Department of Computer Methods,
Nicholas Copernicus University,

Grudziadzka 5, 87-100 Toruñ, Poland

Floating Gaussian Mapping: a New Model of Adaptive Systems

Wlodzislaw Duch[1]

Adaptive systems are usually realized by the neural network type of algorithms. In this contribution an alternative approach, based on products of Gaussian factors centered at the data points, and acting as feature detectors, is proposed. Comparing with the feedforward neural networks with backpropagation learning is much faster because explicit construction of the approximation to the desired mapping is performed, with fine tuning via subsequent adaptation of the shapes and positions of the feature detectors. Comparing with the recurrent feedback networks this approach allows for full control of the positions and sizes of the basins of attractors of the stationary points. Retrieving information is factorized into a series of one-dimensional searches. The FGM (Floating Gaussian Mapping) model is applicable to learning not only from examples but also from general laws. It may serve as a model of associative memory or as a fuzzy expert system. Examples of application include identification of spectra and intelligent databases (associative memory type), analysis of simple electrical circuits (expert system type), and classification problems (two-spirals problem).

UMK-KMK-TR-5/93
Neural Network World **4** (1994) 645-654

1. Introduction

In this paper a model of an adaptive system capable of arbitrary associations, based on multidimensional Gaussian functions, is presented. This model employs a new way of knowledge representation, storing complex and fuzzy facts and allowing not only for simple associations but also for drawing logical inferences. Since these tasks are most frequently done at present using models of adaptive systems such as Artificial Neural Networks (ANNs), Learning Vector Quantization (LVQ) and self-organizing mappings [1], models that became very fashionable in recent years, I will describe it using the language of neural algorithms.

Adaptive system $A_{\mathbf{W}}$ is a system with internal adjustable parameters $\mathbf{W}$ performing vector mappings from the space of inputs $\mathbf{X}$ to the space of outputs $\mathbf{Y} = A_{\mathbf{W}}(\mathbf{X})$. Neural networks are the most popular but not the only adaptive systems known. Some of the most successful adaptive systems (like LVQ [2]) are not, strictly speaking, of the neural network type, although they are usually classified as such. The design of ANNs was motivated by the parallel processing capabilities of the real brains, but the processing elements and the architectures used in artificial neural networks frequently have nothing to do with biological structures. Artificial neural networks are networks of simple processing elements (usually called "neurons") operating on their local data and communicating with other elements. Thanks to this global communication ANN reaches a stable state consistent with the current input and output values. The strength (weights) of connections between neural elements are the

[1] duch@phys.torun.edu.pl

adjustable parameters **W** of the adaptive system. Their modification allows the network to realize a variety of functions.

ANNs are adaptive systems with the power of a universal computer, i.e. they can realize an arbitrary mapping (association) of one vector space (inputs) to the other vector space (outputs). Many problems in science and engineering require guessing the global mapping from a set of data points (training examples). Given a set of examples, or statistical sample of data points, adaptive systems acquire an idea how the global mapping looks like. It is of great importance to understand that this mapping is not model-free, the choice of the processing functions of network elements determines the type of mappings that the systems can learn. Such understanding enables us to look for applications of adaptive systems that may lead to new results, hard to obtain with standard methods. Unfortunately rarely relevant mathematical theories such as theory of statistical decisions or approximation theory are invoked in neural networks papers. As we have recently shown [3] in most physical and chemical problems direct fitting techniques should be more accurate than predictions of trained neural networks because better approximation functions are used.

The motivation to develop FGM model came from analysis of the global mapping changes (the landscape of solutions) during training of neural networks and the desire to construct this mapping explicitly. The solution came from quantum chemistry where floating Gaussian functions are used since many years as the one-electron basis sets for molecular calculations. In the next section FGM model is introduced and presented in the form of a network of processing elements. In the third section the associative capabilities of FGM are illustrated on a classical example of schemata formation and on the recognition of spectra. In the fourth section an example of qualitative electric circuit analysis is given illustrating how general laws instead of specific examples may be used for training the FGM system. Finally, results for a classic problem of the two spirals are presented showing the capability of FGM to generalize unknown facts.

