Artificial Neural Networks
T. Kohonen, K. Mäkisara, O. Simula and J. Kangas (Editors)

A "Neural-Gas" Network Learns Topologies

Thomas MARTINETZ and Klaus SCHULTEN
Beckman-Institute and Department of Physics
University of Illinois at Urbana-Champaign
405 North Mathews Ave., Urbana, IL 61801, USA

A neural network algorithm for vector quantization of topologically arbitrarily structured manifolds of input signals is presented and applied to a data manifold M which consists of subsets of different dimensionalities. In addition to the quantization of M each neural unit $i, i = 1, ..., N$ of the network A develops connections, described by $C_{ij} \in \{0,1\}$, to those neural units j with adjacent receptive fields. The resulting connectivity matrix C_{ij} describes asymptotically the neighborhood relationships among the input data of the quantized manifold and defines a graph which reflects the often a priori unknown dimensionality and topological structure of the data manifold M.

1. Introduction

Vector quantization as a technique for data compression is widely used in technical applications [1-5] and is also assumed to be an important principle employed by biological information processing systems. Several approaches based on neural network models have been proposed which are capable of adaptively quantizing a given set of input data [6-11].

Vector quantization techniques encode a manifold of data, e.g., a submanifold $M \subseteq R^n$, by employing a finite set A of reference (or "codebook") vectors $\mathbf{w}_i \in R^n, i = 1, ..., N$. A data vector $\mathbf{v} \in M$ is described by the best-matching or "winning" reference vector $\mathbf{w}_j$ of A, for which a distortion measure $d(\mathbf{v}, \mathbf{w}_j)$, e.g., the square error $\|\mathbf{v} - \mathbf{w}_j\|^2$, is minimal. This procedure divides the manifold M in a number of subregions

$$M_j = \{\mathbf{v} \in M \,\big|\, \|\mathbf{v} - \mathbf{w}_j\| \leq \|\mathbf{v} - \mathbf{w}_i\| \; \forall i\},$$

so-called Voronoi polygons, out of which each data vector $\mathbf{v}$ is now described by the corresponding reference vector $\mathbf{w}_j$. If the probability distribution of data vectors over the manifold M can be described by $P(\mathbf{v})$, the average reconstruction error is determined by $\int d\mathbf{v} P(\mathbf{v})\, d(\mathbf{v}, \mathbf{w}_{i(\mathbf{v})})$ and has to be minimized by an optimal choice of the reference vectors $\mathbf{w}_i$. In the neural network models for vector quantization the adaptation process for the synaptic weights $\mathbf{w}_i$ of the neural units can be interpreted as the construction of reference vectors. The synaptic weights $\mathbf{w}_i$ determine the input stimuli neural unit i is tuned to by defining a receptive field which corresponds to the Voronoi polygon determined by $\mathbf{w}_i$. Each time an input stimulus is presented, the best-matching element j of the array A of neural units with $\|\mathbf{v} - \mathbf{w}_j\| = \min_{i \in A} \|\mathbf{v} - \mathbf{w}_i\|$ represents the input.

A neural network model widely used for vector quantization is Kohonen's self-organizing feature map [10, 11]. In Kohonen's model each neural unit has certain neighborhood relations to all the other neural units, defining a topology on the array A of formal neurons. In addition to the quantization of the manifold M of data vectors Kohonen's self-organizing feature map yields a topographic mapping from M to the array A. Neural units which are neighbors within the given topology of the network modify their synaptic weights in a concerted way, which adapts their $\mathbf{w}_i$ to values which are close on the manifold of input data. This topology conserving mapping captures the similarity relationships among the represented input data by providing additional information about the neighborhood relations between the reference vectors $\mathbf{w}_i$. This additional information is highly desirable in several applications, including path planning and obstacle avoidance [12], visuomotor control [13, 14], speech processing [1], and might also play an important role in tasks involving more abstract "semantic" features pertaining to language and higher processing levels [15]. To obtain optimal results concerning the conservation of the topology of the mapping as well as the optimal utilization of all neural units, the topology of the employed network has to match the topology of the manifold of data which is represented. This requires prior knowledge about the topological structure of the manifold M, which is not always available or might be difficult to obtain if the topological structure of M is very heterogeneous, e.g., composed of subsets of different effective dimensions or disjunct and highly fractured.

