Life-long learning in incremental neural networks

Fred Henrik Hamker[1]
Technische Universität Ilmenau, Neuroinformatik, D-98684 Ilmenau, Germany
http://cortex.informatik.tu-ilmenau.de/~fred
e-mail: fred@informatik.tu-ilmenau.de

Abstract
This approach presents a possible solution to the stability-plasticity dilemma in incremental neural networks with a local insertion criterion. The main advantages are i) the capability of life-long learning, i.e., learning throughout the entire lifetime of a neural network, ii) stability in a stationary environment and iii) plasticity in a non-stationary environment, but only if the current knowlege does not fit the need of the task.
Thus, the network structures its internal representation not like a copy of the environment but in order to fulfill the current task.
Keywords: Life-long learning, stability-plasticity dilemma, incremental neural networks, Growing Neural Gas, Dynamic Cell Structures

1 Introduction

Learning is one of the main issues of artificial neural network design. It describes a mechanism by which a system obtains a representation of its environment. Recent research addresses the topic of on-line learning, incremental learning and life-long learning, which all discuss the same problem but emphasize different aspects. The necessity for on-line learning, in which the couplings of the network are updated after the presentation of each example, arises if not all training patterns are available all the time (Freeman and Saad, 1997; Heskes and Kappen, 1993). Most publications referring to on-line learning focus on the role of the learning rule and the convergence of the learning process, but stop learning when a performance criterion is reached. For systems, like robots, which are faced with patterns during their entire lifetime, studying on-line learning in contexts such as a changing environment (Heskes and Kappen, 1993) encounters the problems of stability and plasticity. Incremental learning addresses the ability of repeatedly training a network with new data, without destroying the old prototype pattern. Life-long learning, or also called continuous learning, emphasizes learning throughout the entire life-time and has to cope with changing environments and overlapping decision areas. It is not sufficient to only follow a non-stationary input distribution like (Fritzke,

---

1 Since September 1998 he joined the medical data analysis project MEDAN at the J.W.Goethe-Universität Frankfurt (email: hamker@cs.uni-frankfurt.de).

International Journal of Computing Anticipatory Systems, Volume 3, 1999
Edited by D. M. Dubois, CHAOS, Liège, Belgium, ISSN 1373-5411 ISBN 2-9600179-4-3

1997), life-long learning has to solve the stability-plasticity dilemma, which demands the adaption to new patterns and the preservation of old patterns.
Networks with a local or distributed representation of knowledge appear to be good candidates for life-long learning scenarios. One type of a local representation of knowledge utilized in recent literature of on-line learning are RBF's (Freeman and Saad, 1997) or similar networks (Gaussier and Zrehen, 1994). Nevertheless, they have a fixed number of nodes which has to be determined by the designer.
Incremental networks have the advantage that the number of nodes is also a result of learning by doing. The most important question in life-long learning incremental networks concerns the rule of insertion. ART networks, like FAM (Carpenter et al., 1992) insert new nodes based on a similarity measure. Other families of incremental networks use an error measure to insert new nodes. They can be subdivided into local error based insertion rules like Growing Cell Structures (GCS) (Fritzke, 1994), Growing Neural Gas (GNG) (Fritzke, 1995), Dynamic Cell Structures (DCS) (Bruske and Sommer, 1995) and global error based insertion rules like Cascade-Correlation (Fahlman and Lebiere, 1989). Inserting new nodes solely depending on the similarity of the input pattern leads to a purely sensor-based representation, which does not reflect the requirements of further processing stages. In contrast, an error-based insertion adapts the representation depending on the task and therefore leads to a task-based representation (Hamker and Gross, 1997). Compared to a global insertion criterion, a local criterion has the important advantage that insertion can be controlled locally. Summarizing, incremental networks with a local error based insertion rule are optimal candidates for life-long learning – but only if the insertion of new nodes can be managed properly.