2. Floating Gaussian Mapping Model

One could avoid convergence problems and unpredictable classifications of neural networks by constructing the network mapping in an explicit way. The simplest functions with suitable properties are of the Gaussian type. Some other functions, like products of pairs of sigmoidal functions $\sigma(x)(1-\sigma(x))$, or the sigmoidal functions $\sigma(-\|X-D\|^2)$ are very similar to the Gaussians but only Gaussian functions are factorizable:

$$G(X,D) = e^{-\sum_{i=1}^{N}(X_i-D_i)^2/\sigma_i} = \prod_{i=1}^{N} e^{-(X_i-D_i)^2/\sigma_i}$$

Although the model I shall describe may work with other functions as well this factorization property is crucial for reduction of multidimensional searches to a series of one-dimensional searches in the retrieval of the data (facts). I have justified the use of localized processing functions instead of sigmoidal functions elsewhere [4]. Possible generalizations of this simplest choice of processing functions include asymmetric Gaussian functions, weighted sums of one-dimensional gaussians ("gaussian bars" [5]) and various combinations of sums and products of sigmoidal functions [6]. In this paper I will restrict myself to N-dimensional Gaussian functions only. Gaussian functions are centered on the data points $D=(D_1, D_2, ... D_N)$ with dispersions in each direction proportional to the error or uncertainty of the variables D_i. Each input variable defines a new dimension, the data vector is a point (vector) in N-dimensions and the data vector together with the associated uncertainties defines an ellipsoid in N-dimensional space, described by the density of the $G(X,D)$ function. From another point of view Gaussian processing functions may be regarded as the membership functions for fuzzy representation of the data points [7]. Indeed, one may show that there is a functional equivalence of the two approaches.

I will use the term "fact" for a collection D of *input* and *output* data that we want to store in the FGM adaptive system. Facts belong to the conceptual space, the internal space of the representations of the adaptive system. This conceptual space may have a very large dimensionality but it is finite. In case of human knowledge the number of concepts, or elements of reality that we are able to distinguish, is of the order of 10^5. Combinations of these elements create facts. The FGM function for a collection of facts $\boldsymbol{D}=\{D^p\}$ has the following general form:

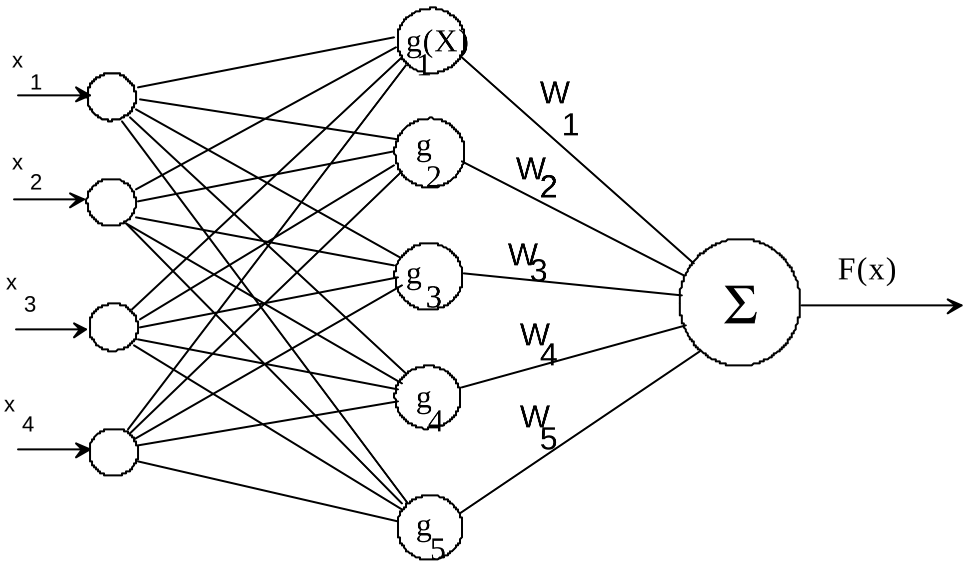

Fig.1 Example of a network realization of the Floating Gaussian Mapping

$$FGM(X,\mathbf{D}) = \sum_p W_p G(X, D^p) = \sum_p W_p \prod_i e^{-(X_i - D_i^p)^2/\sigma_i}$$

