IEEE Proc. Int. Conf. on Tools for
Artificial Intelligence TAI-91, San José 1991

The Determination of Neural Network Parameters by Information Theory

Rüdiger Brause

FB20 VSFT, University of Frankfurt,
Postbox 11 19 32, D- 6000 Frankfurt 11, FRG.

Abstract

It is well known that artificial neural nets can be used as approximators of any continous function 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. The demand for maximal output information gives us the necessary conditions for optimal system parameters and leads to the new principle of optimal information distribution.

For the example of a simple linear approximation of a non-linear, 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 the output of several 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

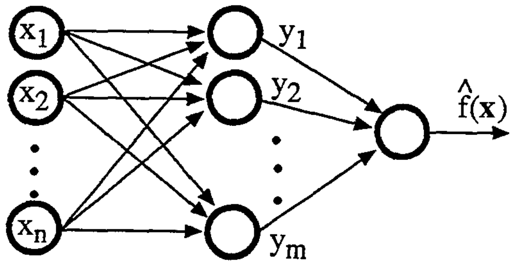

Fig. 1 A two-layer universal approximation network

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 ?

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, the operations are noisy and unprecise and 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

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

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. This argument is also valid for the averaged information, the entropy, introduced by Shannon [7]

$$\begin{aligned} H = \langle I(x)\rangle &= -\Sigma_i\, P_i\, \mathrm{ld} P_i \\ &= -\ p(x)\ \mathrm{ld}\ p(x)\, dx \qquad (1.2) \end{aligned}$$

which also becomes infinity for an uniform distribution p(x) = 1/d over the whole inifinite range of the real variable.

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 distributions

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

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

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 distinguishable, fixed states of the variable f which differ by an increment of $d=2d_f$. All other states are undistinguishable from deviations of these states within the error bounds.

Thus, unless we do not know anything more about the input distribution of {x} and therefore nothing more about the error distribution, the output has (using the maximal error) with (1.1) at least

$$I_{out} = \text{ld}\,(V_f/d) \qquad (2.2)$$

bits of information.

The parameters, which determine the error of the approximation, are on the one hand the resolution of the weights or its information content

$$I_w = \text{ld}\,(V_w/d_w) \qquad (2.3)$$

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 $I = \Sigma_w I_w$ 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) and therefore a finite and constant memory size for the system state, 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.

This is shown in figure 2. Here we have for the example of the approximation of a quadratic function (see part 3) a

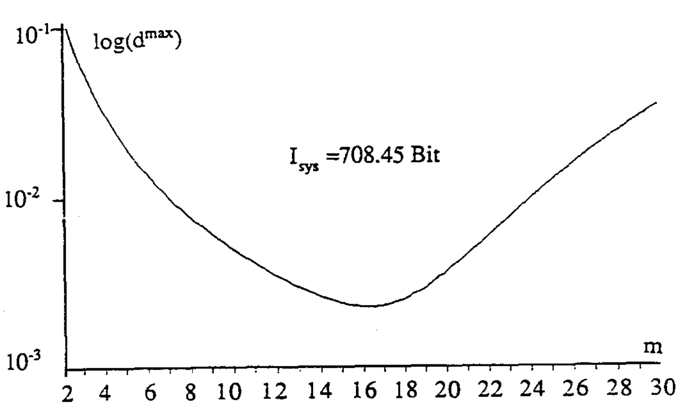

Fig. 2 The approximation error for different *m* at constant system information

plot of the approximation error as a function of the number *m* of neurons used. The constant system information determines directly by (3.12) the remaining value for the weight resolutions when they are chosen to be all equal.

Therefore, we have to solve the problem: What is the best distribution of the information, i.e. what is the best choice for *m* and I_w to maximize the information I_{out} in order to get the minimum approximation error d_f, using a fixed amount of state information ?

Let us denote the parameters *m*, I_w, ... as general system parameters $c_1, ..., c_k$.

2.1 Optimal system parameters

The minimal information I_{out} introduced above is a multivariate function $I_{out}(c_1,...,c_k)$. If we want to get the maximal information out of the system using only a certain amount of system information we look for an optimal parameter tupel $(c_1^*,...,c_k^*)$ such that

$$I_{out}(c_1^*,...,c_k^*) = \max_{c_1,...,c_k} I_{out}(c_1,...,c_k) \qquad (2.4)$$

which is accompanied by the restriction that the whole information I_{sys} in the system should not be changed during the maximization process

$$I_{sys}(c_1,...,c_k) = I_s = \text{const} \qquad (2.5)$$

By these two conditions the relative maximum (2.4) of the multivariate function I_{out} is searched under the restriction of (2.5). The standard method to solve a problem like this is the method of Lagrange multipliers. For this purpose let us define the differentiable functions

$$I(c_1,...,c_k) := I_{sys}(c_1,...,c_k) - I_s = 0 \qquad (2.6)$$

and $L(c_1,...,c_k,\lambda) := I_{max}(c_1,...,c_k) + \lambda I(c_1,...,c_k)$
Lagrange function