2 General approach

On the one hand, incremental networks are not allowed to grow permanently. On the other hand, growing is an important feature to decrease the error of the net for the task and to adapt to changing environments. According to Grossberg (Grossberg, 1988) a switching-off of plasticity is a problem in nonstationary environments. But for the type of incremental networks with a local error based insertion rule, like GCS, GNG and DCS, an error-based learning of the insertion parameters is proposed to dynamically and locally control the stability and plasticity in the network. For this reason, each node not only owns an averaged longterm error counter, it is also equipped with an insertion threshold and an averaged longterm error counter at the moment of the last insertion (insertion error). The learning of the insertion parameter can be explained by an insertion evaluation cycle (Figure 1). By adaptation of an insertion threshold based on the evaluation of previous insertions, the network learns locally when it is useful to insert further nodes or to stop insertion.
The definition of the error counters as averaged error counters similar to (Ahrns et al., 1995) leads to an error measure that is independent of the input probability density in contrast to the error measure in (Fritzke, 1994; Fritzke, 1995; Bruske and Sommer, 1995). It has the advantage that the error is independent of the input probability density, which is important for life-long learning.

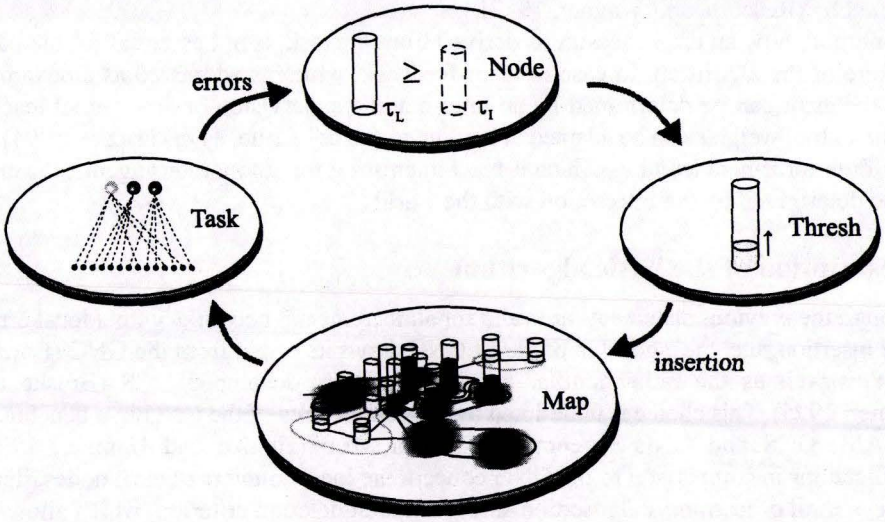


Fig. 1. Insertion evaluation cycle. The average long time error τ_L of the task is compared to the error at the moment of the last insertion τ_I. If this error is greater or equal, the insertion was not successful and the insertion threshold τ_h is increased. If the threshold reaches the average long time error, a further insertion at that location is not possible. To permit exploration in the future, the threshold can be decreased with a large time constant.

Another aspect concerns the adaptivity of the nodes. In (Ahrns et al., 1995), an error-modulated Kohonen type learning rule was used to achieve a uniform approximation error independent of the input probability density. Here, the modulation depends on the ratio of the average long time error and the average short time error and aims at reducing fluctuations when the input probability does not change any more. This means a node learns more, if the input probability changes and new errors occur.

Furthermore, a deletion criterion is introduced to remove redundant nodes. Candidates for deletion are located nearby in the input space and are responsible for similar outputs. For tasks with real-time demands the deletion criterion allows to restrict the number of nodes to an upper bound: By a simultaneous insertion of a new node and the deletion of the "worst" node, the nodes of the network are optimally fitted for static as well as for changing environments.

Interestingly, learning and insertion in the Life-long Learning Growing Neural Gas (LLGNG) shows similarities to the reward-based control of the plasticity of activated Hebbian synapses in biology. While the reward is usually delayed, the ratio of the average long time error and the average short time error reflects very well changes of the expected error i.e., the average long time error. Similarly, the insertion of new nodes depends on the difference between the predicted error, i.e., the insertion threshold and the actual error, i.e. the average long time error. The basic rule of learning behind learning and insertion in the LLGNG is that 'organisms only learn when events violate their expectations', previously

assumed by (Rescorla and Wagner, 1972).
The method, how an error measure is derived from the task, is not essential for the basic structure of the algorithm. In case of error-feedback, which is addressed as an example here, the error can be determined by an inter-module supervision or an external teacher and the output weights can be adapted according to the delta rule, as in (Fritzke, 1994). In case of reinforcement learning, which is most interesting for autonomous agents, the error can be determined by the interaction with the world.