and does not vanish only around the data vectors (facts) **D** stored in the FGM function. The weights **W** and the dispersions **s** are the adaptive parameters defining FGM mapping for a given set of **D** input values. In some applications if the data values are noisy also the gaussian centers **D** may be treated as adaptive parameters, as it is done with the codebook vectors in the Learning Vector Quantization model [2]. In fact FGM may be treated as a fuzzy generalization [7] of LVQ method. In simple applications of the associative memory type it may be enough to treat only the dispersions as adjustable parameters.

Depending on the distribution of the data points and the dispersions of the functions the FGM values may be in most points *X* of the parameter space (space of all possible inputs) very small, meaning that no facts similar to *X* are known to the FGM. In this case FGM does not generalize but, if the answer different from "don't know" is given, it is given with the high degree of confidence. For large dispersions at most points of the parameter space the values will be non-zero and gradients will point in the direction of the closest fact, enabling generalization at a cost of decreased confidence. Although the model allows for many choices I am going to investigate in this paper only the simplest one: W_p weights will be taken as binary (i.e. connection is present or not) and the true facts are equivalent to FGM values ≥ 1.

In contrast to the Radial Basis Functions and similar approaches [8] Floating Gaussian Mapping network is created from examples (teaching or training phase) by adding new Gaussian *G(X,D)* nodes centered around distinct data vectors, i.e. outside the regions of the existing centers, and increasing the dispersion of gaussians if *X* falls in proximity of the existing *D*. This algorithm allows to avoid rapid growth of the number of network nodes with a large number of data facts. The network may be sparsely connected if many facts are stored and not all inputs are relevant for all facts. In Fig. 1 only a single hidden layer is shown but many network realizations are possible, including hierarchies of nodes performing some logical functions on facts or the two-layered design of the harmonium model [9]. There are also many possibilities of the FGM network adaptation: adding new gaussian centers, removing isolated gaussians, moving their positions and their dispersions.

Perhaps the most distinct feature of the FGM is the way facts are retrieved or inferences from partial input data are made. If there are many factors influencing the answer to a problem in which some features are fixed and others are unknown humans frequently reason by making a series of one-dimensional searches, i.e. assuming for a moment that only one additional factor is important, temporarily fixing the value of this factor and analyzing the next unknown factor. An expensive alternative used in many neural algorithms is to guess everything at once by searching for a stationary point in many dimensions using simulated annealing or similar procedures. The searching strategy in FGM is as follows:

1. Fix the value of the known factors (X_1, .. X_k)
2. Search for the possible values of X_{k+1} (non-vanishing FGM) as follows:
2.1 Examine all nodes assuming that (X_{k+2}, .. X_N) are irrelevant
2.2 Note the values of X_{k+1} for which FGM values are ≥ 1
2.3 Fix X_{k+1} at one of possible values and repeat the search for X_{k+2}.
3. If searches for all m=k+1,.. N give positive results than new fact has been found; if for some m no values of X_m lead to positive results move back to the m-1 level and try the next X_{m-1} value for which the positive result was obtained (step 2.3)

This algorithm will find all facts consistent with given values of the known factors. The depth of the search is equal to the number of unknown factors, which is usually not large. The number of facts checked is at most equal to the number of relevant facts, since not all nodes are connected to all inputs: some facts may be totally irrelevant since their input space may be completely orthogonal to the input space of the question at hand. All searches are one-dimensional due to the separability of the Gaussian functions: assumption that some inputs are irrelevant simply means that computing the response of a node they are not taken into account. At a given stage the value of only a single Gaussian factor is computed even if the total dimensionality of the Gaussian function *G(X,D)* is quite high.

