Externally Growing Cell Structures for Pattern Classification

Guojian Cheng, Andreas Zell
University Tübingen
Wilhelm-Schickard-Institut for Computer Science
Dept. Computer Architecture
Köstlinstr. 6, D-72074 Tübingen, Germany
{cheng, zell}@informatik.uni-tuebingen.de

Abstract

Based on B. Fritzke's GCS (Growing Cell Structures) we present here a new incremental self-organizing neural network, the Externally Growing Cell Structures (EGCS). Our goals are to speed up the convergence and to improve the generalization performance. The mechanism of internally growing cells in EGCS is the same as in GCS. However, when the Maximum Resource Vertex (MRV) or the Maximum Error Vertex (MEV) is a boundary node, the new cell is grown externally. Simulation results on some neural network benchmarks (two-spiral problem, sonar mine/rock separation problem, Wisconsin breast cancer study) indicate that EGCS performs better than the original GCS, measured by classification rate, the required number of cells and epochs.

1 Introduction

Kohonen's Self-Organizing Maps (SOM) [12] generate mappings from high-dimensional signal spaces to lower dimensional topological structures. Their main features are formation of topology-preserving feature maps and approximation of the input probability distribution. Incremental artificial neural networks grow as they learn, and shrink as they forget. Based on SOM, B. Fritzke introduced an incremental self-organizing network with variable topology, known as Growing Cell Structures (GCS) [9]. The GCS has three main advantages over SOM: First, the network structure is determined automatically by the input patterns. Second, the network size needs not to be predefined. Third, all parameters of the model are constant. It is, therefore, not necessary to define a decay schedule as for SOM.

Since the first presentation of GCS in 1992 [6], many related algorithms of GCS have been developed, such as: dynamic cell structures [3], hierarchical growing cell structures [4], probabilistic growing cell structures [18], growing radial basis function networks [2], growing multi-dimensional SOM [15]. Performance comparisons of GCS with other neural network models have been made by B. Fritzke [8] and F. Hamker [11]. An implementation of the supervised growing cell structure on the CNAPS neurocomputer was made by J. Steffens and M. Kunze [16].

In order to speed up the convergence and to improve the generalization performance, we present a new incremental self-organizing neural network, the Externally Growing Cell Structures (EGCS). The idea of externally growing cells is taken from J. Blackmore's IGG (Incremental Grid Growing) [1]. The mechanism of internally growing cells in EGCS is the same as in GCS. However, when the Maximum Resource Vertex (MRV) or the Maximum Error Vertex (MEV) is a boundary node, the new cell is grown externally. Another characteristic of EGCS is the positioning of new cells. In GCS a new cell is inserted in the middle of the edge connecting the MRV and the most distant node in the direct topological neighbourhood of MRV. But in EGCS a variable position for both internally and externally grown cells is used. This idea is taken from J. Bruske's DCS (Dynamic Cell Structures) [3]. The exact location of its center of receptive field is calculated according to the ratio of the resource values. This gives an estimation of the resource values as if the new unit was in the network right from the start.

The organization of the paper is as follows: After a brief description of the GCS algorithm in section 2, the mechanism of the EGCS is presented in section 3. EGCS is then tested with some neural network benchmarks and the results are discussed in section 4. Some conclusions and plans for future research are given in section 5.

2 A Brief Description of the Growing Cell Structures (GCS)

The goal of a GCS network is to create a mapping of vector space $\mathcal{V}$ with a n-dimensional probability distribution $P(\xi)$ onto a discrete k-dimensional topological structure $\mathcal{A}$. This mapping should have the following properties: First, similar input signals are mapped onto topologically close elements of $\mathcal{A}$. Second, topologically close elements in $\mathcal{A}$ should have similar signals being mapped onto them. Third, regions of $\mathcal{V}$ where the probability density of the input vector distribution is higher should be represented by correspondingly more elements in $\mathcal{A}$. The initial topology of the network is a k-dimensional simplex. During self-organization new cells are added to the network and superfluous cells are removed. After each modification, the network consists solely of k-dimensional simplices (a line for $k = 1$, a triangle for $k = 2$, and a tetrahedron for $k = 3, \cdots$).

