Proc. Int. Conf. on Art. Neural Netw.
ICANN '94, pp. 701–704
Springer Verlag 1994

An Approximation Network with maximal Transinformation

Rüdiger W. Brause
J.W. Goethe-University, Frankfurt, Germany
(brause@informatik.uni-frankfurt.de)

1 Introduction

One of the most important applications of artificial neural networks is the approximation of an unknown function. It is well known (e.g. Hornik et al., 1989) that a two layer feedforward neural network can approximate each function arbitrarily well when a sufficient number of neural units are provided, see fig. 1.

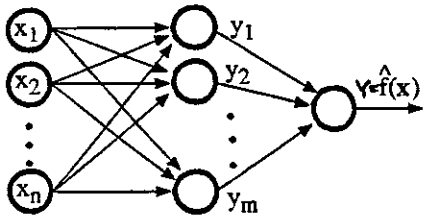

Fig.1 The two layer feedforward approximation network

Very often, units with sigmoidal activation functions and, as learning algorithm, the popular backpropagation algorithm are used, minimizing as objective function the mean squared error and using a gradient descend for the two layers. This approach has some flaws, especially the problems of getting trapped in suboptimal local minima and learning the training samples too specific (*overfitting*).

This paper proposes another approach which avoids those problems. Let us see the approximation as the task to find the parameters of splines interpolating sampled function values f(x). For a linear interpolation (*linear spline*) figure 2 illustrates the situation.

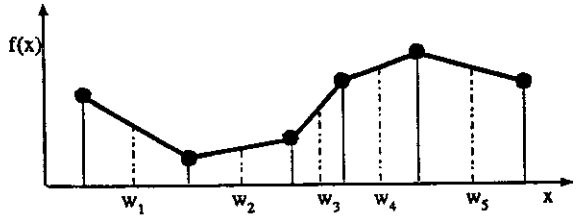

Fig. 2 A piecewise-linear approximation

Here, the approximation task is broken up into two parts. The first part performs the division of the input space {x} into disjoint sets (intervals in fig.2) of events (*classes*) and assigns a typical input value w_i to each class *i*, called a *class prototype*. This is essentially a vector quantization and can be implemented e.g. by a topology-approximating mapping. The second part, the approximation of the bias $f(w_i)$ and the slope of the linear spline, is performed locally by a neuron for each class. For constant-valued splines (*bars*) it is well known that such a mapping has the ability to approximate every arbitrary function sufficiently well if the necessary number of neurons (classes) is provided (Blum et. al. 1991). We implement each part as a separate, independent network layer.

In fact, this kind of two layer neural networks has been already used for robot control by Ritter, et al. (1989). In this paper, we propose new learning rules for the two layers which relay on the objective function of *maximal transinformation*.

2 The non-linear topology-approximating mapping

A topology-approximating mapping version using the least squared error, made by (Kohonen, 1982), became quite popular by its broad range of applications.

This paper makes a new, synthetic approach by using a performance measure of *transinformation* or *mutual information* between the input events and the response of the neurons, see (Linsker, 1988). Following (Shannon, 1949) we preserve the maximal amount of information in the data compression stages and model the information processing process as a pipeline of actions by optimal layers.

Let us consider a clustering or mapping (vector quantization) of an input pattern $\mathbf{x}$ to a class c as it is defined by

$$|\mathbf{x}-\mathbf{w}_c| = \min_k |\mathbf{x}-\mathbf{w}_k| \qquad \mathbf{x},\mathbf{w}_k \in \Re^n \qquad (2.1)$$

Knowing the input pattern $\mathbf{x}$, the Shannon average transmitted information H_{trans} for all inputs and outputs is with the expectation operation $\langle\ \rangle$ for N output points (class prototypes) $\mathbf{w}_i$

$$\begin{aligned} H_{trans} &= \langle I_{trans}\rangle_{w_i,x} = \langle I_{out}\rangle_{w_i,x} - \langle I_{out/inp}\rangle_{w_i,x} \\ &= -\sum_i P(w_i)\log[P(w_i)] \ - \ \sum_x P(x) \sum_i P(w_i/x)\log[P(w_i/x)] \end{aligned} \qquad (2.2)$$

The average transmitted information H_{trans} is maximized when

$$\langle I_{out}\rangle_{w_i,x} \stackrel{!}{=} \max \quad (2.3) \qquad \text{and} \qquad \langle I_{out/inp}\rangle_{w_i,x} \stackrel{!}{=} \min \quad (2.4)$$

It can be easily shown that (2.3) is satisfied when $P(w_i) = P(w_j) = 1/N \quad \forall i,j$. Additionally, for deterministic mappings (see Brause, 1993b) we have $\langle I_{out/inp}\rangle=0$ which is the minimal achievable value.

