Adaptation of Kohonen Feature Map Topologies by Genetic Algorithms

Daniel Polani and Thomas Uthmann[a]

[a]Institut für Informatik · FB Mathematik, Johannes Gutenberg-Universität Mainz,
Staudingerweg 9, D-6500 Mainz 1, Germany,
polani@informatik.mathematik.uni-mainz.de,
uthmann@informatik.mathematik.uni-mainz.de

Abstract

The following paper presents simulational results of coupling Genetic Algorithms to the Kohonen Feature Map paradigm. The Genetic Algorithm is used to improve the Kohonen Net topology, thus yielding better adaptation to the input vector space $[0,1]^2$. Different parameters of the process and their influence as to the resulting topologies are discussed.

1. Introduction

Recently there has been an increased interest in the application of Genetic Algorithms (GA) to (artificial) Neural Networks (NN). The appeal of combining the GA and the NN paradigm arises from the expectation that GA might yield a systematic approach for an efficient search for "optimal" nets in the large space of network structures and/or dynamics. A second motivation, of course, are the biological roots which both paradigms share.

One main approach to combine GAs and NNs has been the usage of GAs instead (or in addition to) Backpropagation or related algorithms to train the weights of Feed-Forward-Networks [7]. GAs have also been used to find optimal learning rules [3] or to search for an efficient net structure and topology ([2], [5]). It has been felt that a direct genotype-phenotype-coupling in the combined algorithm (e.g. when explicit coding the weights of a net into the chromosomes) is inappropriate and may lead to an unnecessarily large GA search space, discarding its possible reduction by problem-inherent redundancies [1]. Even in biological systems one can hardly expect to find an explicit detailed coding of structure or weights of the nervous system.

Kohonen Feature Maps appear to be suited to study the possibility of compact encoding of a net topology due to the rather uniform structure of the nets and of the spaces they adapt to. In the classical approach the net topology is defined following a priori assumptions about structure and dimensionality of the input vector space. The Kohonen training procedure is then applied to establish a "topological mapping" between neurons and input vector space. Using the GA to optimize the Kohonen net topology itself there is no need to impose an a priori knowledge on the net any more. Instead, the net topology is adapted to the input space to improve the effectivity of the training procedure.

Figure 1. Interaction between the GA and the Kohonen Feature Map

2. Interaction between the GA and the Kohonen Feature Map

Fig. 1 shows the structure of the algorithm. The GA creates by mutation and recombination chromosomes, which are elements of $\{0,1\}^q$. Every such genotype then defines the topology of a Kohonen Net via a transcription rule (which will be described in more detail below); the net is trained and after completion of the training it is subjected to a "quality test", which serves as a fitness function to be employed by the GA.

2.1. The Kohonen Feature Map

We will regard the Kohonen Feature Map as an undirected graph G, where a weight vector $\in [0,1]^m = I^m$ is associated with every vertex. We have a metric d_I defined on I^m (typically the Euclidean distance) and the canonical metric d_G defined on the set of vertices of G (two vertices v and v' have distance 1 iff they are connected by an edge).

The training procedure for the net has been derived from the standard one (s. [4], [6]). In every training step first a vector $x \in I^m$ is presented (e.g. according to a probability distribution). Then the vertex v^* is chosen, whose weight vector w^* is closest to x with respect to the metric d_I on I^m. Then for every vertex of the graph the weight vectors are corrected according to:

$$w_v(t+1) = w_v(t) + \alpha(t) \cdot h_t(d_G(v^*, v)) \cdot [x(t) - w_v(t)] \tag{1}$$

where the correction depends on the learning rate $\alpha(t)$ and on the activation profile h_t. The latter determines how strong the weights of a vertex v are corrected depending on

its graph-immanent distance $d_G(v^*, v)$ from the "maximum activation vertex" v^*. This more general formulation of the Kohonen learning prescription renders possible to apply the training rule to a net of arbitrary topology.