3 Description of the basic algorithm

Although the previous statements are valid for all incremental networks with a local error-based insertion rule, the algorithm of the LLGNG draws its origin from the GNG (Fritzke, 1995) as well as the rather similar but independently developed DCS (Bruske and Sommer, 1995). This choice is underlined by the good results of the GNG in a benchmark on FAM, GCS and GNG in comparison to a MLP (Heinke and Hamker, 1998). Modifications in comparison to the GNG concern the local counters of each node (Figure 2), the control of learning and insertion, and an explicit deletion criterion, which allows to steer the density of the nodes considering their output-weight similarity. The network consists of two layers. The input determines the representation layer, which is followed by an output or task layer. The representation layer is described by a graph **G**, in which the set of neighbors N_i of a node i is defined by all nodes who share an edge with the node i.

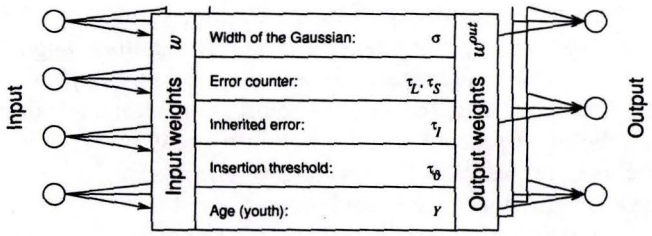


Fig. 2. Node of the life-long GNG. Besides the width of the Gaussian each nodes owns a longtern error counter τ_L, a shortterm error counter τ_S, the inherited error at the moment of insertion τ_I, an insertion threshold τ_θ, and the youth of the node Y, which decreases exponentially with the time constant T_γ from one to zero when the node was best matching. Despite the inherited error, which remains fixed until the node is selected for insertion again, the error counters are defined as moving averages with their individual time constant.

Adaptation of the representation layer

- For all nodes i, calculate the Euclidian distance d_i of the input $\boldsymbol{x}$ to the weight vector $\boldsymbol{w}_i$ and locate the best matching unit b and the second best s (equal to (Fritzke, 1995)):

$$d_b = \min_{i \in G} (d_i); \qquad d_s = \min_{i \in G, i \neq b} (d_i); \qquad d_i = \|\boldsymbol{x} - \boldsymbol{w}_i\| \qquad \forall i \in G$$

- Calculate the activation of all nodes y_i with a Gaussian function (Fritzke, 1994)):

$$y_i := e^{-\frac{\|x-w_i\|^2}{\sigma_i^2}} \quad ; \quad \sigma_i := \frac{1}{\|N_i\|}\sum_{j\in N_i}\|w_i - w_j\| \qquad \forall i \in G$$

- Determine the quality measure for learning B^L of the best node b and its neighbors $c \in N_b$:

$$B^L_{(b/c)} := \frac{\tau_{S(b/c)} \cdot 1}{\tau_{L(b/c)} \cdot 1} \qquad \forall c \in N_b$$

- Determine the input learning rate η^i of the best node and its neighbors from the quality measure B^L, the youth Y, the learning rate of the winner η_b and the neighbors η_n and the input adaptation threshold ϑ^i_L:

$$\eta^i_{(b/c)} := \begin{cases} 0 & \text{if} \quad \alpha^i_{(b/c)} < 0 \\ \eta_{(b/n)} & \text{if} \quad \alpha^i_{(b/c)} > 1 \\ \alpha^i_{(b/c)} \cdot \eta_{(b/n)} & \text{else} \end{cases} \qquad \alpha^i_{(b/c)} := \frac{B^L_{(b/c)}}{1 - \vartheta^i_L} \cdot Y_{(b/c)} - 1 \qquad \forall c \in N_b$$

and allow a minimal learning rate of the input weights determined by ϑ_M:

$$\eta^{i'}_{(b/c)} := \max(\eta_{M(b/c)}, \eta^i_{(b/c)}) \; ; \qquad \eta_{M(b/c)} := \eta^i_{(b/c)} \cdot (1 - y_{(b/c)}) \cdot \vartheta_M \qquad \forall c \in N_b$$