Let us use directly a stochastic approximation algorithm using the information as an error criterion for a gradient search

$$\mathbf{w}_k(t+1) = \mathbf{w}_k(t) + \gamma(t+1)\, \partial H_{Trans}/\partial \mathbf{w}_k \qquad (2.5)$$

for the maximum of the transinformation H_{Trans}. Evaluating the gradient $\partial H_{Trans}/\partial \mathbf{w}_k$ we get (see Brause, 1993b)

$$\mathbf{w}_k(t+1) = \mathbf{w}_k(t) - \gamma(t+1) \sum_{i\neq k} \frac{(\mathbf{w}_i - \mathbf{w}_k)}{|\mathbf{w}_i - \mathbf{w}_k|^2} (P_k \log P_k - P_i \log P_i + P_k - P_i) \qquad (2.6)$$

As we can see, for equally probable classes with $P_i=P_k$ no change takes place in equation (2.6); the class prototypes remain fixed.

To test this algorithm, we can use the random test which was conceived by (Ritter and Schulten, 1986) with an 1-dim increasing distribution $p(x)=2x$. For the Kohonen map, they found that the point density $M(x)$ of the class prototypes we have $M(x)\sim p(x)^{2/3}$. It was shown (Brause, 1992) that for the topology approximating mapping which preserves the maximum of information, $M(x)$ must approximate the probability distribution of the input patterns directly proportional $M(x)\sim p(x)$. For the random test, the new algorithm showed good results (Brause, 1993b).

3 The local, linear interpolation

Let us consider an input $\mathbf{x}$ which is disturbed by an independent, additive error deviation η. After the linear layer the error still exists and is linearly transformed to the output

$$F(\mathbf{x}) = \bar{\mathbf{y}} = \mathbf{W}(\mathbf{x}+\eta) = \mathbf{y}+\mathbf{y}_\eta \quad \text{with} \qquad \mathbf{y}_\eta = \bar{\mathbf{y}} - \mathbf{W}\mathbf{x} \qquad (3.1)$$

How should we choose the coefficients of W (weights of the neurons) to maximize the transinformation for the approximation of the function values $\bar{y}$? Here, the basic assumption lies in the nature of the random process as a physically generated, normally, spacial distributed error deviation.
Thus, we know

$$p(y_\eta) = p(y_\eta|x) = A \exp(-y_\eta^T y_\eta / 2\sigma_\eta^2) \qquad (3.2)$$

The transinformation becomes

$$H_{trans} = H(\bar{y}) - H(\bar{y}|x) = H(x) + H(\eta) - H(y_\eta|x)$$

Since the random variables x and η and therefore H(x) and H(η) are given, we can only *maximize* the transinformation by *minimizing* $H(y_\eta|x)$. This means

$$H(y_\eta|x) = \langle -\log p(y_\eta|x) \rangle = \langle -\ln A \rangle + \langle (\bar{y}-Wx)^2/2\sigma_y^2 \rangle = \min \qquad (3.3)$$

which in turn is minimal when the squared error becomes minimal. Certainly, this is not true if the patterns are *not* normally distributed as it is often the case for classification purposes. Here, the general criterion of maximal information yields better results than the ordinary least squared error, see for instance (Bridle 1990).

Now let us evaluate the conditions for the best fitting of the data points. In each class region, the unknown vector-valued function **f**(x) is approximated in each component by the a linear spline, i.e. in the multivariate case by a hyperplane. Let us regard this more closely. For each vector component, equation (3.3) means

$$\langle (\bar{y}-w^T x)^2 \rangle = \min \qquad (3.4)$$

By the notation $w \rightarrow w=(w_1,..,w_n,-1)^T$, $x \rightarrow x=(x_1,..,x_n,\bar{y})$ and the introduction of a bias *s* this becomes

$$R(x,w,c) = \langle (w^T x - s)^2 \rangle = \min \qquad (3.5)$$

The objective function R(x,s) takes its minimum for s at $s = w^T\langle x \rangle$. The objective function R(w) easily becomes zero if we reduce the length of w to zero - but this is only the trivial solution. To get the best base vectors of our linear transformation, let us assume a non-zero, constant transformation with det(W)=const.
Therefore, (3.5) becomes

$$R(w,s) = \langle (w^T x - s)^2 \rangle = \langle (w^T(x-\langle x \rangle))^2 \rangle = w^T C w \stackrel{!}{=} \min \qquad (3.6)$$

with the covariance matrix $C=\langle (x-\langle x \rangle)(x-\langle x \rangle)^T \rangle$. By the method of Lagrange multiplier it can be shown (see Brause 1992b) that R(w) has only one minimum and takes it at w being the normalized eigenvector of the covariance matrix C with the smallest eigenvalue. Thus, we are looking for the direction of the smallest diameter of the data cloud {x} and project all samples in the remaining subspace orthogonal to the direction. This method is known as *eigenvector fitting* (Duda et al., 1973) and has some approximation advantages over the simple least squared error method in the sections of a fast changing f(x), see e.g. (Xu et al.,1992). An neural implementation of this kind of eigenvector search was proposed by (Brause 1992a,b, 1993a) who used an anti-Hebb rule for learning.

