give guaranteed local exponential convergence, in the sense that there are initial parameter estimates in every neighbourhood of the true parameters that do not lead to convergence to the true parameters.

A Rigorous Analysis Of Linsker-type Hebbian Learning Networks

by *Vwani P. Roychowdhury*

We propose a novel approach for a rigorous analysis of the nonlinear dynamics of Linsker's unsupervised Hebbian learning network. Our analysis allows us to determine the whole set of fixed point attractors of the nonlinear synaptic stabilization process, and explicitly obtain a necessary and sufficient condition for the emergence of structured receptive fields. These results provide for the first time comprehensive explanations of the generation of the various structured connection patterns, and of the roles of the different system parameters of the model. In particular, the crucial role of the synaptic density function is explicitly demonstrated. The parameter regimes for the emergence of commonly observed receptive fields (e.g., center-surround, oriented, and bi-lobbed cells) are explicitly derived from our framework. The theoretical results derived in our work provide (without any approximation) rigorous analytical justification of several key observations made about the dynamics of the Linsker's network. Our theoretical predictions are also confirmed by numerical simulations.

Note: This is a joint work with Hong Pan and Jianfeng Feng, and was supported in part by the NSF Grant No. ECS-9308814

Approximation and learning of real-valued functions

by *Pascal Koiran* (joint work with Leonid Gurvits)

We present a fairly general method for constructing classes of functions of finite scale-sensitive dimension (the scale-sensitive dimension is a generalization of the Vapnik-Chervonenkis dimension to real-valued functions). The construction is as follows: start from a class F of functions of finite VC dimension, take the convex hull $\mathrm{co}F$ of F, and then take the closure $\overline{\mathrm{co}F}$ of $\mathrm{co}F$ in an appropriate sense. As an example, we study in more detail the case where F is the class of threshold functions. It is shown $\overline{\mathrm{co}F}$ includes two important classes of functions:

- neural networks with one hidden layer and bounded output weights;
- the so-called Γ class of Barron, which was shown to satisfy a number of interesting approximation and closure properties.

We also give an integral representation in the form of a "continuous neural network" which generalizes Barron's. It is shown that the existence of an integral representation is equivalent to both L^2 and L^∞ approximability.

Object Recognition by Elastic Graph Matching

by *Christoph von der Malsburg*

Retinal images of objects vary trivially due to changing perspective. Effective learning from examples is possible only on the basis of an invariant representation. I desribe a neural system that represents objects and images as two-dimensional labeled graphs. Approximate isomorphy (in the sense of similar label arrangement) between a model and a segment of the image is discovered by elastic matching. As labels we employ sets of wavelets. The match process takes place in two stages, a "global move" and a "local move." In the global move a model graph is shifted, rotated and scaled without distortion to find a global optimum of the sum of pairwise label similarities between nodes in the model and the image. In the ensuing local move, individual model nodes are allowed to diffuse over the image plane, each trying to maximize label similarity while minimizing graph distortion. Matches of different objects are compared in terms of a global cost function characterizing the match. A fully neural version of the match process is presented by Laurenz Wiskott in his talk.

Object Recognition with Dynamic Link Matching

by *Laurenz Wiskott* and *Christoph von der Malsburg*

A fully neural system for translation and distortion invariant object recognition is presented. As an example we use faces. A couple of faces are stored as small layers of neurons carrying Gabor jets as features. Gabor jets are local descriptors of the underlying grey value distribution of the respective face images. A new face to be recognized is represented on a larger layer. It can be at any position and distorted by rotation in depth or a different mimic. The key problem in this kind of task is to find the right mapping between the input image layer and the model layers of stored faces. Here we apply the Dynamic Link Matching. Its principle mechanism is the following: The initial connectivity between the model layer and the image layer is given by the similarity of the respective features. Neighbouring neurons of one layer and connected nodes of the two layers do cooperate and tend to have correlated time signals. By this means a regular, neighbourhood preserving mapping between the two layers depending on the feature similarities is induced by correlation. Based on the correlations the synaptic links can switch on a fast time scale and finally converge to the correct mapping between image and model domain. Once the correct mapping was found the recognition task just consists in selecting the model layer with the strongest connections to the image, given that the synaptic weights are bound by the similarities between the local features.

Learning with incremental self-organizing networks

by *Bernd Fritzke*

A class of self-organizing networks is presented which can be applied to unsupervised and supervised learning. In both cases the network structure is incrementally constructed by a growth process. Insertion of new units are done on the basis of statistics which are gathered locally at the existing units. At the same time a neighborhood connectivity is constructed.

In the case of unsupervised learning there are (at least) two important problem classes: a) In *topology learning* one likes to describe the topology of a signal distribution by a graph consisting of nodes in n-dimensional space and connecting edges. In other words, the submanifold where the signal density is non-zero is to be identified. b) In *dimensionality reduction* one likes to map the possibly high-dimensional data onto a lower-dimensional sheet of neurons. One application of this procedure is data visualization.