Another way to simplify searching is to use a multi-scale approach to find quickly all possible areas of the conceptual space containing facts relevant to the problem. Before the search is started the dispersions are temporarily set to large values, making it easy to identify broad regions where detailed search is performed with the original values of dispersion. This "defocusing" behavior is also known from reasoning by humans.

3. Learning arbitrary associations

The simplest nontrivial problem for neural networks is the XOR (exclusive OR) since it defines mapping that is not linearly separable [9]. In the FGM linear separability is never an issue since all data points are defined in the conceptual space that includes inputs and outputs. In this case it is 3-dimensional space, therefore XOR facts may be represented as a combination of 4 Gaussian functions centered at the corners (0,0,1), (0,1,0), (1,0,1) and (1,1,1) of a cube. The fuzziness of this relation is controlled via the dispersion parameters of the Gaussians - some facts may be more sharply defined than the others.

Many examples of associations and retrieval of information from partial inputs, as given in the PDP books [9], are solved in a trivial way by the FGM model. Learning of schemata, like schemata for rooms or "Jets and Sharks" example are given for interactive activation model, constraint satisfaction and competitive learning models as illustrations. In the room schemata (cf. Vol. II, p. 22, [9]) 40 descriptors are given for five different kinds of rooms: living room, kitchen, office, bathroom, bedroom. One can create FGM mapping giving examples of room furniture

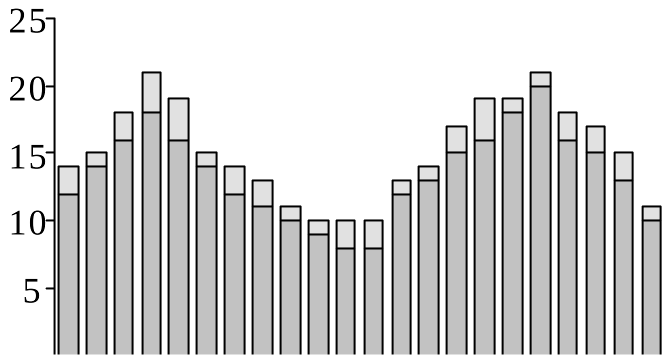

Fig. 2 Histogram of a spectrum with uncertainties corresponding to dispersions of Gaussian function in 22 dimensions.

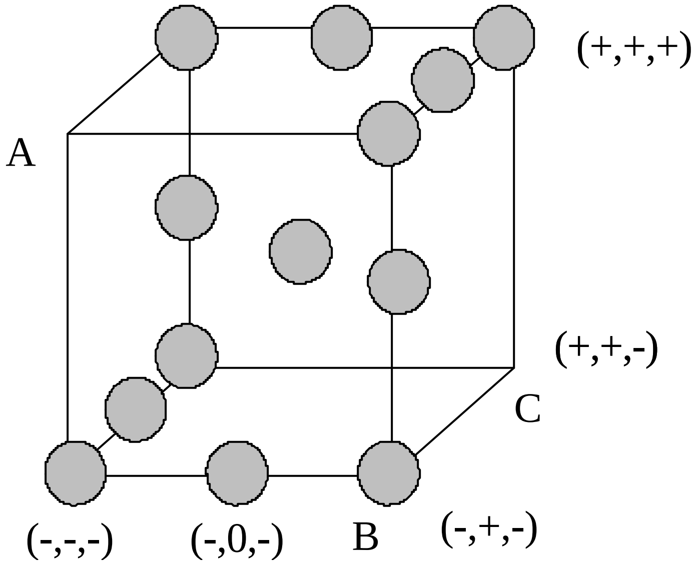

Fig. 3. Representation of a relation A=B·C or A=B+C in FGM model.