2.2. Genotype-Phenotype Mapping

In the next step we will define the transcription rule used in our simulations. This rule yields a net with a unique topology for every given chromosome. It requires the chromosome to consist of 2 bytes + n double-bytes (1 double-byte = 2 bytes = 16 bits):

The number k of vertices in the graph is given by the content of the first byte plus 1, guaranteeing the net to have at least one vertex. The second one, multiplied by 16, yields the maximum number of transcription steps that may be done before the procedure has to stop (`maxsteps`). These two "header bytes" are followed by n double-bytes, each of which defines one transcription step. In every transcription step a vertex is connected with a different one. Whenever the rule tries to establish a connection which already exists or tries to connect a vertex with itself, the algorithm stops. The central part of the transcription algorithm follows (an example showing some transcription steps is illustrated in Fig. 2):

```
/* transcription; k,n and maxsteps as above */

from    :=  0
to      :=  0
step    :=  0
i       :=  0
while step < maxsteps do    /* transcription step */
    from   :=  (to   + a[i]) mod k
    to     :=  (from + b[i]) mod k
    if (from = to) or (vertex[from] is connected with vertex[to])
        then exit while loop
    connect vertex[from] with vertex[to]
    step := step + 1
    i := (i + 1) mod n
endwhile
```

Given the topology as resulting by the transcription rule, the net is trained according to (1) until a termination condition is satisfied.

The trained net can then be regarded as phenotypic expression of the chromosomes. Thus the genotype-phenotype-mapping consists of two phases: in the first the transcription rule is applied to the chromosome. This deterministic process "unfolds" the compact information stored in the chromosome into a complete network topology (since usually the explicit information contained in a chromosome is much smaller than that of a graph, this **is** "unfolding" in most cases). In the second the net's weight vectors are "unfolded"

Figure 2. Example of a 2 bytes + 2 double-bytes chromosome (in general a chromosome may contain more double-bytes)

in weight vector space. Since the training data in most cases result from a random distribution of event vectors x this part of the genotype-phenotype-mapping is no longer deterministic. Thus the full phenotypic expression is a result of the combination of genetic and "environmental" factors.

One should note here that `maxsteps` may, but need not influence the net topology, only setting an upper limit for the connectivity of the net. Depending on the number of vertices k and the content of the double-bytes in some cases the algorithm may as well create a net with a large set of sparsely or unconnected vertices without having been limited by `maxsteps`.

2.3. Fitness function

The fully developed phenotype has now to be subjected to a quality or fitness function, which is going to be used by the GA. Being acquainted with a typical appearance of trained Kohonen nets one would choose some kind of smoothness function which would yield higher quality values for smoother nets. For the sake of simplicity we preferred to choose a different quality function, which gives an estimate of the average distance from an arbitrary $x \in I^m$ to the nearest weight vector of one of the vertices of G. Given a sample of q points $x_i \in I^m, i = 1, \ldots, q$ the quality function used is given by:

$$Q(G) = \frac{1}{\sum_{i=1}^{q} (w^*(x_i) - x_i)^2}$$

where $w^*(x_i) := w_{v^*}$, if for all vertices v: $d_I(x_i, w_{v^*}) \leq d_I(x_i, w_v)$ holds. We call v^* the vertex activated by the sample vector x_i; v^* and hence w^* is almost always unique. $Q(G)$ is essentially a measure of the average square distance from an input vector to the vertex it activates; a smaller average distance yields a higher quality function and indicates a better adaptation to the input space.

For a smaller estimated average distance the quality function increases. It is not obvious that this quality function that essentially measures the equidistribution of the vector weights of the trained nets should prefer nontrivial net topologies to a set of unconnected vertices. But when starting a training sequence with weight vectors only distributed inside a small area (in our case e.g. $[0.45, 0.55]^2$) only part of the neuron weights are spread throughout weight vector space when training a net with unconnected vertices (Fig. 3.a). In systems with a higher degree of connectivity the vertices mutually help each other to spread throughout weight space towards a more uniform distribution.

3. Experiments and Results

In our simulations we used a GA with roulette wheel selection using the fitness function described above. Recombination took place with two randomly chosen crossover points. The population consisted of 30 individuals (chromosomes). It was initialized with random binary strings. The best chromosome of the current population was always retained and never subjected to mutation or substitution by offspring. To compute the fitness function the chromosome was transcribed according to 2.2 into a net topology: the net weights were randomly distributed in $[0.45, 0.55]^2$ (Fig. 3.c). The Kohonen Net training procedure with a random input vector sequence was then applied. Finally the quality function has been computed for a sample of 100 test vectors. For the sake of reproducibility the random vector sequences used were the same for every chromosome, so the genotype-phenotype-mapping was in fact deterministic instead of, as it should be in principle, being stochastic: so the same chromosome in our experiments would always yield the same trained net. The sample used to calculate the quality function was also random but fixed and is shown in Fig. 3.b. One more remark as to the training of the nets: α was chosen to be $\alpha(t) = 0.25/(1 + 0.001\,t)$. h_t was chosen following the standard suggestion. It had the value 1 up to a certain distance from the maximum activation vertex and the value 0 farther away; up to 1000 training steps this distance was 3, up to 2000 the distance was 2 and afterwards 1, defining decreasing influence ranges of the maximum activation vertices.