For both unsupervised learning problems incremental algorithms are presented which differed mainly in the way the topology was updated. The Growing Neural Gas Method used a competitive Hebbian learning to generate a topology which closely reflects the topology of the data submanifold. Thereby, the network dimensionality may locally vary with the dimensionality of the data submanifold. The Growing Cell Structures, in contrast, generate a topology which has a fixed dimensionality, no matter what the dimensionality of the given data may be. The result is a dimensionality-reducing mapping which tries to preserve neighborhood relations.

For supervised learning the described incremental models can be coupled with the radial basis function (RBF) approach. This leads to incremental RBF networks. In this case accumulated classification error is used to guide the insertion of new units (problem dependent positioning). The result are small networks which generalize well and can be constructed in with a fraction of the training epochs needed, e.g., for a back-propagation trained multi-layer

perceptron.
General advantages of all proposed models over other approaches are that all parameters are constant (no variation over time) and that network size and structure need not to be predefined but result from the growth process and some user-definable stopping criterion.

Local Minima of Least Square Problems Associated to Sigmoidal Nets

by *Eduardo Sontag*

In this talk, I described techniques that allow estimating the number of local minima in least square problems associated to sigmoidal nets. The techniques combine nonlinear approximation facts and Khovanskii-type estimates (to count critical values), tools from semianalytic set theory (to show that the estimates are good for almost all input/output data, leading to good teaching dimension), and finally the uniqueness results developed in joint work with Francesca Albertini (for showing that weights are determined by the values).

Discovering neural nets with low Kolmogorov complexity and high generalization capability

by *Jürgen Schmidhuber*

Many machine learning algorithms aim at finding "simple" rules to explain training data. The expectation is: the "simpler" the rules, the better the generalization on test data (→ Occam's razor). Most practical implementations, however, use measures for "simplicity" that lack the power, universality and elegance of those based on Kolmogorov complexity and Solomonoff's algorithmic probability. Likewise, most previous approaches (especially those of the "Bayesian" kind) suffer from the problem of choosing appropriate priors. This paper addresses both issues. It first reviews some basic concepts of algorithmic complexity theory relevant to machine learning, and how the Solomonoff-Levin distribution (or universal prior) deals with the prior problem. The universal prior leads to a probabilistic method for finding "algorithmically simple" problem solutions with high generalization capability. The method is based on Levin complexity (a time-bounded generalization of Kolmogorov complexity) and inspired by Levin's optimal universal search algorithm. With a given problem, solution candidates are computed by efficient "self-sizing" programs that influence their own runtime and storage size. The probabilistic search algorithm finds the "good" programs (the ones quickly computing algorith- mically probable solutions fitting the training data). Simulations focus on the task of discovering "algorithmically simple" neural networks with low Kolmogorov complexity and high generalization capability. It is demonstrated that the method, at least with certain toy problems where it is computationally feasible, can lead to generalization results unmatchable by previous neural net algorithms. Much remains to be done, however, to make large scale applications and "incremental" learning feasible.

Associative Memory Capacities

by *Günther Palm*

We give an overview of recent results concerning the information storage and retrieval capacity of neural associative memories. Another emphasis of the talk is on the fine differences in various definitions of these capacities that are used in the literature.

The tasks are auto-association and hetero-association; one has to store and retrieve a set S of patterns $x^\mu (\mu = 1, \ldots, n)$, or a mapping, $x^\mu \to y^\mu$, i.e. a set S of pairs (x^μ, y^μ), respectively. The patterns x^μ and y^μ are binary patterns of length n .

Storage is performed in the sum-of-outer-products matrix C (or in its binary version). In auto- association, retrieval is performed by pattern completion of the stored patterns or by identifying (recognizing) the stored patterns upon presentation of the whole patterns. In hetero-association, retrieval is performed by mapping the inputs x^μ to the outputs y^μ . This is done by means of a simple network of binary threshold neurons with connectivity or weight matrix C. In every case capacities are calculated as the gain in transinformation about the stored patterns in the retrieval process.

In this way one can define and distinguish the following capacities:

- Recognition capacity c_R
- Completion capacity c_C
- Mapping capacity c_M
- Storage capacity c_S .

These obey the obvious relations $c_S \geq c_{M+R} \geq c_M, c_R, c_{M+C} \geq c_C$.