For this reason, it is desirable to employ a more flexible network capable of (i) quantizing topologically heterogeneously structured manifolds and (ii) learning the similarity relationships among the input signals without the necessity of prespecifying a network topology. One approach, suggested by Kohonen and his co-workers, is to readjust the topology of the employed network at different stages during the learning procedure, based on the minimal spanning tree between the reference vectors $\mathbf{w}_i$ [16].

2. The "Neural-Gas" Network

In the approach we present here the synaptic weights $\mathbf{w}_i$ are adapted independently of any topological arrangement of the neural units within the neural net. Instead, the adaptation steps are affected by the topological arrangement of the receptive fields within the input space. Since the synaptic weight changes $\Delta\mathbf{w}_i$ are not determined by the arrangement of the neural units within a topologically prestructured lattice, but by the relative distances between the neural units within the input space, we chose the name "neural-gas" network.

Information about the arrangement of the receptive fields within the input space is implicitly given by the set of distortions $D_\mathbf{v} = \{\|\mathbf{v}-\mathbf{w}_i\|, i = 1, ..., N\}$ associated with each $\mathbf{v}$. Each time an input signal $\mathbf{v}$ is presented, the *ordering* of the elements of the set $D_\mathbf{v}$ determines the adjustment of the synaptic weights $\mathbf{w}_i$. The resulting adaptation rule can be described as a "winner-take-most" instead of a "winner-take-all" rule. Simultaneously, neural units i, j with receptive fields M_i, M_j adjacent on the manifold M develop connections between each other, which is described by setting the matrix element C_{ij} from zero to one. The resulting connectivity matrix C_{ij} at the end of the learning procedure represents the similarity, i.e., the neighborhood relationships, among the input data.

A presented input signal $\mathbf{v}$ is received by each neural unit i and induces "excitations" $f_i(D_\mathbf{v})$ which depend on the set of distortions $D_\mathbf{v}$. Assuming a Hebb-like rule with an additional memory decay term, the coincidence of presynaptic input $\mathbf{v}$ and postsynaptic excitation f_i changes $\mathbf{w}_i$ by

$$\Delta\mathbf{w}_i = \epsilon \cdot f_i(D_\mathbf{v}) \cdot (\mathbf{v} - \mathbf{w}_i). \tag{1}$$

The step size $\epsilon \in [0,1]$ describes the overall extend of the modification and $f_i(D_\mathbf{v}) \in [0,1]$ accounts for the topological arrangement of the $\mathbf{w}_i$ within the input space. The excitation $f_i(D_\mathbf{v})$ depends on the number k of neural units which are excited more strongly than neural unit i itself and is largest for the neural unit $i = i_0$ for which

$$\|\mathbf{v} - \mathbf{w}_{i_0}\| = \min_{j\in A} \|\mathbf{v} - \mathbf{w}_j\|.$$

By i_1 we denote the neural unit with its $\mathbf{w}_{i_1}$ second closest to $\mathbf{v}$, and $i_k, k = 0, ..., N-1$ is the neural unit for which there are k units j of the array A with $\|\mathbf{v}-\mathbf{w}_j\| < \|\mathbf{v}-\mathbf{w}_{i_k}\|$. If we denote