Since the Lagrange function includes the restriction (2.5), the necessary conditions for a relative maximum of the Lagrange function gives us the optimal values for the system parameters

$$\frac{\partial}{\partial c_1} L(c_1^*) = 0 \quad \dots \quad \frac{\partial}{\partial c_k} L(c_k^*) = 0$$
$$\frac{\partial}{\partial \lambda} L(\lambda^*) = 0 \qquad (2.7)$$

The conditions above transform to the equations

$$\frac{\partial}{\partial c_1} I_{out}(c_1^*) + \lambda \frac{\partial}{\partial c_1} I(c_1^*) = 0$$
$$\dots$$
$$\frac{\partial}{\partial c_k} I_{out}(c_k^*) + \lambda \frac{\partial}{\partial c_k} I(c_k^*) = 0 \qquad (2.8a)$$
$$I(c_1^*,...,c_k^*) = 0 \qquad (2.8b)$$

Let us assume that the function $I(c_1,...,c_k)$ is invertible for each system parameter. Then we know that

$$\frac{\partial}{\partial c_i} I(c_i) = \frac{\partial}{\partial c_i} I_{sys}(c_i) = \left[\frac{\partial c_i}{\partial I_{sys}}(c_i)\right]^{-1} \qquad (2.9)$$

and the conditions (2.8) become

$$\frac{\partial}{\partial c_1} I_{out}(c_1^*) \frac{\partial c_1}{\partial I_{sys}} = -\lambda = \dots = \frac{\partial}{\partial c_k} I_{out}(c_k^*) \frac{\partial c_k}{\partial I_{sys}} \qquad (2.10a)$$
$$I_{sys}(c_1^*,...,c_k^*) = I_0 \qquad (2.10b)$$

The last condition (2.10b) is just our well-known restriction (2.5). The *k* terms of (2.10a) say that for the necessary condition of an optimal information distribution all the terms should be equal. Then a small change in the parameters (e.g. increasing c_i and decreasing c_j), i.e. a virtual change in the information distribution will produce the same amount of increase and decrease in I_{out} and do not change it. This is the *principle of optimal information distribution* as it was first introduced in [2].

The *k* independant terms give us (k-1) equations for *k* variables $c_1, ..., c_k$, leaving us with a degree of freedom of one. So, choosing the amount of available information storage $I_{sys}(c_1, ...,c_k):=I_s$, the parameters $c_1, ..., c_k$ are fixed and with I_{out} the smallest error d_f for the particular application will result. On the other hand, for a certain maximal error a certain amount of network information is necessary.

In order to get the parameter values **c*** for the optimal information distribution of a given system the conditions (2.10a) have to be evaluated. Let us do this now for a simple example.

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.

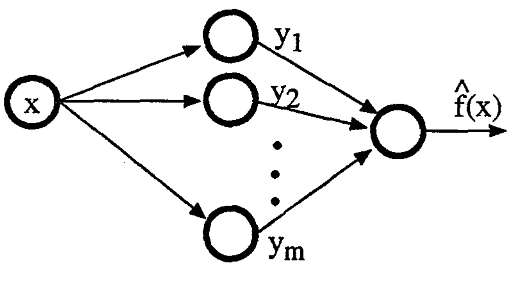

Fig. 2 The network for approximating $f(x) = ax^2 + b$

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

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

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 [5].

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 output of each neuron *i* is only linear when x is from the interval $[x_i-\Delta x/2, x_i+\Delta x/2]$ with $x_i=x_0+i\Delta x-\Delta x/2$, otherwise it is constant 0 or 1. Therefore, for the piecewise-linear polygone approximation the whole input interval $[x_0,x_1]$ is divided by the *m* neurons of the first layer into *m* non-overlapping intervals Δx. The normalized variable $z_i \in [0,1]$ is 1/2 at x_i.

In the second layer the output z_i is then weighted by W_i. Together with an offset of the previous intervals it represents there the linear part of the approximation function $\hat{f}(x)$ in the interval $[x_i-\Delta x/2, x_i+\Delta x/2]$.

$$\hat{f}(x) = \sum_{i=1}^{m} W_i S(z_i) + T = \sum_{i=1}^{k-1} W_i + W_k S(z_k) + T \tag{3.5}$$

The resulting approximation and the output functions of the single neurons are shown in figure 3 for the case of m=5 neurons.