One can show that these capacities can be maximized for **sparse** pattern vectors , i.e. for vectors x^μ, y^μ , where most components are 0. In this so called sparse limit the following results can be calculated by statistical methods (compare Palm, Concepts in Neuroscience 2 (1991), 97-128):
(1) For hetero-association with additive learning rule

$$\frac{1}{2\ln 2} = c_S = c_{M+R} = c_{M+C} = c_M = c_R > c_C$$

(2) For hetero-association with binary weights

$$\ln 2 = c_S = c_{M+R} = c_{M+C} = c_M = c_R > c_C$$

(3) For auto-association with additive learning rules

$$\frac{1}{4\ln 2} = c_S = c_R > c_C$$

(4) For auto-association with binary weights:

$$\frac{ln2}{2} = c_S = c_R > c_C$$

Computational Complexity Issues in Recurrent Nets

by *Pekka Orponen*

We consider the computational power and other computational complexity aspects of discrete recurrent network models. With the help of known constructions of symmetric networks with exponential transient times we show that polynomially growing sequences of symmetric (i.e., Hopfield) networks have the full computational power of polynomial space bounded Turing machines. Restricting the networks to have polynomially bounded weights constrains their computational power to that of polynomial time bounded Turing machines. Based on this equivalence, a toy compiler translating simple parallel condition-action programs into Hopfield networks has been designed and implemented.

We also point to some known results and open questions concerning the complexity of analyzing and synthesizing associative memory networks for a given set of binary patterns.

On Ensembles of Competing Experts

by *Thomas Martinetz*

Ensembles of individual experts or agents allow solutions of action control tasks by creating subtasks, each of which is handled by the most suited expert of the ensemble. This approach is very promising, however, it turns out that adapting such an ensemble of competing experts to the global state which yields the best solution for a given task is very difficult. The reason is that the best expert for a subtask strongly tends to increase its task domain, which leads to a suppression of other experts and prevents an optimal exploitation of the ensemble's resources. Up to now, only heuristic solutions to the problem are known.

We present a solution based on the maximum entropy principle. Under given constraints, e.g., a desired average output error of the ensemble of experts, the maximum entropy principle provides the probability that a given state of the ensemble yields the best solution. This probability can be used to select an ensemble state which optimizes certain criteria, e.g., provides the best solution on average (which naturally leads to mixtures of experts) or, alternatively, provides the best solution most likely. The maximum entropy principle leads to a "computational temperature" of the ensemble and sugests deterministic annealing for steering a self-organizing adaptation process. With this adaptation process experts are formed by phase transitions.

3 List of Participants

Francesca Albertini Universitá di Padova, Dipartimento Mathematica, Via Belzoni 7, I-35100 Padova, Italy
albertini@russel.unipd.it, tel.: +39-49-831-966

Peter Auer TU Graz, Institut für Grundlagen der Informationsverarbeitung, Klosterwiesgasse 32, A-8010 Graz , Austria
pauer@igi.tu-graz.ac.at

Peter Bartlett Australian National University, Department of Systems Engineering, 0200 Canberra, Australia
peter.bartlett@anu.edu.au, tel.: +61-6-249-5198

Jehoshua Bruck Caltech- Pasadena, Electrical Engineering Dept., 116-81, Pasadena CA 91125, USA
bruck@systems.caltech.edu

J.M. Buhmann Universität Bonn, Institut für Informatik II, Römerstr. 164, D-53117 Bonn, Germany
jb@informatik.uni-bonn.de tel.: +49-228-550-380

Georg Dorffner Österreichisches Forschungsinsitut für AI, Schottengasse 3, A-1010 Wien , Austria
georg@ai.univie.ac.at, tel.: +43-1-532-32 81-0

Bernd Fritzke Universität Bochum, Institut für Neuroinformatik, D-44780 Bochum, Germany
fritzke@neuroinformatik.ruhr-uni.bochum.de, tel.: +49-234-700-7921

Wulfram Gerstner TU München, Institut für Theoretische Physik, D-85747 Garching, Germany
wgerst@physik.tu-muenchen.de, tel.: +49-89-3209-2193

Thomas Hofmeister Universität Dortmund, Fachbereich Informatik, D-44221 Dortmund, Germany
hofmeist@zorro.informatik.uni-dortmund.de, tel.: +49-231-755-48 08

Kurt Hornik TU Wien, Institut für Statistik und Wahrscheinlichkeitstheorie, Wiedner Hauptstraše 8-10, A-1040 Wien, Austria
kurt.hornik@neuro.tuwien.ac.at, tel.: +43-1-58801-45 42

Günter Hotz Universität des Saarlandes, Fachbereich 14 - Informatik, Postfach 15 11 50, D-66041 Saarbrücken, Germany
hotz@cs.uni-sb.de, tel.: +49-681-302-2414

Pascal Koiran Rutgers Univ. - Piscataway, DIMACSCenter, P.O.Box 1179, Piscataway NJ 08855-1179, USA
koiran@dimacs.rutgers.edu