- Increase matching for b and its neighbors $c \in N_b$ (similar to (Fritzke, 1995)):

$$\Delta w_b := \eta^{i'}_b (x - w_b)$$
$$\Delta w_c := \eta^{i'}_n (x - w_c) \qquad \forall c \in N_b$$

Insertion and deletion of nodes in the representation layer

After $\lambda \cdot n_N$ steps:

- Determine the quality measure for insertion B^I considering the insertion tolerance ϑ_{ins}:

$$B^I_i := \tau_{Li} - \tau_{\vartheta i} \cdot (1 - \vartheta_{ins}) \qquad \forall i \in G$$

- Find node q and its neighbor f for insertion, if the following criterion is fulfilled:

$$0 < K_{ins,q} := \max_{i \in G}(K_{ins,i}) \; ; \quad B^I_f := \max_{i \in N_q}(B^I_i) \; ; \quad K_{ins,i} := B^I_i - Y_i \qquad \forall i \in G$$

If q and f exist:

- Delete the edge between q and f, insert a new node r, and connect r with q and f. The weights w_r, w_r^{out} and the counters τ_{Sr}, τ_{Lr}, $\tau_{\vartheta r}$ and τ_{Ir} are determined by the arithmetical average of the weights and error counters of q and f.
- If

$$\tau_{Li} > \tau_{Ii} \cdot (1 - \vartheta_{ins}) \qquad \forall i \in \{q, f, r\}$$

the last insertion was not successful. Thus, adapt the moving insertion threshold:

$$\tau_{\vartheta i} := \tau_{\vartheta i} \cdot \eta_\vartheta \cdot (\tau_{Li} - \tau_{\vartheta i} \cdot (1 - \vartheta_{ins})) \qquad \forall i \in \{k \mid \tau_{Lk} > \tau_{Ik} \cdot (1 - \vartheta_{ins}); q, f, r\}$$

- Determine the new inherited error τ_I of q and f:

$$\tau_{Ii} := \tau_{Li} \qquad \forall i \in \{q,f,r\}$$

- Check the deletion criteria considering a minimal age ϑ_{delY} and find node d, whose criterion K is lower than the deletion threshold ϑ_{del}:

$$\vartheta_{del} > K_{del,d} := \min_{i \in G}(K_{del,i}) \quad \wedge \|N_d\| \geq 2 \quad \wedge Y_d < \vartheta_{delY}$$

with

$$K_{del,i} := \frac{\overline{\Delta w_i}}{\bar{l}} \cdot \overline{\Delta w_i^{out}} \qquad \forall i \in G$$

the local similarity of the input weights:

$$\overline{\Delta w_i} := \frac{1}{\|N_i\|} \sum_{j \in N_i} \|w_i - w_j\|$$

the average similarity of the input weights:

$$\bar{l} := \frac{1}{n_N} \sum_{j=1}^{n_N} \overline{\Delta w_j}$$

and the local similarity of the output weights:

$$\overline{\Delta w_i^{out}} := \frac{1}{\|N_i\|} \sum_{j \in N_i} \|w_i^{out} - w_j^{out}\|$$

Adaptation of the output layer

- In case of the error-driven example discussed here, determine the squared error.

$$E_{task}(x) :\equiv E_{\substack{squared \\ error}}(x) := \|\zeta - o\|^2$$

- Determine the local output learning rates from the quality measure B^L, the youth Y, the output adaptation rate η_o and the output adaptation threshold ϑ^o_L:

$$\eta_i^o := \begin{cases} 0 & if \quad \alpha_i^o < 0 \\ \eta_o & if \quad \alpha_i^o > 1 \\ \alpha_i^o \cdot \eta_o & else \end{cases} \qquad \alpha_i^o := \frac{B_i^L}{1 \cdot \vartheta_L^o} \cdot Y_i - 1 \qquad \forall i \in G$$

- Adapt the weights of the nodes j of the output layer:

$$\Delta w_{ji} := \eta_i^o(\zeta_j - o_j)y_i \quad ; \qquad \forall j \in \{1 \ldots m\}, \quad \forall i \in G$$

Adaptation of the counters and edges of nodes in the representation layer

- Adapt the long time error counter τ_L and the short time error counter τ_S for the winner b with the time constant T and the error of the task:

$$\tau_{(L/S)b} := e^{-\frac{1}{T_{(L/S)}}} \cdot \tau_{(L/S)b} \cdot (1 - e^{-\frac{1}{T_{(L/S)}}}) \cdot E_{task}(x)$$

