Artificial Neural Networks
T. Kohonen, K. Mäkisara, O. Simula and J. Kangas (Editors)
Elsevier Science Publishers B.V. (North-Holland), 1991

Approximator Networks and the Principle of Optimal Information Distribution

Dr. R. Brause, FB20 VSFT, University of Frankfurt,
Postbox 11 19 32, D- 6 Frankfurt 11, FRG.

Abstract

It is well known that artificial neural nets can be used as approximators of any continous functions to any desired degree. Nevertheless, for a given application and a given network architecture the non-trivial task rests to determine the necessary number of neurons and the necessary accuracy (number of bits) per weight for a satisfactory operation.

In this paper the problem is treated by an information theoretic approach. The accuracy of the weights and the number of neurons are seen as general system parameters which determine the the maximal output information (i.e. the approximation error) by the absolute amount and the relative distribution of information contained in the network. A new principle for an optimal information distribution is proposed and the conditions for the optimal system parameters are derived.

As example, for a linear approximation of a quadratic function the optimal system parameters, i.e. the number of neurons and the different resolutions of the variables, are computed.

1. Introduction

One of the most common tasks of artificial neural nets is the approximation of a given function by the superposition of several functions of single neurons. Similar to the well-known theorem of Stone-Weierstraß (see e.g. [4] for regularization networks) Hornik, Stinchcomb and White have shown [5], [6] that every function can be approximated by a two layer neural network (see figure 1) when a sufficient large number m of units is provided. *Sufficient large* - What does this mean? How do we select the appropriate number of neurons for a certain application ?

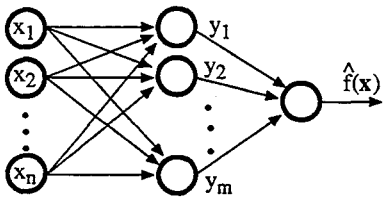

Fig. 1 A two-layer universal approximation network

To give an answer to this question, we first have to remark that our standard modelling of artificial neural nets do not reflect an important feature of reality: the descreteness of all real valued events. Contrary to the modelling of synaptic weights and neuronal activity (spike-frequency) by real numbers, there *do not exist real numbers in reality*.

Instead, there exist a kind of noise and unprecise operations which give rise to a certain amount of error in all real world systems. Especially in simulations and implementations of neural nets we replace all real numbers by more or less fine-grained physical variables, e.g. counters or other descrete variables, with a finite error. This concept is consistent with the restriction of "finite information" in our system: the information of a variable x is defined by

$$\textit{Information} \qquad I(x_i) := -\,\mathrm{ld}\,(P\{x_i\}) \qquad [\,\mathrm{Bits}] \qquad (1.1)$$

If all states x_i are equiprobable, the information is the logarithm of the number of possible states.

For a real number, the number of different values x_i is infinite. Thus, a real number has an infinite amount of information. Because all systems deal with finite amounts of information, there are no "real" real numbers used in neural systems; all weights have a distinguishable number of states (at least due to quantum physics) and therefore contain a certain amount of information in the sense of definition (1.1).

2. Optimal information distribution

Let us now regard an approximation $\hat{f}$ for the function $f: \Re^n \ni \mathbf{x} \rightarrow f(\mathbf{x}) \in \Re$. For example, this can be done by the two-layer neural network of figure 1. Let the maximal error of this approximation be d_f with

$$d_f^2 = (f(\mathbf{x})-\hat{f}(\mathbf{x}))^2 \tag{2.1}$$

Then we can regard the error as a kind of discretization error. Denoting the complete value range with $V_f := |f_{max} - f_{min}|$ we can conclude that there are only V_f/d_f distinguishable, fixed states of the variable f. All other states are undistinguishable from deviations of the fixed states.

Thus, unless we do not know anything more about the input distribution of $\{\mathbf{x}\}$ and therefore nothing more about the error distribution, the output has maximal

$$I_{max} = \mathrm{ld}\,(V_f/d_f) \tag{2.2}$$

bits information. Another parameters, which determine the error of the approximation, is the resolution of the weights or its information content

$$I_w = \mathrm{ld}\,(V_w/d_w) \tag{2.3}$$

with the weight increment d_w and on the other hand the number m of neurons.

Certainly, when we increase the number of neurons and the number $I = \sum_w I_w$ of bits per neuron the approximation will become better and the error will decrease. Nevertheless, for a certain system with a finite amount of information storage capacity (such as a digital computer) the question arises:

What is the best distribution the information, i.e. what is the best choice for m and I_w

1) either to get the minimum approximation error d_f, using a fixed amount of information

or 2) to use the minimal amount of information for a fixed error ?