Every cell has a n-dimensional vector $\mathbf{v}_i$ attached, which denotes the position of the cell i in the input vector space $\mathcal{V}$. The adaption of the synaptic vectors is done as in Kohonen networks. But there are two important differences: First, only the best matching unit (bmu) and its direct topological neighbouring (dtn) cells are adapted. Second, the adaption strength for bmu and dtn cells is constant over time (ε_{bmu} and ε_{dtn} respectively). Each cell has a counter variable τ_i attached, which contains the number of times it has been best matching unit.

When every cell has the same probability of being best-matching unit for the current input vector ξ, a structure with the synaptic vectors $\mathbf{v}_i$ distributed according to $P(\xi)$ can be obtained. With local signal counters, an estimation of $P(\xi)$ can be computed, namely the relative frequency of input signals h_c received by a certain cell c. All cells should have similar relative signal frequencies. A high value of h_c indicates a good position to insert a new cell since the new cell is likely to reduce this high value to a certain degree.

After a fixed number (λ) of adaption steps the vertex with the highest relative signal frequency (V_{max}) and the most distant vertex (V_{mdn}) in the direct topological neighbourhood of V_{max} are determined (V_x represents the cell x in the network or the vertex in the graph and $\mathbf{v}_x$ represents the synaptic vector). A new cell V_{new} is inserted between V_{max} and V_{mdn} and connected to the other cells so that the resulting structure again consists only of k-dimensional simplices. The synaptic vector $\mathbf{v}_{new}$ of the new cell V_{new} is initialized as the mean of the two previously connected cells: $\mathbf{v}_{new} = (\mathbf{v}_{max} + \mathbf{v}_{mdn})/2$.

Through a combination of GCS with radial basis functions (RBF) B. Fritzke developed the Supervised GCS (SGCS) [9]. It consists of three layers, including one hidden layer. The units of the hidden layer have a Gaussian activation function. The synaptic vector $\mathbf{v}_c$ of each unit defines the centre of the Gaussian function and an additional parameter σ_c defines its size. In addition, the hidden layer forms a topological structure defined through hypertetrahedrons which are a special type of neighbourhood definition of the Kohonen feature map. The neighbourhood also defines the standard deviation of the Gaussian activation function by using the mean distance between a unit and its neighbours. The output of the network is simply computed by a weighted sum of the unit activations in the hidden layer. The output unit with the largest activation gives the classification result.

The SGCS has three main advantages over other traditional supervised neural networks: First, number, diameter and position of RBF units are determined automatically through the growth process, which can be stopped as soon as the network performance is good enough. Second, since positioning of RBF units and supervised training of connection weights is performed in parallel, the current classification error can be used to determine where to insert new RBF units. Third, the networks can generalize well and have a relatively small size [11].

3 Externally Growing Cell Structures (EGCS)

3.1 Resource Variable and Insertion Criterion

We can use GCS with different means. It is essential that new cells are inserted in such a way that the expected value of a certain error measure or signal counter or resource value (the accumulated resource values play a nutrition-like role for the cell growing), becomes equal for all neurons [9]. Appropriate resources must have the property that the insertion of a new neuron V_{new} near an existing neuron V_{max} reduces the expected value of the resource of V_{max}.

The quantization error generated by the neurons is generally used as insertion criterion. This is the accumulated squared distance between the synaptic vector of the neuron and all input signals $(\xi_1, \dots, \xi_n)$ being

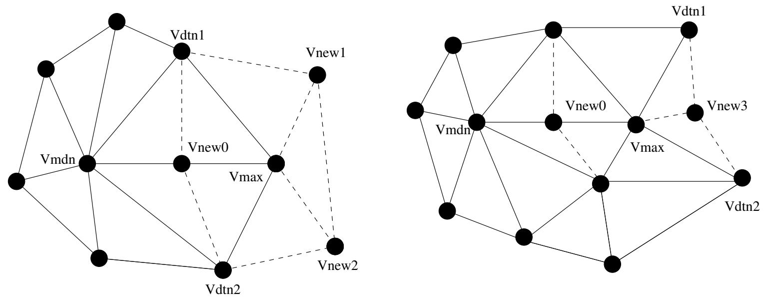

Figure 1. The EGCS Structure. *left*: Convex weight externally growing structure. *right*: Concave weight externally growing structure.