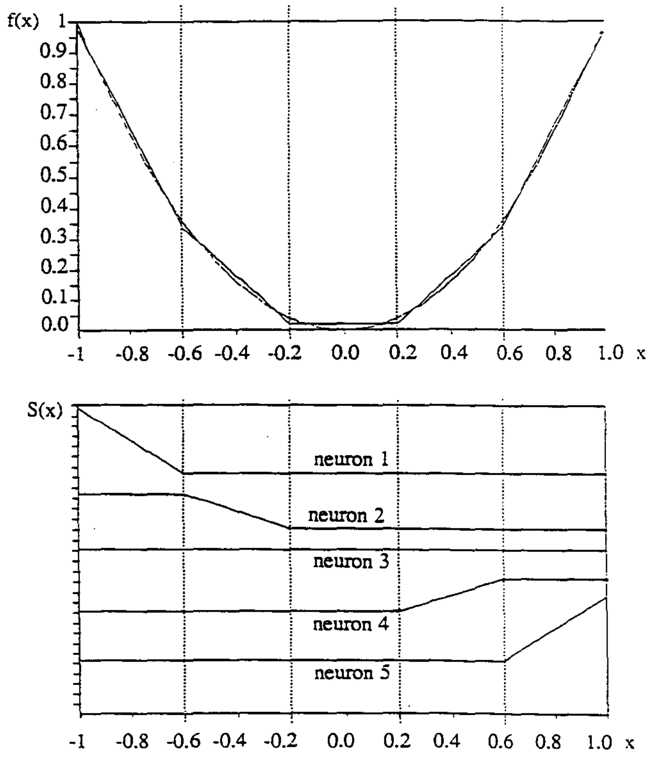

Fig.3 The approximation as sum of the neural output

The corresponding values for w_i, t_i, W_i and T can be easily calculated.
From the conditions of (3.4) we can conclude

$$z|_{x_i-\Delta x/2} = 0 \qquad z|_{x_i+\Delta x/2} = 1$$

and by (3.2) we get

$$w_i = 1/\Delta x = m / (x_1-x_0) \qquad (3.6)$$
$$t_i=-w_i(x_i-\Delta x/2)=(x_0/\Delta x)+1-i=-mx_i/(x_1-x_0) +1/2 \qquad (3.7)$$

Let us choose W_i such that in each segment the spline is the tangent of f(x) in x_i

$$\frac{\partial f(x_i)}{\partial x} = \frac{\partial (ax^2+ b)}{\partial x}\bigg|_{x_i} = 2ax_i := \Delta y/\Delta x$$

Since the output S(z) is normalized between 0 and 1, we have to choose the weights W_i as the normalized tangent $\Delta y/1$. Therefore, the weights become

$$W_i := \Delta y/1 = 2ax_i \,\Delta x \qquad (3.8a)$$

Then the basic threshold T becomes the offset of the approximation at x_0. Using (A1.1) of [3] we get

$$T = f(x_0) -d_{lin} = ax_0^2 + b - a/2\ (\Delta x/2)^2 \qquad (3.8b)$$

To calculate the information after (2.3) for w_i, t_i, W_i and T we have first 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)

$$\delta_v = d_v/2 = V_v/2\ 2^{-I_v} \qquad (3.9)$$

and can be easily calculated for δ_w, δ_t, δ_W and δ_T 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 add up to

$$d_f^{max} = d_{lin}^{max} + d_{res}^{max} \qquad (3.10)$$

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 to

$$d_{lin}^{max} = a/2(\Delta x/2)^2 \qquad (3.11)$$

$$d_{res}^{max}= 2ax_1\Delta x(\delta_w x_1+\delta_t) + m\delta_W + \delta_T \qquad (3.12)$$

The whole system state 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

$$I_{sys} = m(I_w+I_t+I_W) + I_T \qquad (3.13)$$

The condition (2.5) for an optimal information distribution then becomes (3.14):

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

This gives us 5 terms which should be all equal to yield an optimal information distribution. The 4 equations for the 5 parameters provides us in [3] with the following solutions:

$$I_t = I_w + C \qquad \text{with } C:= ld((x_1-x_0)/x_1) \quad \textit{constant offset}$$
$$I_W = I_t + C$$
$$I_T = I_W + ld(g_T(m)/2) -ld((x_1-x_0)^2/m)$$

and the equation for the number of neurons

$$m = h(m,I_T)^{1/3}$$

This we can use for numerically given I_T as an iteration formula at the (t+1)-th iteration for *m* by the function h(.):

$$m(t+1) = h(m(t),I_T)^{1/3}$$

Since the derivative of $h(m)^{1/3}$ is lower 1, the convergence condition is satisfied and the iteration converges to m*.

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 system storage (system description information) for a given maximal performance error.

In the paper the principle was derived by maximizing the output information of the network. 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 and their solutions were analytically computed.

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*, Biol. 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

[7] C.E. Shannon, W.Weaver: *The Mathematical Theory of Information,* University of Illinois Press, Urbana 1949