Neither one neuron with high-resolution weights nor many neurons with one bit weights will give the optimal answer; the solution is in between the range. Let us denote the paramters m, I_w, ... as general system parameters c_1, ..., c_k. Assume on the one hand that we transfer a fixed, small amount of information from one parameter to another and we will find the maximal output information I_{max} increasing by decreasing the approximation error. In this case the information distribution induced by the parameter values of c_1, ..., c_k was not optimal; the new one is better. Let us assume that on the other hand we find that the output information has decreased, then the information distribution is not optimal, too; by making the inverse transfer we can also increase I_{max}. These considerations lead us to the following extremum principle:

> In an *optimal information distribution* a small (virtual) change in the distribution (a change in c_1, ..., c_k) neither increases nor decreases the maximal output information.

A small increment of information δI will produce a change δI_{max} in the maximal output information

$$\delta I_{max} = \frac{\partial}{\partial I} I_{max}\, \delta I = \delta I \sum_{i=1}^{k} \frac{\partial}{\partial c_i} I_{max}(c_1, ..., c_k)\, \frac{\partial c_i}{\partial I} \tag{2.4}$$

Each term in the sum of equation (2.4) represents an information contribution of a system parameter when we increase the overall system information I. According to the principle above, an optimal distribution is given when all terms in the sum i.e. all information contributions of the system parameters are equal.
With the definition (2.2) we get for each term of the sum of (2.4)

$$\frac{\partial}{\partial c_i} I_{max}(c_1, ...,c_k) = \frac{\partial}{\partial c_i} (ld\ (V_f)-ld(d_f)) = -\frac{1}{d_f} \frac{\partial d_f}{\partial c_i} \tag{2.5}$$

and so the optimal distribution resides when

$$\frac{\partial d_f}{\partial c_1} \frac{\partial c_1}{\partial I} = ... = \frac{\partial d_f}{\partial c_k} \frac{\partial c_k}{\partial I} \tag{2.6}$$

is fulfilled. The k independant terms gives us (k-1) equations fo k variables c_1, ..., c_k, leaving us with a degree of freedom of one. So, choosing the amount of available information storage $I(c_1, ...,c_k)$, the parameters c_1, ..., c_k are fixed and the smallest error for the particular application will result. On the other hand, for a certain maximal error a certain amount of network information is necessary.

It should be noted that condition (2.6) is identical to the solution suppliedby the classical approach with Lagrange multipliers for finding an extremum subject to a constraint condition of a multi-argued function [3].

3. The approximation of a quadratic form

In this section first we want to demonstrate the use of the principle above by a very simple example: the approximation of a quadratic form by a polygone. Throughout in this example, all design decisions (choice of value ranges etc.) are taken for demonstration purposes only.
The use of the information distribution principle in a more "realistic" example of the neural network for a robot control algorithm (which uses a non-linear, learned mapping) is shown elsewhere [2] in the context of storage optimization [1]. In contrast to this quite complicated application which uses some numerical approximation methods let us evaluate in this paper a simple, analytically treatable example for demonstration purpose.

Let us consider the simple, non-linear function $f(x) = ax^2 + b$. The approximation to this function can be accomplished by a network with one input x shown in figure 2.

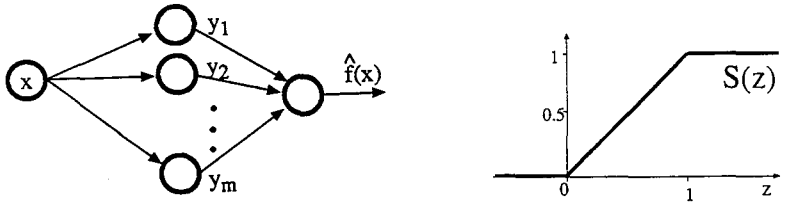

Fig. 2 The network for approximating $f(x) = ax^2 + b$ and the unit output function

Each neuron has the output function $y_i = S(z_i)$ with the activation function z_i

$$z_i = \Sigma_j\ w_{ij}\ x_j \tag{3.1}$$

which becomes for the first layer

$$z_i = w_i x + t_i \qquad \text{with the threshold } t_i \tag{3.2}$$

and for the second layer

$$\hat{f}(x) = \Sigma_i W_i S(z_i) + T \tag{3.3}$$

Let us assume that we use simple limited linear output functions as squashing functions

$$S(z_i) = \begin{cases} 1 & 1 < z_i \\ z_i & 0 \le z_i \le 1 \\ 0 & z_i < 0 \end{cases} \tag{3.4}$$

The definition (3.4) satisfy the conditions $S(\infty)=1$, $S(-\infty)=0$ of [HOR89] and is shown in figure 2 on the right hand side.