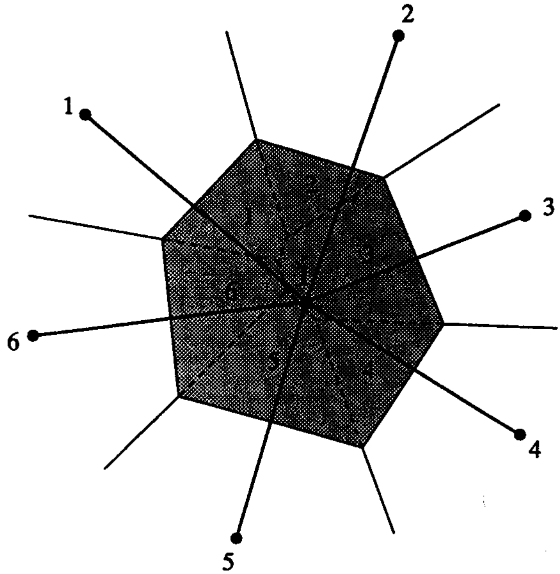

Fig.1: Illustration of the update rules for the connections of the network. Each time an input signal is presented within the shaded area, which depicts the receptive field (Voronoi polygon) M_i of neural unit i, a connection from i to the neural unit which is second closest to the input signal is established. The numbers $1, ..., 6$ denote the subregions within the shaded area for which the correspondingly numbered neighboring neural units are second closest to an input signal $\mathbf{v} \in M_i$. Neural unit i only develops connections to neural units the receptive fields of which share common borders with its own receptive field.

the number k associated with each neural unit i by k_i, then $f_i(D_\mathbf{v})$ is given by a function $f(k_i)$ which is unity for $k_i = 0$ and decays to zero with increasing k_i. In the simulations described below we chose $f(k_i) = e^{-k_i/\lambda}$ with λ determining the number of neural units significantly changing their synaptic weights with each adaptation step. Then, the adaptation step (1) for the $\mathbf{w}_i$ is determined by

$$\mathbf{w}_i^{new} = \mathbf{w}_i^{old} + \epsilon \cdot e^{-k_i/\lambda} (\mathbf{v} - \mathbf{w}_i^{old}). \tag{2}$$

To capture the neighborhood relationships between the reference vectors $\mathbf{w}_i$ each time an input stimulus is presented we establish a connection between the neural unit i_0, which had its $\mathbf{w}_i$ closest to $\mathbf{v}$, and the neural unit i_1, which had its $\mathbf{w}_i$ second closest to the input signal. The creation of this connection is described by setting the matrix element $C_{i_0 i_1}$ from zero to one.

Each connection $i - j$ has an "age" t_{ij}. When the age of connection $i - j$ exceeds a prespecified lifetime T, it is removed, i.e., C_{ij} is set to zero. If the connection $i_0 - i_1$ the algorithm tries to establish exists already, its lifetime $t_{i_0 i_1}$ is set to zero and the connection "ages" again. There are two ways to update the age of the connections. One way would be to set the age of each connection equal to the number of learning steps passed since it was established. This requires a synchronous update of the age of all connections with each adaptation step. The second and less time consuming way is to increase with each incoming input stimulus only the age of the connections of neural unit i_0 which was closest to the input stimulus. Since both ways are equivalent on average as long as the probability of a neural unit i to be the "winner" is equal for each i of the network A, we chose the asynchronous and faster update rule for the presented algorithm.

In Fig.1 we show schematically which of the neural units are connected by the introduced adaptation rule. The neural unit denoted by i is the "winner" for input signals presented within the shaded area, the receptive field or Voronoi polygon of neuron i. The numbers $1, ..., 6$ denote the neural units which are second closest to input signals appearing within the correspondingly numbered subregions of the grey shaped area. Only to the neural units $1, ..., 6$ the receptive fields of which share common borders with its own receptive field neural unit i develops connections.