The characteristic phases in the development of a fully connected net created by the above transcription rule (applying it to an arbitrarily chosen 2 + 2 double-byte chromosome) are shown in Fig. 3.c - 3.f. The training starts with the initial distribution of weights shown in Fig. 3.c. Up to 1000 steps the coarse structures are formed (Fig. 3.d). When the range of the activation profile decreases at later training times the finer structures begin to develop (Fig. 3.e). After 3000 steps (Fig. 3.f) no substantial changes take place any more.

In Fig. 3.g - 3.i the results of a GA run are illustrated, where the nets are trained only for 300 steps; the figures always present the fittest phenotype of the population. Till later generations the phenotypes exhibit a strongly irregular embedding into I^2 and a not fully

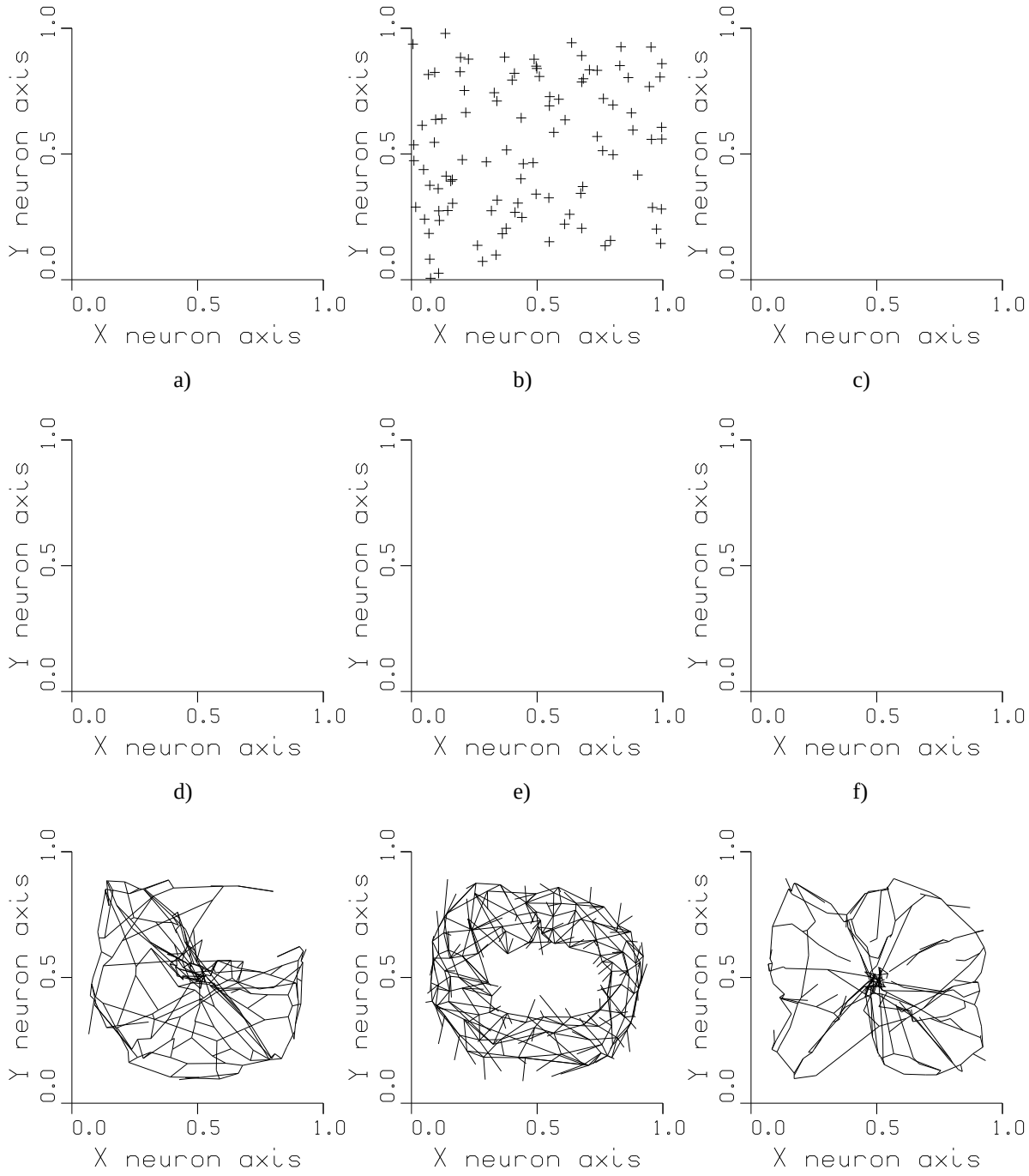

Figure 3. Diagrams
a) Net of unconnected vertices after 2000 steps with quality 1.26. b) Standard sample (100 training vectors) c)-f) Training sequence of a typical net with qualities 0.08, 1.06, 1.97, 3.56 after 0, 1000, 2000 and 3000 cycles g)-i) fittest phenotype of population at generation 0, 100, 500 trained for 300 cycles (qual. 1.26, 1.59, 2.08)

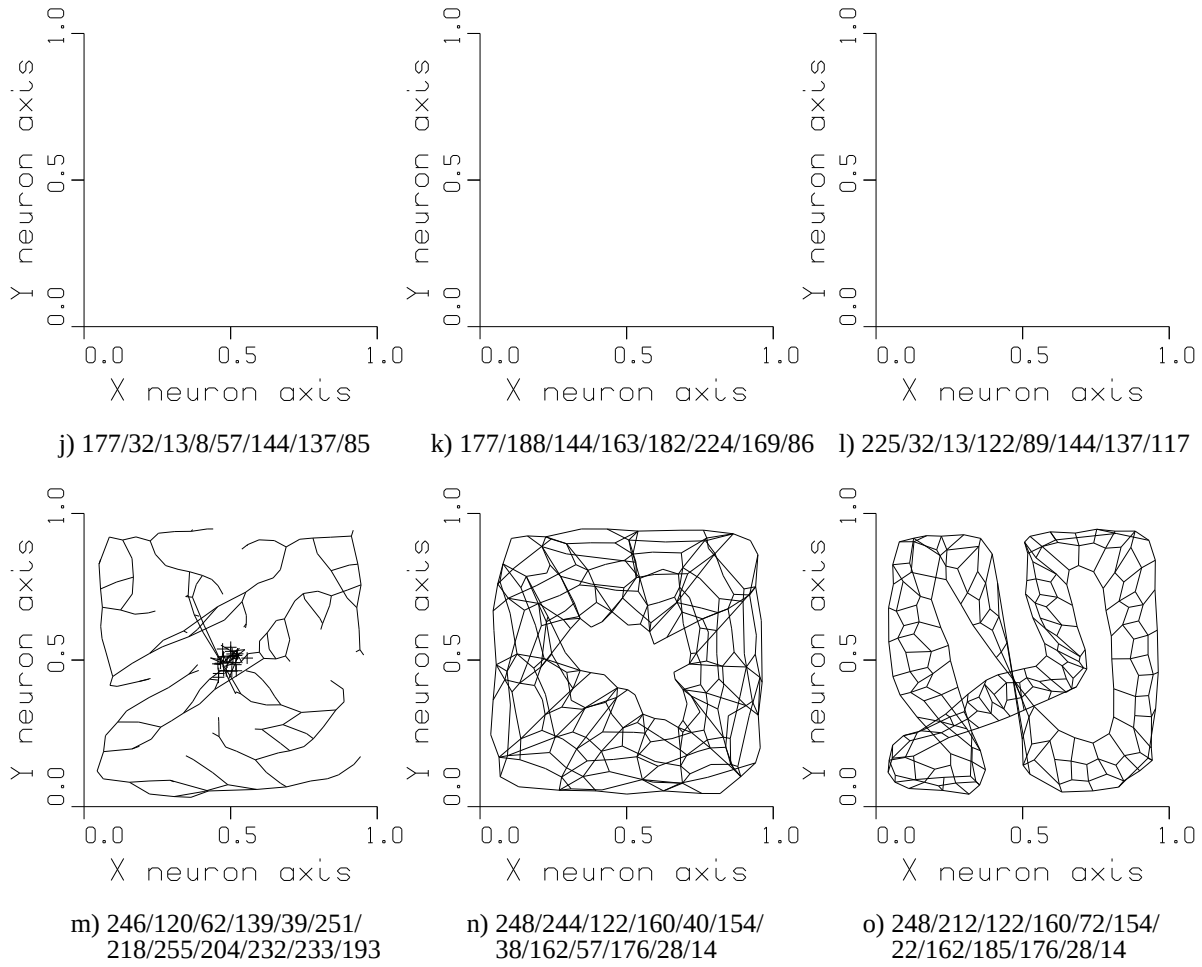

Figure 3. Diagrams (continued)
j)-l) fittest phenotype at generation 0, 20, 50 trained for 3000 cycles (qual 5.88, 6.35, 6.69) m)-o) fittest phenotype at generation 2, 150, 500 with longer genotypes and smaller `maxsteps` (qual. 4.98, 8.26, 9.04)

developed connectivity.