and other descriptors for these schematic rooms and retrieve a prototype room description from partial description. Most of the descriptors, like oven, computer, toilet, are of the binary type: present - not present, some have few values (room size may be very-large, large, medium, small, very-small). Treating all descriptors as binary 40-dimensional hypercube is obtained with 2^{40} possible states (corners). The 5 schemata for rooms correspond to more than 5 corners of this hypercube since such descriptors as walls or ceiling are always present and television set may be present or not present in all of the room types. However, in this 40-dimensional space there are only 5 areas, overlapping in some dimensions but well resolved in others, defining the schemata for rooms. We add one extra dimension for the room type, which is kept with other descriptors forming a fact in the FGM space. Fixing the room type all descriptors forming a room schemata are easily recovered in 40 searches, each a binary checkup answering the question: does it have such-and-such descriptor? Fixing one descriptor that is characteristic to some room schemata, like an oven, immediately recreates the whole schemata for kitchen since already at the highest level of searching only one type of a room gives positive answer. Fixing descriptors that could apply to many rooms, like telephone, will activate several searching path (at most 5) and for each one again a simple search involving just 40 steps is made. A question like "can the kitchen have a telephone" requires fixing the type of room variable at kitchen - this immediately leaves only one active FGM node. The dispersion of the "telephone" variable at this node determines the probability of the positive answer to the question and may be verbalized in such terms as "impossible", or "rather unlikely".

As an example of nontrivial associations that FGM is capable of let us consider a spectrum stored in a form of a histogram (Fig. 2). Since the spectrum corresponding to a given chemical system may be distorted or taken under different conditions each value of the histogram is given with an uncertainty bar equal to the dispersion of the Gaussian in the given direction. The Gaussian represents therefore a range of spectrums connected with the same system. A database of such spectra enables identification of many different systems from distorted spectra or partial spectra.

4. Learning from general laws

Frequently the knowledge that we have is derived from a set of examples. In many cases, especially in natural sciences, there are general laws that may be applied to a given situation. These laws may be either deduced from examples or stored as *a priori* knowledge or constraints on the type of internal representations. Neural networks are usually trained on examples while expert systems are based on the rules.

Solving problems we use knowledge contained in an equation in a qualitative way. Let us consider a specific example: Ohm's law $V=I \cdot R$. It involves 3 parameters, voltage V, current I and resistance R. Geometrical

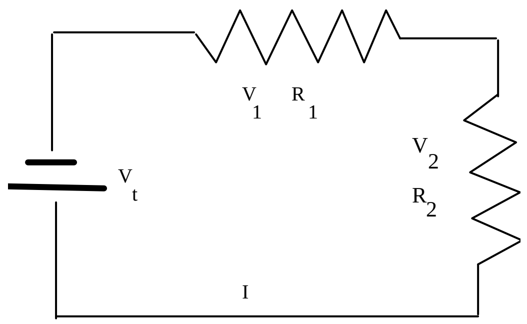

Fig. 4 A simple electrical circuit

interpretation of this law involves a hyperboloid or a set of hyperbolas on a plain for different values of *V*. This does not help us in using Ohm's law in practical cases. A set of training facts is derived and internalized as "intuition" from this law: when the current grows and the resistance is constant, what happens to the voltage? If we designate changes of *V, I* and *R* as + for increase, 0 for no change and – for decrease than the number of all possible combinations of the 3 values for the 3 variables is $3^3 = 27$. Ohm's law says in effect that 13 of them are true and 14 false, for example if *V* is constant *I* and *R* may not decrease or increase simultaneously. A convenient way of expressing these intuitions about Ohm's law or any other law of the form $A=B\times C$ or $A=B+C$ is to show the true facts in the FGM representation of knowledge (cf. Fig. 3). They may be either represented by 13 Gaussians localized in 3-dimensions or by 6 Gaussians in 2-dimensions and one Gaussian in 3-dimensions (in the center).