- Decrease the youth Y of the best node b:

$$Y_b := e^{-\frac{1}{T_Y}} \cdot Y_b$$

- Compared to (Hamker and Gross, 1997) an advanced criterion for the decrease of the insertion threshold τ_ϑ is presented. It takes the changes of the errors into account and reduces the insertion threshold only, if the distribution of the data changes:

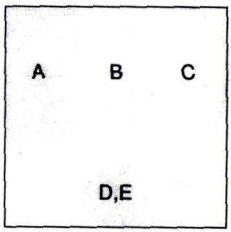
$$\tau_{\vartheta b} := (1 - \Lambda(\alpha_b)) \cdot e^{-\frac{1}{T_\vartheta}} \cdot \tau_{\vartheta b} \qquad \text{if} \quad \Lambda(\alpha_b) > 0$$

$$\alpha_b = \frac{1 \cdot |B_b^L - 1|}{1 \cdot \vartheta_L^i} - 1 \; ; \quad \Lambda(x) = \begin{cases} 0 & \text{if} \quad x < 0 \\ 1 & \text{if} \quad x > 1 \\ x & \text{else} \end{cases}$$

- Adapt the edges as follows (equal to (Fritzke, 1995)):
  - Increase all edges emanating from b by one.
  - Set the age of the edge between b and s to zero. If no edge between b and s exists, create a new one.
  - Remove all edges older than ϑ_{age}.
  - Remove all nodes without an edge.

4 Results

For a demonstration of the above ideas, we previously performed simulations of life-long error-feedback learning on an open data set containing overlaps but without changes in the environment. Results presented in (Hamker, Gross, 1997) showed that the network stabilizes and although due to overlapping classes a permanent error occurs, no further insertion takes place. Furthermore, it was shown that in case of a changing environment, the network structure remains adaptive to insert new nodes and to change the weights.
Here, we will focus on the internal dynamics of the algorithm in a changing environment. Mathematically speaking, a changing environment corresponds to a time-dependent input probability (Heskes and Kappen, 1993). For illustration purposes the 2D artificial data set in Figure 3 is chosen.
Figure 4 shows the behavior of the algorithm. In the first 20000 steps the input contains two awfully overlapping classes which cause a high error (b). Nevertheless after 20000 steps, the algorithm has learned by increasing its insertion threshold (c) that a further insertion does not improve the squared error and stabilizes, as can be seen in (d), and the amount of nodes. Now the environment changes, new errors occur and the algorithm tries to minimize them by changing its weights and inserting new nodes. Although the environment gets much easier, there is still an unsolvable overlapping between the ellipse and the line that would cause a further insertion of nodes. By increasing the insertion



	Class	Region	Environment (probability)			
			1	2	3	4
A	1	Rectangle	1	1	0	1
B	1	Line	1	1	1	0
C	2	Ellipse	0	1	1	1
D	3	Circle	1	0	0	0
E	2	Circle	1	1	1	0

Fig. 3. Changing environment based on five areas (A-E). The environment changes from 1-6 after every 20000 steps. The two regions D and E are completely overlapped and the class 1 of the line has an overlap with class 2 of the ellipse. The used parameters are $\eta_b = 0.1$; $\eta_n = 0.01$; $\eta_o = 0.15$; $\eta_\phi = 0.5$; $T_S = 20$; $T_L = T_Y = T_\phi = 100$; $\lambda = 10$; $\vartheta_{age} = 50$; $\vartheta'_L = 0.05$; $\vartheta^o_L = -0.05$; $\vartheta_{ins} = 0.1$; $\vartheta_{del} = 0.05$; $\vartheta_{delY} = 0.01$.

threshold (c) of the relevant nodes, the algorithm learns to stop insertion in the overlapping areas. At least after 40000 steps it has adopted to the environment that no further learning is needed (d). If the probability changes in some regions to zero, like in the environment from 40000 to 60000 steps, those remaining nodes, often called "dead nodes", play a major role in life-long learning. They are in no way "dead nodes", instead they preserve the knowledge of previous situations for future decisions. If the old prototype patterns were removed, the knowledge would be lost and the same, already learned situations will again cause errors. Due to a further insertion at the overlapping, still a bit learning takes place (d). In the environment from 60000 to 80000 steps, most of the neurons remain at their positions. Since the environment shows no overlappings the error dereases to zero.