One should note, however, that the transcription rule yields a relative regular net topology for shorter chromosomes, as long as the transcription process is not limited by `maxsteps`. Neglecting the influence of `maxsteps` the irregularity of the trained phenotype is chiefly due to the weights and not to the topology of the net.

Furthermore the nets resulting from the evolution with short training times exhibit effects like "dangling bonds", which are not caused only by the `maxsteps`-limitation, and which can be seen e.g. in Fig. 3.h.

On the other hand, what can be expected to be achieved by increasing training times? It is shown in Fig. 3.j - 3.l that as early as in the initial generation with a population of random chromosomes phenotypes with a more regular structure and a stronger connectivity (no "dangling bonds") are preferred (see especially Fig. 3.k, "cobra"-pattern). A

stronger connectivity and the absence of dangling bonds appear to be relevant to achieve a higher regularity for the weight vector distribution of the trained phenotype, which, again, seems to yield better adaptation results. Even longer training times applied to the nets of Fig. 3.g - 3.i did neither result in essentially higher regularity of the phenotype (not shown here) nor in in quality function values as high as for the Fig. 3.j - 3.l.

To check the connectivity issue once more and to further explore the genotype-phenotype-relation we present one of a series with longer chromosomes and where the second byte contains only `maxsteps`/2 (i.e. `maxsteps` is only 2 times the contents of the second byte, Fig 3.m - 3.o). The longer genotypes give rise to an increased complexity of the nets. In spite of that and even though `maxsteps` is much more restrictive than in the runs Fig. 3.g - 3.l one notes that the successful nets tend to exhaust the connectivity reservoir by a larger `maxsteps`.

4. Conclusions

It has been demonstrated that a coupling of the GA and the Kohonen Feature Map paradigm leads to reasonable results regarding the capability of adaptation to a given event space. In particular Kohonen nets proved to be promising candidates for phenotype-genotype-mapping by a fairly simple transcription rule due to their highly uniform structure.

The results show that longer training times which permit full development of a net are much more effective yielding good adaptational results than short training times (even at longer evolution periods) which only result in immature nets. It is remarkable that the advantage of the nets with longer training times immediately strikes one's eyes. Higher connectivity seems to be favourable for a more efficient exploitation of the input space.

Usage of a different fitness function as well as different parameters for the GA and the Kohonen Net training has been envisaged. A deeper analysis of the current transcription rule and of possible generalizations will be the topics of further research.

1 Richard K. Belew, John McInerney, and Nicol N. Schraudolph. Evolving networks: Using the genetic algorithm with connectionist learning. Technical Report CS 90-174, University of California, San Diego, 1990.

2 Stefan Bornholdt and Dirk Graudenz. General asymmetric neural networks and structure design by genetic algorithms. Technical Report 91-046, Deutsches Elektronen-Synchrotron, Notkestraße 85, D-2000 Hamburg 52, May 1991.

3 David J. Chalmers. The evolution of learning: an experiment in genetic connectionism. In D.S. Touretzky, J.L. Elman, T.J. Sejnowski, and G.E. Hinton, editors, *Proceedings of the 1990 Connectionist Models Summer School*, San Mateo, California, 1990. Morgan Kaufmann.

4 T. Kohonen. *Self-Organization and Associative Memory*, volume 8 of *Springer Series in Information Sciences*. Springer-Verlag, Berlin, Heidelberg, New York, third edition, May 1989.

5 Geoffrey F. Miller, Peter M. Todd, and Shailesh U. Hedge. Designing neural networks using genetic algorithms. In David Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 379–384, San Mateo, California, 1989.

6 Helge Ritter, Thomas Martinetz, and Klaus Schulten. *Neuronale Netze*. Addison-Wesley, 1990.

7 Darrell Whitley and Thomas Hanson. Optimizing neural networks using faster, more accurate genetic search. In David Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 391–396, San Mateo, California, 1989.