Matthias Krause Universität Dortmund, Fachbereich Informatik II, D-44221 Dortmund, Germany
krause@daedalus.informatik.uni-dortmund.de, tel.: +49-231-755-4702

Wolfgang Maass TU Graz, Institut für Grundlagen der Informationsverarbeitung, Klosterwiesgasse 32/2, A-8010 Graz, Austria
maass@igi.tu-graz.ac.at, tel.: +43-316-8100-63 / 22

Angus Macintyre Oxford University, Mathematical Institute, 24-29 St. Giles, Oxford OX1 3LB, Great Britain
ajm@vax.ox.ac.uk / angus@gnumath.rutgers.edu, tel.: +44-865-273535

Christoph von der Malsburg Ruhr-Universität Bochum, Institut für Neuroinformatik, D-44780 Bochum, Germany
malsburg@neuroinformatik.ruhr-uni-bochum.de, tel.: +49-234-700-7997

Thomas Martinetz Siemens AG, ZFE ST SN 41, Corporate R & D, Otto-Hahn-Ring 6, 81739 München, Germany
tm@inf21.zfe.siemens.de, tel.: +49-89-636-49532

Giancarlo Mauri Universitá degli Studi di Milano, Dip. Scienze dell' Informazione, via Comelico 39-41, I-20135 Milano, Italy
mauri@hermes.unimi.it, tel.: +39-2-55 00 62 29

Hans-Helmut Nagel Fraunhofer Inst. für Informations-& Datenverarbeitung Haid-und-Neu-Straše 10-14, D-75131 Karlsruhe, Germany
hhn@iitb.fhg.de, tel.: +49-721-6091-210

Pekka Orponen, University of Helsinki, Dept. of Computer Science, Teollisuuskatu 23, FIN-00014 Helsinki, Finland
orponen@cs.helsinki.fi, tel.: +358-0-708-4224

Günther Palm Universität Ulm, Abteilung Neuroinformatik, D-89069 Ulm, Germany
palm@neuro.informatik.uni-ulm.de, tel.: +49-731-502-4151

Rüdiger Reischuk Med. Universität zu Lübeck, Naturwissenschaftliche Fakultät, Institut für Theoretische Informatik, Wallstraše 40, D-23560 Lübeck, Germany
reischuk@informatik.mu-luebeck.de, tel.: +49-451-7030-416

Helge Ritter Universität Bielefeld, Universitätsstr. 25, D-33501 Bielefeld, Germany
helge@techfak.uni-bielefeld.de, tel.: +49-521-106-60 62

Raphael Ritz TU München, Institut für Theoretische Physik, D-85747 Garching, Germany
ritz@physik.tu-muenchen.de, tel.: +49-89-3209-2193

Vwani Roychowdhury Purdue University, School of Electrical Engineering, West Lafayette IN 47907, USA
vwani@ecn.purdue.edu, tel.: +1-317-494-0636

Berthold Ruf TU Graz, Institut für Grundlagen der Informationsverarbeitung, Klosterwiesgasse 32, A-8010 Graz, Austria
bruf@igi.tu-graz.ac.at, tel.: +316-8100-6324

Jürgen Schmidhuber TU München, Fakultät für Informatik, D-80290 München, Germany
schmidhu@tumult.informatik.tu-muenchen.de, tel.: +49-89-2105-2406

Michael Schmitt TU Graz, Institut für Grundlagen der Informationsverarbeitung, Klosterwiesgasse 32, A-8010 Graz, Austria
mschmitt@igi.tu-graz.ac.at, tel.: +43-316-8100-6314

John Shawe-Taylor Royal Holloway and Bedford New College, Department of Computer Science, Egham Hill, Egham TW20 0EX, Great Britain
john@dcs.rhbnc.ac.uk, tel.: +44-784-443-430

Hans Ulrich Simon Universität Dortmund, Fachbereich Informatik, Lehrstuhl II, D-44221 Dortmund, Germany
simon@152.informatik.uni-dortmund.de, tel.: +49-231-755- 5132

Eduardo D. Sontag Rutgers University-New Brunswick, Dept. of Mathematics, New Brunswick NJ 08903, USA
sontag@control.rutgers.edu, tel.: +1-908-932- 3072

György Turan The University of Illinois at Chicago, Dept. Math. Stat. & Comp. Sci., M/C 249, 851 S. Morgan, Chicago IL 60607-7045, USA
u11557@uicvm.bitnet

Ingo Wegener Universität Dortmund, Fachbereich Informatik II, D-44221 Dortmund, Germany
wegener@ls2.informatik.uni-dortmund.de, tel.: +49-231-755-2776

Laurenz Wiskott Ruhr-Universität Bochum, Institut für Neuroinformatik, D-

44801 Bochum, Germany
laurenz@neuroinformatik.ruhr-uni-bochum.de, tel.: +49-234-700-7921