mapped onto the neuron. For unsupervised learning the resource variable of the bmu cell is increased by the squared error: $\Delta\mathcal{E}_{bmu} = \|\xi - \mathbf{v}_{bmu}\|^2$. By using this measure as insertion criterion, new neurons are inserted near the neurons which input signals are very different from their synaptic vectors. For supervised learning the resource variable of the current bmu cell is updated by adding the overall squared error between actual output $\mathbf{o} = (o_1, \dots, o_m)$ and desired output $\zeta = (\zeta_1, \dots, \zeta_m)$ to it: $\Delta\mathcal{E}_{bmu} = \|\zeta - \mathbf{o}\|^2$.

3.2 The Mechanism of Externally Growing Cell Structures

Figure 1 shows the EGCS Structure. The boundary-MEV is marked with V_{max}. It has three direct topological neighbours: V_{dtn1}, V_{dtn2} and V_{mdn}. V_{mdn} is the most distant neighbour to V_{max}. The internal cell growing mechanism of EGCS is the same as in GCS if V_{max} isn't located at the boundary. However, if the MEV is located at the boundary, then the new cell grows externally. According to the convexity-concavity of the MEV there are two cases: convex or concave external growth. Depending upon whether internally growings happened or not at the same time with externally growing actions for the MEV, two variations of EGCS exist: externally growing only (EGCS1) and externally growing at the same time with internally growing (EGCS2).

3.2.1 Convex Weight Externally Growing

Figure 1 *left* shows the convex weight externally growing structure. Two new cells, V_{new1} and V_{new2}, are externally grown from V_{max}. The new weight vectors, $\mathbf{v}_{new1}$ and $\mathbf{v}_{new2}$, are induced as following:

Suppose V_{max} is the mid-point of $V_{dtn1}V_{new2}$ or $V_{dtn2}V_{new1}$. We call this full distance growing (growing factor $f_{ex} = 1.0$). It is obvious that:

$$\mathbf{v}_{new1} = 2\mathbf{v}_{max} - \mathbf{v}_{dtn2}, \quad \mathbf{v}_{new2} = 2\mathbf{v}_{max} - \mathbf{v}_{dtn1}.$$

When the growing distance factor f_{ex1} and f_{ex2} is variable, we call this variable distance growing. In this case:

$$\mathbf{v}_{new1} = (1 + f_{ex1})\mathbf{v}_{max} - f_{ex1}\mathbf{v}_{dtn2},$$

$$\mathbf{v}_{new2} = (1 + f_{ex2})\mathbf{v}_{max} - f_{ex2}\mathbf{v}_{dtn1},$$

$$f_{ex1} = \frac{\mathcal{E}_{dtn1}}{\mathcal{E}_{dtn1} + \mathcal{E}_{max}}, \qquad f_{ex2} = \frac{\mathcal{E}_{dtn2}}{\mathcal{E}_{dtn2} + \mathcal{E}_{max}}.$$

That is to say that the exact location of the centre of receptive fields $\mathbf{v}_{new1}$ and $\mathbf{v}_{new2}$ is calculated according to the ratio of the resource values $\mathcal{E}_{dtn1}, \mathcal{E}_{dtn2}$ and $\mathcal{E}_{max}$. In fact, the optimum value of f_{ex1} or f_{ex2} has been determined empirically to be around $1/2$ and we call this half distance growing. In this case:

$$\mathbf{v}_{new1} = \frac{1}{2}(3\mathbf{v}_{max} - \mathbf{v}_{dtn2}),$$

$$\mathbf{v}_{new2} = \frac{1}{2}(3\mathbf{v}_{max} - \mathbf{v}_{dtn1}).$$

The new cell V_{new1} is then connected to V_{dtn1} and V_{max}. V_{new2} is connected to V_{dtn2} and V_{max}. At last two new cells, V_{new1} and V_{new2}, are connected to each other.