I will present now a slightly more complicated example of the FGM for representation of qualitative knowledge necessary for understanding simple electric circuits. Although the circuit shown in Fig. 4 is very simple untrained people need some time to answer questions related to the behavior of the circuit.

There are 5 relevant equations:

$$V_t = V_1 + V_2$$
$$R_t = R_1 + R_2$$
$$V_1 = I \cdot R_1 ; V_2 = I \cdot R_2 ; V_t = I \cdot R_t$$

Each equation has 3 variables. Five cubes, corresponding to each of the equations, are present in the 7-dimensional $(V_t,V_1,V_2,R_t,R_1,,R_2,I)$ space. A typical question that we may ask is (cf. Smolensky, in: [9]): if R_2 increases and V_t and R_1 are constant what happens with *I* and V_1, V_2 ? This example was originally given for the Boltzman machine and the harmony model type of neural network and is not so trivial to solve in these models: answering such a question is done via searches using simulated annealing and require lengthy computations, requiring many repetitions of the procedure since in the individual runs sometimes wrong results are obtained (local minima problem).

In the FGM the answer requires searching for non-zero values at a few points of the function $FGM(V_t=0,V_1,V_2,R_t,R_1=0,R_2=+,I)$ along the four (V_1,V_2,R_t,I) directions according to the algorithm described above. Since all 5 equations have to be fulfilled simultaneously FGM function is taken as a product of 5 Gaussian functions, each containing 13 terms corresponding to the internalized facts about Ohm and Kirchoff equations. It is a trivial exercise that I shall leave to the reader to find out that at most 3 short path are generated in the search procedure, only 8 nodes are visited and the only possible outcome is

$$FGM(V_t= 0, V_1 =0, V_2 =+, R_t =+, R_1 = 0, R_2 = +, I= -)$$

Experience in solving such problems involves selection of the order of unknown variables before searching. For example, taking total resistance R_t as the first unknown variable only one path is generated in the search procedure. This type of meta-knowledge may also be taken into account in the FGM model.

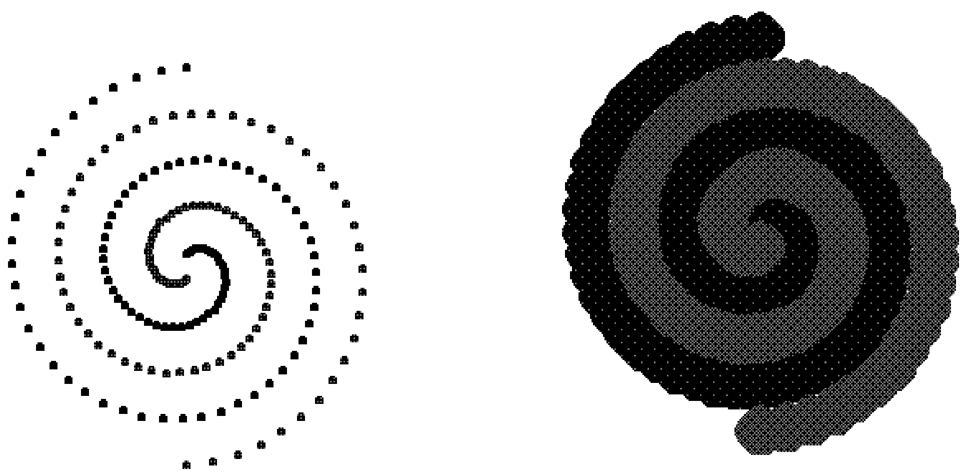

Fig. 5 FGM Solution to the 2-spiral problem.