Sumarizing, the algorithm is able to cope with all life-long learning scenarios, like overlaps, never seen inputs and temporarily not appearing inputs.

5 Conclusion

A life-long learning incremental neural network was presented to coordinate insertion and learning. On an abstract level, it demonstrates a biologically feasible selective modification of plasticity induced by a "global teacher" signal.

The experiments show that the network can learn to stop insertion in regions where the error can not be decreased. Furthermore, in changing environments the network remains stable for old prototype patterns and adaptive for new or different patterns. The neural network neither freezes by any decaying parameters nor switches between different learning modes, instead it is able to learn continuously by evaluating its own insertions.

The results obtained indicate a good performance and are a promising step towards life-long learning in neural networks. A performance evaluation on real data shows (Hamker, 1998).

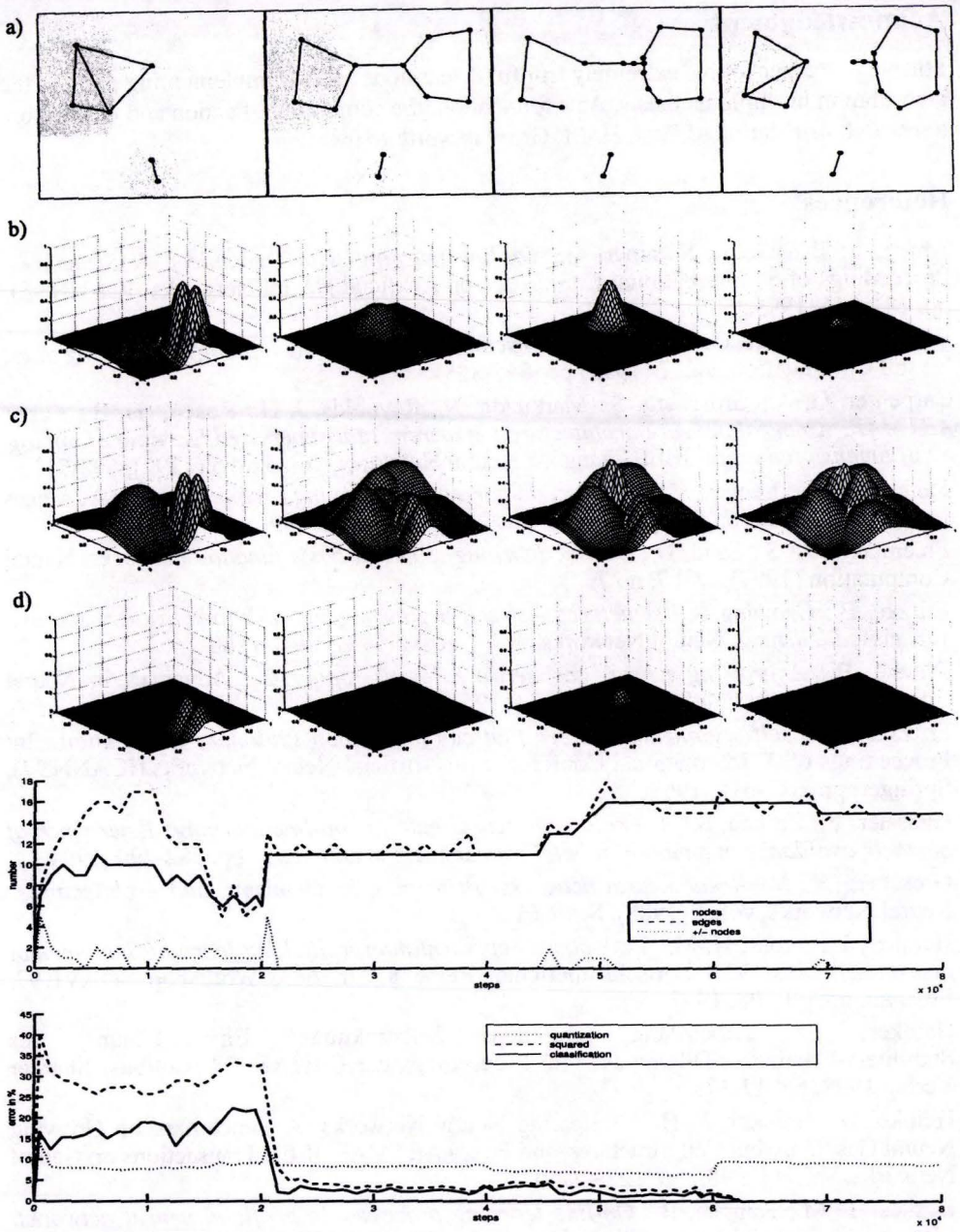


Fig 4. From left to right: internal parameters of every node before changing the environment (20000 steps). From the top to the bottom the: **a)** input weights, **b)** longterm error τ_L **c)** insertion threshold τ_θ **d)** learning parameter α^i, the amount of nodes, and the errors are shown.

Acknowledgment

I thank T. Vesper for his extremely fruitful discussions and for implementing parts of the algorithm in his diploma thesis. As a foundation, the combining of action and perception, a research orientation of Prof. H.-M. Gross is worth to mention.

References

Ahrns, I.; Bruske, J.; Sommer, G.: *On-line learning with Dynamic Cell Structures.* Proceedings of 5th International Conference on Artificial Neural Networks (ICANN'95), pp. 141-146, 1995.

Bruske, J.; Sommer, G.: *Dynamic cell structure learns perfectly topology preserving map.* Neural Computation, vol. 7 (1995) pp. 845-865.

Carpenter, G. A.; Grossberg, S.; Markuzon, N.; Reynolds, J. H.; Rosen, D. B.: *Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps.* IEEE Trans. on Neural Networks, vol. 3 no 5 (1992), 698-713.