3.2.2 Concave Weight Externally Growing

Figure 1 *right* shows the concave weight externally growing structure. One new cell V_{new3} is externally grown from V_{max}. The new cell's weight vector $\mathbf{v}_{new3}$ is induced as follows:

Suppose V_{new3} is the mid-point of $V_{dtn1}V_{dtn2}$. We call

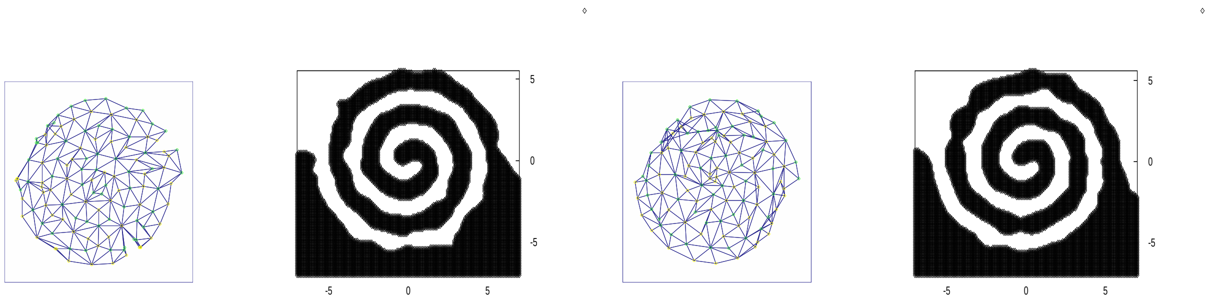

Figure 2. Performance of the EGCS1 and EGCS2 on the two-spiral benchmark. *left half*: Final EGCS1 network with 122 cells and its decision regions. *right half*: Final EGCS2 network with 116 cells and its decision regions.

this full distance growing (growing factor $f_{ex3} = 1.0$). We have:

$$\mathbf{v}_{new3} = (\mathbf{v}_{dtn1} + \mathbf{v}_{dtn2})/2.$$

For variable distance growing we obtain:

$$\mathbf{v}_{new3} = (1 - f_{ex3})\mathbf{v}_{max} + \frac{1}{2} f_{ex3}(\mathbf{v}_{dtn1} + \mathbf{v}_{dtn2}),$$

$$f_{ex3} = \frac{\mathcal{E}_{dtn1} + \mathcal{E}_{dtn2}}{\mathcal{E}_{dtn1} + \mathcal{E}_{dtn2} + \mathcal{E}_{max}}.$$

In fact, the optimum value of f_{ex3} has been determined empirically to be around $2/3$. When the growing factor $f_{ex3} = 2/3$, this may also be called centroid growing, because V_{max} locates the centroid of the triangle $V_{max}V_{dtn1}V_{dtn2}$. In this case:

$$\mathbf{v}_{new3} = \frac{1}{3}(\mathbf{v}_{dtn1} + \mathbf{v}_{dtn2} + \mathbf{v}_{max}).$$

The new cell V_{new3} is directly connected to V_{dtn1}, V_{dtn2} and V_{max}.

3.2.3 EGCS1 and EGCS2

Both EGCS1 and EGCS2 are the same in the convex/concave weight externally growing. The difference is in internally growing. For EGCS1 there is no new cell V_{new0}. But for EGCS2 a new cell V_{new0} is at the same time inserted at the edge $V_{mdn}V_{max}$. This is called internally growing and its mechanism is the same as in the original GCS. Just as shown in Figure 1, a new cell V_{new0} is inserted at the edge $V_{mdn}V_{max}$. The new weight vector $\mathbf{v}_{new0}$ is calculated as:

$$\mathbf{v}_{new0} = (\mathbf{v}_{mdn} + \mathbf{v}_{max})/2.$$

In this case the internally growing distance factor is $f_{in} = 1/2$. When the internally growing distance factor f_{in} is variable we have:

$$\mathbf{v}_{new0} = (1 - f_{in})\mathbf{v}_{max} + f_{in}\mathbf{v}_{mdn},$$

$$f_{in} = \frac{\mathcal{E}_{mdn}}{\mathcal{E}_{mdn} + \mathcal{E}_{max}}.$$

The new cell V_{new0} is then connected to V_{mdn} and V_{max}. The original connection between V_{mdn} and V_{max} should be deleted. At last the new cell V_{new0} is connected to the common topological neighbours of V_{mdn} and V_{max}.

4 Supervised Learning Simulation Results

We applied our supervised EGCS1 and EGCS2 algorithms to three neural network learning benchmarks: the two-spiral problem, the sonar mine/rock separation and the Wisconsin breast cancer study. The first two benchmarks are from the CMU database [19], the last one is from PROBEN1 [14].

4.1 Two Spirals Problem

The task is to learn to discriminate between two sets of training points which lie on two distinct spirals in the x-y plane. These spirals coil three times around the origin and around one another. The training set has 194 examples. No test set is provided.

In order to solve the two-spiral problem we generated a two-dimensional EGCS. Figure 2 shows the EGCS1 and EGCS2 and their corresponding decision regions. Black indicates assignment to the first, white

Network Model	Number of epochs	Reported in
BP (SSE)	20000	K. J. Lang [13]
BP (CE)	11000	K. J. Lang [13]
Quickprop	7900	K. J. Lang [13]
Cascor + RPROP	2437	N. K. Treadgold [17]
Cascor	1700	S. E. Fahlman [5]
SGCS	180	B. Fritzke [7]
DCS-GCS	177	J. Bruske [3]
EGCS1	104	this article
EGCS2	86	this article

Table 1. Epochs for Supervised Learning of the Two-Spirals Problem. BP – Backpropagation, SSE – Sum Squared Error, RPROP – Resilient Backpropagation, Cascor – Cascade Correlation.

assignment to the second spiral. The classification error on the training set dropped to 0 and the accumulated squared output error served for resource updating, $\Delta\mathcal{E}_{bmu} = (||\zeta - \mathbf{o}||^2)/2, \zeta, \mathbf{o} \in \{-1, 1\}$.

It took EGCS1 104 training epochs and 122 growing cells until the network achieved zero classification error. In EGCS2 it took 86 epochs and 116 cells. For the original SGCS it took 180 epochs and 145 growing cells. Table 1 lists the number of epochs for the two-spiral problem for some earlier methods and for our approach. As it can be seen the number of epochs required by the EGCS1 and EGCS2 is about two times smaller than for original SGCS or DCS-GCS.

4.2 Mines vs. Rocks Classification with Sonar

The Sonar data set was used by Gorman and Sejnowski in their study of the classification of sonar signals using a neural network [10]. The task is to train a network to discriminate between sonar signals bounced off a metal cylinder and those bounced off a roughly cylindrical rock. The input vector is 60 dimensional and we employed a 1-dimensional output vector. The training and the test set contain 104 examples each. Gorman and Sejnowski report best results of 90.4% correctly classified test examples for a standard BP network with 12 hidden units and 82.7% for a nearest neighbour classifier.

In our simulation the accumulated squared output error served as resource updating: $\Delta\mathcal{E}_{bmu} = ||\zeta - \mathbf{o}||^2, \zeta, \mathbf{o} \in \{-1, 1\}$. The performance comparison of EGCS with other networks is given in Table 2. We obtained 94.2% correctly classified test samples with only 111 epochs and 123 grown cells in the EGCS1 network. At the same time, the classification error on the training set dropped to 0. In the EGCS2 network the accuracy in the test set was 93.3% after 107 epochs and 129 grown cells while the classification error on the training set dropped to 0. By comparison, after 120 epochs the SGCS reached 90.4% accuracy for test samples and the classification error on the training set fell to 6.7%. Supervised DCS-GCS of Bruske reaches an accuracy of 91.4% in test samples after 95 epochs of training [3].