5. The two-spiral problem

One of the hard problems for the backpropagation neural networks is the two-spirals classification problem. 194 points (x,y) are given, half of these points in class 1 and half in class 2, belonging to the two spirals (Fig. 5). Standard backpropagation algorithm [10] does not converge in this case and the cascade correlation algorithm, perhaps the best for this problem, produces [11] less than perfect results after 17000 epochs of learning. We have used the simplest version of FGM with spherical Gaussian units, although more sophisticated choice of units could easily reduce their number. The learning step involves setting up the Gaussians at the data points and the generalization step setting up their dispersions to half of the distance from the nearest neighbor. The results are shown in Fig. 5. If the dispersions are optimized for each individual data point in each direction in an independent way more than half of the points may be dropped without significant increase of errors.

6. Summary

Floating Gaussian Mapping is a new model of adaptive systems suitable for certain applications for which neural networks or expert systems are usually employed, offering interesting combination of the two technologies. Logical constraints are easily implemented in the associative mappings. Although FGM model is not biologically plausible it offers two features known from cognitive psychology. The dependence of the reasoning, or drawing inferences, on the search strategy is similar to what the humans do: trying several possibilities and than changing the strategy if it doesn't work. Defocusing mechanism for preliminary searches is the second interesting feature.

Many interesting problems for which neural networks have been used have simple solutions in the FGM model. I have given examples of learning from examples and from general laws, making simple associations and drawing logical inferences. Direct modeling of conceptual space followed by the subsequent fine tuning by reorganization of the positions and dispersions of Gaussian functions using self-organizing mapping algorithm [2] is a great advantage of the FGM model over the typical feedforward neural networks. The FGM model has great potential for development but at present we still do not understand its limitations.

Acknowledgments

This paper has been partially written during my summer visit to the Max-Planck Institute of Astrophysics in Garching bei München, Germany. It is a pleasure to thank prof. Geerd Diercksen for his hospitality and interest in this work.

References

[1] J.A. Anderson, and E. Rosenfeld, (Eds). Neurocomputing: Foundations of Research. (The MIT Press: Cambridge, MA.(1988)); J.A. Anderson, A. Pellionisz, and E. Rosenfeld, (Eds). Neurocomputing 2: Directions for Research. (The MIT Press: Cambridge, MA. 1990); M. Caudill and C. Butler, Naturally Intelligent Systems. (The MIT Press: Cambridge, Massachusetts, 1990);

[2] T. Kohonen, Proceedings of the IEEE 78 (1990) 1464

[3] W. Duch and G.H.F. Diercksen, Chem. Phys. (submitted)

[4] W. Duch, UMK-KMK-TR-4/93

[5] E. Hartman and J. D. Keeler, Neural Computation 3 (1991) 566

[6] W. Duch and G.H.F. Diercksen, Comp. Phys. Comm. (to be submitted)

[7] G.J. Klir and T.A. Folger, Fuzzy Sets, Uncertainity and Information. (Prentice Hall, NJ 1988)

[8] T. Poggio and F. Girosi, Proc. of the IEEE 78 (1990) 1481 ; M.J.D. Powell, "Radial basis functions for mulitvariable interpolation: a review", in: J.C. Mason and M.G. Cox, eds, Algorithms for Approximation. (Clarendon Press, Oxford 1987); E.J. Hartman, J.D. Keeler and J.M. Kowalski, Neural Computation 2 (1990) 210; J. Park and I.W. Sandberg, Neural Computations 3 (1991) 246

[9] J.L. McClelland and D.E Rumelhart, Explorations in Parallel Distributed Processing: Computational Models of Cognition and Perception. (The MIT Press, Cambridge, MA 1988)

[10] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, Learning representations by backpropagating errors. Nature, 323 (1986) 533-536; D.E. Rumelhart and J.L. McClelland, Parallel Distributed Processing: Explorations in the Microstructure of Cognition, (The MIT Press, Cambridge, MA 1986), Vol.1, pp 318-362; P.J. Werbos, Proc. of IEEE 78 (1990) 1550

[11] S.E. Fahlman and C. Lebiere, in "Advances in neural information processing systems, D. S. Touretzky (ed), 2, pp. 524-532 (Morgan Kaufman 1990)