For each neuron selected, the input $x = (x_1,..,x_n, x_{n+1})$ with $x_{n+1} = f(x_1,..,x_n)$ is centered by the preprocessing stage

$$x(t) \rightarrow x(t) - b_i(t) \quad b_i(t) = \langle x \rangle_i$$

If the random variable x_i is stationary, i.e. the intervall is stable, this can be done iteratively, otherwise, when the first layer is still learning, the iterative average should be replaced by a floating one.

The *training* is accomplished by an Anti-Hebbian learning rule and an normalization for the weights (see Brause 1992a,b)

$$\tilde{w}(t) = w(t-1) - \gamma(t)\, x(t)\, y \qquad y = w^T x = \sum_i^{n+1} w_i x_i \qquad (3.7)$$
$$w(t) = \tilde{w}(t) / |\tilde{w}(t)|$$

In the *activity phase*, the approximation $F(x_1,..,x_n)$ can be obtained directly by the input $(x_1,..,x_n,0)$. We know that for all points of the hyperplane we have no deviation, i.e. $g(x)=w^Tx - s = 0$ with $x = (x_1,..,x_n, x_{n+1})$ and $x_{n+1} = F(x_1,..,x_n)$. So, we get

$$F(x_1,..,x_n) = b_{n+1} - \frac{1}{w_{n+1}} \sum_{i=1}^{n} w_i(x_i-b_i) \qquad (3.8)$$

This can be seen as using the neural network as an *continuous autoassociative memory*: for the input data $x_1,..,x_n$ and zero $f(x_1,..,x_n)$ the neuron computes an output which is the (linearly transformed) function value $F(x_1,..,x_n)=b_{n+1} - y/w_{n+1}$.

4 Convergence analysis

As the main result of this section the convergence of the whole network to one unique stable state is shown. All proofs are given in (Brause, 1993b).

Lemma1 The weights of the quantization network will converge to a stable state which corresponds to the optimal configuration of equal probable classes.

Lemma 2 Given a stable state of the quantization network, each weight of the interpolation network converge to a unique, stable fixpoint.

Theorem Given a stationary distribution of the input samples, the approximation network will converge to a unique, stable fixpoint.

References

E. Blum, L. Li (1991) Approximation Theory and Feedforward Networks; Neural Networks, 4, pp.511-515,

R.Brause (1992) Optimal Information Distribution and Performance in Neighbourhood-conserving Maps for Robot Control; Int. Journal for Computers and Artif. Intelligence, Vol. 11, No.2, pp. 173-199,

R. Brause (1992a) The minimum entropy neuron- A new building block for clustering transformations; in: I. Aleksander, J.Taylor (eds.), Artificial Neural Networks 2, North Holland Publ, Amsterdam , pp. 1095-1098,

R. Brause(1992b) The minimum entropy network; Proc. IEEE Int. Conf. on Tools with Artificial Intelligence TAI-92, pp. 85-92

R. Brause (1993a) A VLSI-Design of the Minimum Entropy Neuron; in: J.Delgado-Frias, W. Moore (Eds.): VLSI for Artificial Intelligence and Neural Networks, Plenum Publ. Corp.,

R. Brause (1993b) An Information-based Approximation Network; (preprint) submitted to IEEE Transactions on Neural Networks

J. Bridle (1990) Probabilistic Interpretation of Feedforward Classification Networks Outputs, with Relationship to Statistical Pattern Recognition; in: F. Fogelman Soulié, J.Hérault (eds.), Neurocomputing, NATO ASI Series, Vol. F68, Springer Verlag, pp. 227-236

R. Duda, P. Hart (1973) Pattern Pattern classification and Scene Analysis; John Wiley&Sons, New York

K. Hornik, M. Stinchcombe, H. White (1989) Multilayer Feedforward Networks are Universal Approximators; Neural Networks, Vol.2, pp.359-366

T. Kohonen (1982) Self-organized Formation of Topologically Correct Feature Maps; Biological Cybernetics, , Vol 43, pp 59-69

R. Linsker (1988) Towards an Organizing Principle for a Layered Perceptual Network; in : D. Anderson (ed), Neural Information Processing Systems, Amer. Inst. of Physics (NY)

H. Ritter, K. Schulten (1986) On the Stationary State of Kohonen's Self-Organizing Sensory Mapping; Biolog. Cybernetics Vol 54, pp. 99-106

C.E. Shannon, W.Weaver (1949) The Mathematical Theory of Information; University of Illinois Press, Urbana

L. Xu, E. Oja, C. Suen (1992) Modified Hebbian Learning for Curve and Surface Fitting; Neural Networks, Vol.5, pp.441-457