The "neural-gas" algorithm for vector quantization and learning of topological relations can now be summarized by the following steps:

0. Assign initial values to the weights $\mathbf{w}_i \in R^n$ and set all C_{ij} to zero.
1. Select an input vector $\mathbf{v}$ of the input manifold M.
2. For each unit i determine the number k_i of neural units j with

$$\|\mathbf{v}-\mathbf{w}_j\| < \|\mathbf{v}-\mathbf{w}_i\|$$

by, e.g., determining the sequence $(i_0, i_1, ..., i_{N-1})$ of neural units with

$$\|\mathbf{v}-\mathbf{w}_{i_0}\| < \|\mathbf{v}-\mathbf{w}_{i_1}\| < ... < \|\mathbf{v}-\mathbf{w}_{i_{N-1}}\|.$$

3. Perform an adaptation step for the weights according to

$$\mathbf{w}_i^{new} = \mathbf{w}_i^{old} + \epsilon \cdot e^{-k_i/\lambda}(\mathbf{v}-\mathbf{w}_i^{old}), \quad i = 1, ..., N.$$

4. If $C_{i_0 i_1} = 0$, set $C_{i_0 i_1} = 1$ and $t_{i_0 i_1} = 0$. If $C_{i_0 i_1} = 1$, set $t_{i_0 i_1} = 0$.
5. Increase the age of all connections of i_0 by setting $t_{i_0 j} = t_{i_0 j} + 1$ for all j with $C_{i_0 j} = 1$.
6. Remove all connections of i_0 which exceeded their lifetime by setting $C_{i_0 j} = 0$ for all j with $C_{i_0 j} = 1$ and $t_{i_0 j} > T$. Continue with 1.

Step 2 of determining k_i for each neural unit i is the most time consuming part of this algorithm and, on a sequential computer, corresponds to sorting the distances $\|\mathbf{v}-\mathbf{w}_i\|$ which goes with $N \log_2 N$. However, the synaptic change $\Delta\mathbf{w}_i$ for neural units with their $k_i >> \lambda$ is neglectable, which for small λ allows us to "cut off" the sorting at a k with $k << N$. In a parallelized form the computational time of the algorithm only increases with $\log_2 N$. Note, that each neural unit only has to know how many but not which of the other neural units are more closely tuned to the input signal.

3. Result of a Simulation and Discussion

In Fig.2, we show the result of a computer simulation of the described algorithm. The neural net consisted of an array of $N = 200$ units. The manifold M from which the input vectors were randomly chosen consisted of a combination of a three-dimensional, a two-dimensional, and a one-dimensional submanifold. This topological structure of M was chosen to test the capability of the "neural-gas" algorithm to map the array A of neural units onto manifolds with heterogeneous and therefore hardly prespecifiable topologies.

We show the initial state, the result after 5 000, after 10 000, after 15 000, after 25 000, and the final state after 40 000 adaptation steps. The initial values for the synaptic weights $\mathbf{w}_i$ were chosen randomly from a right parallelepiped in which the manifold M was embedded (top left). The "snapshot" after 5 000 adaptation steps (center left) shows the neural net during the first stage of the simulation. Neural units the receptive fields of which were adjacent initially and therefore became connected still might drastically change there assigned locations within the input space and might drift apart with further adaptation steps. The corresponding connections then are no longer refreshed and are removed when their age exceeds their lifetime T.

After 10 000 adaptation steps (bottom left), a few connections between neural units with non-neighboring receptive fields still exist. With further adjustment steps (top right, center right) the neural network adapts more and more to the underlying data manifold M, and at the end of the learning procedure after 40 000 adaptation steps (bottom right), only neural units with adjacent