4.3 Wisconsin Breast Cancer Study

The database of cancer classification problems is adopted from the PROBEN1 benchmark[14]. For the application of the validation method the datasets were divided into three sets: training set, validation set and test set which contained 350, 175 and 174 patterns, respectively.

The classification problem is diagnosing breast cancer, which is originally obtained from the University of Wisconsin Hospitals. The diagnosis is derived from the descriptions gathered by microscopic examination. These descriptions include the clump thickness, the uniformity of cell size and cell shape, the amount of marginal adhesion, and the frequency of bare nuclei. The dataset consists of 9 inputs and 699 examples.

For a better comparison we used the same stopping point for SGCS, EGCS1 and EGCS2. For cancer1 the training was stopped after 65 epochs. For cancer2 and cancer3 it were 55 and 45. Table 3 shows the difference between SGCS and EGCS1 and EGCS2 for this problem. The generalization performance is estimated by the error with respect to the testing set. Although the number of cells required by EGCS is larger than by SGCS, the generalization performance of EGCS is much

Algorithm	Epochs	Training Set MSE, CR	Testing Set CR
KNN			82.7%
MLP+BP			90.4%
DCS-GCS	95		91.4%
SGCS	120	0.224, 93.3%	90.4%
EGCS1	111	0.072, 100%	94.2%
EGCS2	107	0.055, 100%	93.3%

Table 2. Comparison of EGCS1 and EGCS2 with SGCS for the sonar mine/rock separation problem. KNN – K-Nearest Neighbour Classifier, MSE – Mean of Square Error, CR – Classification Rate.

Database	Algorithm	Epochs	Growing Cells	Training Set MSE, CR	Validation Set MSE, CR	Testing Set CR
Cancer1	SGCS	65	67	0.077, 92.86%	0.073, 92.57%	91.95%
	EGCS1		75	0.066, 94.86%	0.052, 95.43%	96.55%
	EGCS2		83	0.055, 97.14%	0.063, 96.00%	96.55%
Cancer2	SGCS	55	57	0.076, 94.00%	0.072, 94.86%	91.95%
	EGCS1		76	0.064, 96.57%	0.071, 95.43%	92.53%
	EGCS2		69	0.060, 96.00%	0.067, 95.43%	92.53%
Cancer3	SGCS	45	47	0.056, 96.57%	0.087, 93.71%	94.25%
	EGCS1		54	0.054, 96.29%	0.085, 96.57%	95.98%
	EGCS2		65	0.050, 98.00%	0.075, 95.43%	95.40%

Table 3. Comparison of EGCS1 and EGCS2 with SGCS for the Wisconsin cancer classification problem.

better than the latter. For example, the classification rate is increased from 91.95% to 96.55% for cancer1.

5 Conclusions

In this paper we have developed a novel dynamic self-organizing neural network that extends Fritzke's GCS network. The main advantage of GCS is that it can automatically find a network structure and size suitable for a given classification pattern through cell growing. The EGCS algorithm combines the internal-cell growing mechanism of Fritzke's growing cell structure and the external-cell growing mechanism of Blackmore's incremental growing grid. The position of cell insertion is calculated according to the ratio of the resource values of two end-nodes. Comparative experimental results on the three benchmark datasets revealed that EGCS is both effective and efficient as a learning algorithm for classification problems by internally and externally grown cells. EGCS can dynamically produce a self-organizing network with smaller size and better generalization performance and rapid convergence capacity. Its disadvantage is slightly higher computation costs due to the computation of convexity-concavity of the maximum resource vertex in each insertion step. Our future projects include further improvement and testing on EGCS and its application in the fields of function approximation and time series prediction.

6 Acknowledgments

We would like to thank B. Fritzke, M. Kunze, F. Hamker, J. Bruske and L. Prechelt for valuable and constructive discussions about their growing neural network algorithm and neural network benchmark problems. We also wish to thank our colleagues Igor Fischer, Jürgen Wakunda, Fred Rapp and Jutta Huhse for their help in programming and proof reading. The first author also thanks German KAAD (Katholischer Akademischer Ausländer-Dienst) very much for the scholarship.