Let us assume that all the weights have converged by a proper algorithm for an approximation of the non-linear function by linear segments. The whole input interval $[x_0,x_1]$ is then divided by the *m* neurons of the first layer into *m* intervals Δx. The output $z_i \in [0,1]$ of each neuron is only linear when x is from its interval $[x_i-\Delta x/2, x_i+\Delta x/2]$, otherwise it is constant 0 or 1. In the second layer it is then weighted by W_i. Together with the offset of the previous intervals the weighted influence $W_i S(z_i)$ represents the approximation function $\hat{f}(x)$ in the interval $[x_i-\Delta x/2, x_i+\Delta x/2]$.

$$\hat{f}(x) = \sum_{j=1}^{m} W_j S(z_j) + T = \sum_{j=1}^{i-1} W_j + W_i S(z_i) + T \tag{3.5}$$

The resulting approximation is shown in figure 3.

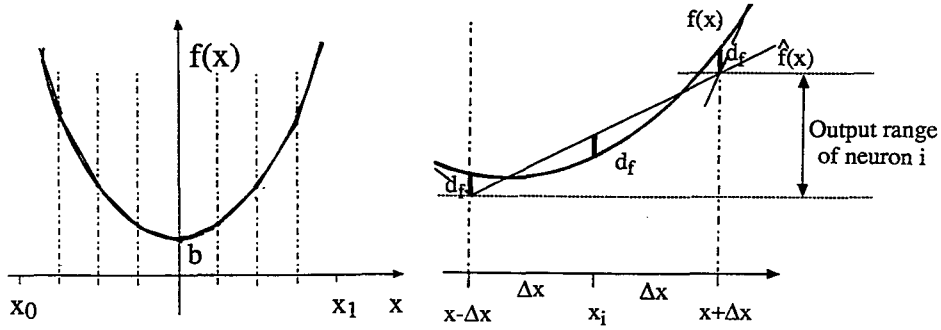

Fig.3 The non-linear function and its approximation

The corresponding values for w_i, t_i, W_i and T can be easily calculated.
From the conditions of (3.4) $z(x_i-\Delta x/2) = 0$ and $z(x_i+\Delta x/2) = 1$ and by (3.2) we get

$$w_i = 1/\Delta x = m / (x_1-x_0) \tag{3.6}$$

and

$$t_i = - w_i (x_i-\Delta x/2) = - m x_i/(x_1-x_0) + 1/2 \tag{3.7}$$

For W_i let us regard the interval in figure 3 on the right hand side.
Let us choose W_i such that in each segment the spline is the tangent of f(x) in x_i

$$W_i = \frac{\partial f(x_i)}{\partial x} = \frac{\partial}{\partial x}(ax^2 + b)\Big|_{x_i} = 2ax_i \tag{3.8}$$

Then the basic threshold T becomes the offset of the approximation at x_0.

To calculate the information after (2.3) for w_i, t_i, W_i and T first we have to define the range V_w, V_t, V_W and V_T of the variables, see [3]. The maximal resolution error δ of a variable *v* in one state is just the half of the resolution increment of equation (2.3)

$$\delta v = d_v/2 = V_v/2\ 2^{-I_v} \tag{3.9}$$

and can be easily calculated for δw, δt, δW and δT using V_w,V_t,V_W and V_T.
In the present approximation function example our information distribution system parameters c_1, ...,c_k are represented by the number of bits per variable I_w,I_t,I_W and I_T and the number m of neurons in the first layer.

Since we do not assume anything about the input probability distribution p(x), we can not compute the average error. Instead, as a performance measure of the approximation network let us compute the maximal possible error. The maximal approximation error is given by the worst case condition that the linear approximation error d_{lin} and the resolution error d_{res} do not compensate each other but adding up to

$$d_f^{max} = d_{lin}^{max} + d_{res}^{max} \tag{3.10}$$

In [3] the error of the linear approximation in the interval i and the error due to the finite resolutions I_w,I_t,I_W, I_T and m are evaluated

$$d_{lin}^{max} = a/2((x_1-x_0)/2m)^2 = m^{-2}a(x_1-x_0)^2/8 \tag{3.11}$$

$$d_{res}^{max} = 2ax_1\,(\delta w\, x_1+\delta t) + m\delta W + \delta T \tag{3.12}$$

The whole information contained in the network is the sum of the information $m(I_w+I_t)$ of the m weights and thresholds in the first layer and the information mI_W+I_T of the m weights and the threshold in the second layer

$$I = m(I_w+I_t+I_W) + I_T \tag{3.13}$$

The condition (2.5) for an optimal information distribution then becomes

$$\frac{\partial}{\partial m}(d_{lin}^{max} + d_{res}^{max})\left(\frac{\partial I}{\partial m}\right)^{-1} = \ldots = \frac{\partial}{\partial I_T}(d_{lin}^{max} + d_{res}^{max})\left(\frac{\partial I}{\partial I_T}\right)^{-1} \tag{3.14}$$

This gives us **5 terms** (see [3]) which should be equal to yield an optimal information distribution. Let us evaluate the equalities.