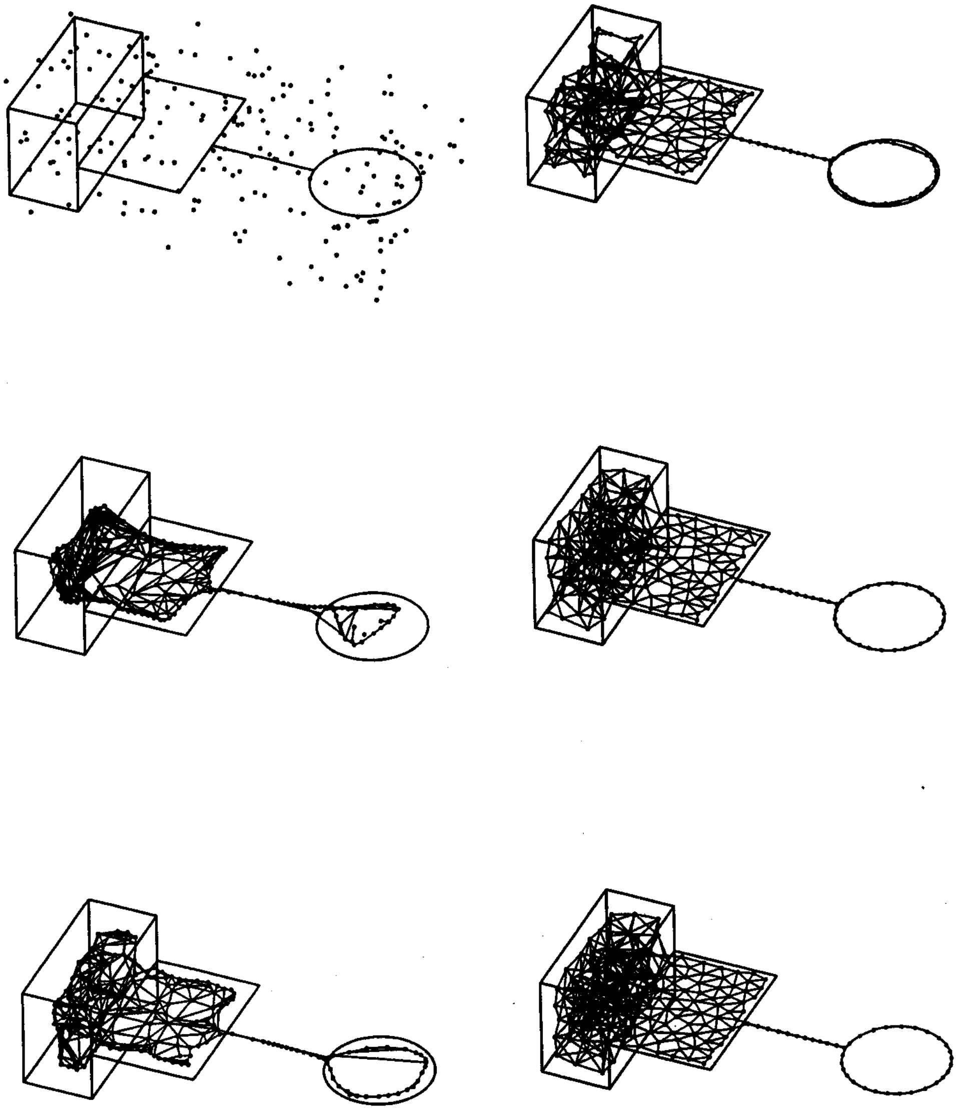

Fig.2: The "neural gas" network quantizing a topologically heterogeneously structured input data manifold. The data manifold consists of a three-dimensional (right parallelepiped), a two-dimensional (rectangle), and a one-dimensional (circle and connecting line) subset. The dots mark the centers of the receptive fields M_i determined by the formal synaptic weights $\mathbf{w}_i$. Connections between neural units i, j, i.e., $C_{ij} = 1$, are indicated by connecting lines between the locations $\mathbf{w}_i$ and $\mathbf{w}_j$. Depicted are the initial state, the network after 5 000, 10 000, 15 000, 25 000, and at the final state after 40 000 adaptation steps (from top left to bottom right) At the end of the adaptation procedure the connections between the neural units reflect the topological structure and the corresponding dimensionality of the data manifold.

receptive fields established and maintained their connections. Within the one-dimensional subsets of M each neural unit developed two connections, except the unit which accounts for the bifurcation at the location where the line encounters the circle. This unit developed three connections. Within the two-dimensional area each neural unit on average established about six (Delaunay triangulation), and within the three-dimensional region each unit tended to develop fourteen connections, corresponding to the average number of neural units with neighboring receptive fields.