References

[1] J. Blackmore. Visualizing high-dimensional structure with the incremental grid growing neural network. Tech-

nical Report AI95-238, University of Texas, Austin, Aug. 1, 1995.

[2] A. G. E. Blanzieri and P. Katemkamp. Growing radial basis function networks. In *Proc. of the 4th Workshop on Learning Robots*, Karlsruhe, Germany, Dec. 1995.

[3] J. Bruske and G. Sommer. Dynamic cell structure learns perfectly topology preserving map. *Neural Computation*, 7(4):845–865, 1995.

[4] V. Burzevski and C. K. Mohan. Hierarchical growing cell structures. In *ICNN96: Proc. Int'l. Conf. Neural Networks*, Washington D.C., June 1996. Tech. Rep., Syracuse University.

[5] S. E. Fahlman and C. Lebiere. The cascade-correlation learning architecture. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems 2. Proceedings of the 1989 Conference*, pages 524–532, San Mateo, CA, 1990. Morgan Kaufmann.

[6] B. Fritzke. Growing cell structures – a self-organizing network in k dimensions. In I. Aleksander and J. Taylor, editors, *Artificial Neural Networks II*, pages 1051–1056. North-Holland, Amsterdam, 1992.

[7] B. Fritzke. Growing cell structures – a self-organizing network for unsupervised and supervised learning. TR-93-026, International Computer Science Institute, Berkeley, 1993.

[8] B. Fritzke. Kohonen feature maps and growing cell structures – a performance comparison. In L. Giles, S. Hanson, and J. Cowan, editors, *Advances in Neural Information Processing Systems 5*, pages 123–130. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[9] B. Fritzke. Growing cell structures – a self-organizing network for unsupervised and supervised learning. *Neural Networks*, 7(9):1441–1460, 1994.

[10] R. P. Gorman and T. J. Sejnowski. Analysis of hidden units in a layered network trained to classify sonar targets. *Neural Networks*, 1:75–89, 1988.

[11] F. Hamker and D. Heinke. Implementation and comparison of growing neural gas, growing cell structures and fuzzy artmap. In *Technical report, Schriftenreihe des FG Neuroinformatik der TU Ilmenau*. Report 1/97, ISSN 0945-7518, 1997.

[12] T. Kohonen. *Self-Organizing Maps*. Springer, Berlin, Heidelberg, 1995. (Second Extended Edition 1997).

[13] K. J. Lang and M. J. Witbrock. Learning to tell two spirals apart. In *Proc. of the 1988 Connectionist Summer School*. Morgan Kaufmann, 1988.

[14] L. Prechelt. PROBEN1 — A set of benchmarks and benchmarking rules for neural network training algorithms. Technical Report 21/94, Fakultät für Informatik, Universität Karlsruhe, Karlsruhe, Germany, Sept. 1994.

[15] U. Seiffert and B. Michaelis. Growing 3D-SOMs with 2D-input layer as a classification tool in a motion detection system. *International Journal of Neural Systems*, 8(1):81–9, 1997.

[16] J. Steffens and M. Kunze. Implementation of the supervised growing cell structure on the CNAPS neurocomputer. In F. Fogelman-Soulie and P. Gallinari, editors, *ICANN '95. International Conference on Artificial Neural Networks. Neuronimes '95 Scientific Conference*, volume 2, pages 51–6, Paris, France, 1995. EC2 & Cie.

[17] N. Treadgold and T. Gedeon. A cascade network algorithm employing progressinve rprop. In *Proceedings International Work-Conference on Artificial and Natural Neural Networks (IWANN'97)*. Canary Islands, 1997.

[18] N. A. Vlassis, A. Dimopoulos, and G. Papakonstantinou. The probabilistic growing cell structures algorithm. *Lecture Notes in Computer Science*, 1327:P649, 1997.

[19] M. White and S.Fahlman. Cmu repository of neural network benchmarks. Technical report, The Carnegie Mellon University, 1997.