With **term 2) = term 3)**, we know that

$$x_1\,\delta w = \delta t \tag{3.15}$$

The resolution errors of the weights and the threshold of the first layer should be in the same order since they produce the same final error by multiplication with W. Using the equations due to (3.9) (see [3]) gives us

$$I_t = I_w + C_1 \qquad \text{with } C_1 := \mathrm{ld}((x_1-x_0)/x_1) \tag{3.16}$$

The information of the threshold has a constant offset from the information of the weights.

term4) = term5)

The analogue case for the threshold and weights of the second layer reveals

$$\delta W = \delta T \tag{3.17}$$

The threshold should be as fine grained as the weights since it is always involved in the output accuracy.

$$I_T = I_W + C_2 \qquad \text{with } C_2 := \mathrm{ld}((ax_1^2+b)/2a(x_1-x_0)) \tag{3.18}$$

The threshold information of the second layer has also an offset to the weights which depends on the number of inputs from the first layer.

term3) = term4)

The comparison between the threshold of the first layer and the weights of the secon layer gives

$$2ax_1 \frac{\delta t}{m} = \delta W \qquad (3.19)$$

and therefore

$$I_t = I_W + C_3 \qquad \text{with } C_3 := \mathrm{ld}(x_1/(x_1-x_0)) \qquad (3.20)$$

term1) = term2)

The condition for the number of neurons gives (see [3])

$$(-\frac{a}{4m^3}(x_1-x_0)^2 + 2ax_1^2\frac{\delta w}{m} + 2ax_1\frac{\delta t}{m} + \delta W)\,(I_w+I_t+I_W)^{-1} = -\frac{\ln(2)}{m} 2ax_1^2\,\delta w$$

which, after some manipulations [3] using equations (3.15), (3.16), (3.19), (3.20), becomes

$$m = [2^{I_t}(x_1-x_0)^2/(x_1 12(\ln(2)I_t+1))]^{1/3} \qquad (3.21)$$

Example

Let us consider an information of 32 bits in the threshold t. In the simple case of x_0=-1, x_1=+1, a=1, b=0 we have with I_t := 32 bit

$$m = 397 \text{ neurons}, \quad I_w = I_t - C_1 = 31 \text{ bit}, \quad I_W = I_t - C_3 = 33 \text{ bit}, \quad I_T = I_W + C_2 = 31 \text{ bit}$$

The overall information in the network is with (3.13)

$$I = m(I_w+I_t+I_W) + I_T = m3I_t + I_T = 38 \text{ kBit} \sim 4.8 \text{ kByte}$$

4. Conclusion

The principle of *optimal information distribution* is a criterium for the efficient use of the different information storage ressources in a given network. Furthermore, it can be used as a tool to balance the system parameters and to obtain the optimal network parameter configuration according to the minimal usable storage for a maximal given error.

In the paper the use of the principle was demonstrated for the example of a simple non-linear function approximation. The conditions for optimal system parameters were stated, their solutions were analytically computed and their nature was explained.

References

[1] R.Brause,
Performance and Storage Requirements of Topology-Conserving Maps for Robot Manipulator Control,
Internal Report 5/89, Fachbereich Informatik, University of Frankfurt, FRG

[2] R.Brause
Optimal Information Distribution and Performance in Neighbourhood-conserving Maps for Robot Control, IEEE Proc. Tools for Art. Int. TAI-90, Dulles 1990

[3] R.Brause
Approximator Networks and the Principle of Optimal Information Distribution,
Internal Report 1/91, Fachbereich Informatik, University of Frankfurt, FRG

[4] F.Girosi, T. Poggio
Networks and the Best Approximation Property,
Biolog. Cybern. (1990) Vol. 63, pp. 169-176

[5] K. Hornik, M. Stinchcomb, H.White
Multilayer Feedforward Networks are Universal Approximators,
Neural Networks (1989), Vol 2, pp. 359-366

[6] M. Stinchcomb, H.White
Universal approximation using feedforward networks with non-sigmoid hidden layer activation functions,
Proc. Int. Joint Conf. Neural Networks, Washington DC, June 1989,pp. I/607-611