Fahlman, S. E.; Lebiere, C.: *The cascade-correlation learning architecture.* In: Advances in Neural Information Processing Systems 2, pp. 524-532, 1989.

Freeman, J. A. S.; Saad, D.: *On-line learning in radial basis function networks.* Neural Computation (1997), vol 9, no 7.

Fritzke, B.: *Growing cell structures – A self-organizing network for unsupervised and supervised learning.* Neural Networks, vol. 7 no 9 (1994), 1441-1460.

Fritzke, B.: *A growing neural gas network learns topologies.* Advances in Neural Information Processing Systems, vol. 7 (1995).

Fritzke, B.: *A self-organizing network that can follow non-stationary distributions.* In: Proceedings of 7th International Conference on Artificial Neural Networks (ICANN'97), Springer, pp. 613-618, 1997.

Gaussier, P.; Zrehen, S.: *A topological neural map for on-line learning: Emergence of obstacle avoidance in a mobile robot.* From animals to animats 3, pp. 282-290, 1994.

Grossberg, S.: *Nonlinear neural networks: Principles, Mechanisms, and Architectures.* Neural Networks, vol. 1 (1988), S. 17-61.

Hamker, F.; Gross, H.-M.: *Task-based representation in lifelong learning incremental neural networks.* VDI Fortschrittberichte, Reihe 8, Nr. 663, Workshop SOAVE'97, Ilmenau, pp. 99-108, 1997.

Hamker, F.: Lebenslang lernfähige Zellstrukturen: Eine Lösung des Stabilitäts-Plastizitäts-Dilemmas? In: Proceedings der CoWAN '98, Cottbus: Sharker Verlag 1998, pp. 17-37.

Heinke, D.; Hamker, F. H.: Comparing Neural Networks: A Benchmark on Growing Neural Gas, Growing Cell Structures, and Fuzzy ARTMAP. IEEE Transactions on Neural Networks, vol. 9 (1998), pp. 1279-1291.

Heskes, T. M.; Kappen, B.: *On-line learning processes in artificial neural networks.* Mathematical Foundations of Neural Networks. Elsevier, pp. 199-233, 1993.

Rescorla, R. A.; Wagner, A. R.: *A theory of Pavlovian conditioning; variations in the effectiveness of reinforcement and nonreinforcement.* Classical Conditioning 2, Current Theory and Research. A. H. Black and W. H. Prokasy (Eds.), New York: Appleton-Century-Crofts, pp. 64-99, 1972.