As we see in Fig.2, the "neural-gas" algorithm quantized the manifold M by distributing the receptive fields of all the 200 neural units homogeneously over the relevant parts of the input space. At the end of the learning procedure, the graph determined by the connections C_{ij} matches the topology of M and can be regarded as a map which describes the topological structure of the represented manifold. This was achieved without the use of any prior knowledge about the topological structure of M. For a fixed dimensionality, each neural unit gains asymptotically the same number of connections, independent of the number N of neural units employed in the network. Hence, the number of non-vanishing elements of the connectivity matrix C_{ij} and, therefore, the amount of memory necessary for running the "neural-gas" algorithm increases linear with the number of neural units.

In the described simulation the parameter λ, the step size ϵ, and the lifetime T were dependent on the number of already performed adaptation steps t. This time dependence had the same form for all three parameters and was determined by $g(t) = g_i(g_f/g_i)^{t/t_{max}}$ with $\lambda_i = 30$, $\lambda_f = 0.01$, $\epsilon_i = 0.3$, $\epsilon_f = 0.05$, $T_i = 20$, $T_f = 200$, and $t_{max} = 40\,000$.

Acknowledgement

This work has been supported by the National Science Foundation under Grant DIR-90-15561 and by a fellowship of the Volkswagen Foundation to T.M..

References

[1] Kohonen T, Mäkisara K, Saramäki T (1984a) Phonotopic maps — insightful representation of phonological features for speech recognition. Proc 7th Int Conf on Pattern Recognition, Montreal, pp 182-185.

[2] Makhoul J, Roucos S, Gish H (1985) Vector quantization in speech coding. Proc. IEEE 73:1551-1588.

[3] Nasrabadi NM, Feng Y (1988) Vector quantization of images based upon the Kohonen self-organizing feature maps. In: IEEE International Conference on Neural Networks, San Diego, 1988, pp 1101-1108.

[4] Nasrabadi NM, King RA (1988) Image coding using vector quantization: A review. IEEE Trans Comm 36(8), pp 957-971.

[5] Naylor J, Li KP (1988) Analysis of a neural network algorithm for vector quantization of speech parameters. In: Proc of the First Annual INNS Meeting, Pergamon Press, New York, p 310.

[6] DeSieno D (1988) Adding a conscience to competitive learning. In: IEEE International Conference on Neural Networks, San Diego, 1988, pp 1117-1124.

[7] Grossberg S (1976a) Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors. Biological Cybernetics, 23, pp 121-134.

[8] Grossberg S (1976b) Adaptive pattern classification and universal recoding: Ii. Feedback, expectation, olfaction, illusions. Biological Cybernetics, 23, pp 187-202.

[9] Hecht-Nielsen R (1988) Applications of counterpropagation networks. Neural Networks, 1(2), pp 131-141.

[10] Kohonen T (1982) Self-organized formation of topologically correct feature maps. Biological Cybernetics 43:59-69.

[11] Kohonen T (1984b) Self-organization and associative memory. Springer Series in Information Sciences 8, Heidelberg.

[12] Graf DH, LaLonde WR (1988) A neural controller for collision-free movement of general robot manipulators. In: IEEE International Conference on Neural Networks, San Diego, 1988, pp 77-84.

[13] Ritter H, Martinetz T, Schulten K (1989) Topology conserving maps for learning visuomotor-coordination. Neural Networks 2, pp 159-168.

[14] Martinetz T, Ritter H, Schulten K (1990) Three-dimensional neural net for learning visuomotor-coordination of a robot arm. IEEE-Transactions on Neural Networks, Vol.1, No.1, pp 131-136.

[15] Ritter H, Kohonen T (1989) Self-Organizing Semantic Maps. Biological Cybernetics, 61, pp 241-254.

[16] Kangas JA, Kohonen TK, Laaksonen JT (1990) Variants of self-organizing maps. IEEE-Transactions on Neural Networks, Vol.1, No.1, pp 93-99.