

A method for finding the optimal number of learning samples and hidden units for function approximation with a feedforward network

Vytautas Vysniauskas*, Frans C.A. Groen, Ben J.A. Kröse

Faculty of Mathematics and Computer Science, University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands

Abstract: This paper presents a methodology to estimate the optimal number of learning samples and hidden units needed to obtain a desired accuracy of a function approximation by a feedforward network. A model of the approximation error is derived of which the parameters can be determined experimentally. Given the computational complexity of the learning rule an optimal learning set size and number of hidden units can be found resulting in minimum computation time for a given desired precision of the approximation. This approach was successfully applied to optimize the learning of a function, which performs camera-robot mapping of a visually guided robot arm.

1 Introduction

Results of recent research established multilayer feedforward networks as a class of universal approximators ([2], [3], [4]). However, failures to approximate a function with the desired accuracy can be attributed to inadequate architecture of a network, inadequate number of learning samples, the limited precision of a computer or to an inadequate learning procedure.

In this paper the attention is devoted to the problem how many hidden units must be used in a feedforward network and how many learning samples provide enough information to construct the mapping with the desired accuracy. The basic idea is to express the approximation error as a function of the number of learning samples and the number of the network weights. For each particular application it is necessary to estimate experimentally only a few parameters. Given the computational complexity of the learning procedure we can evaluate immediately the optimal size of the learning set and suitable architecture of a feedforward network in order to obtain the desired accuracy with minimal computational resources. The last section of this paper describes the application of this method for optimizing the learning of a function, which performs the camera-robot mapping of a visually guided robot arm.

2 Definitions

The input-output function for a feedforward network can be denoted as $G_w(\mathbf{x})$ where w is a set of the network adjustable parameters. A feedforward network with one hidden layer and output units without bias has $g = \dim(w) = (\nu + \mu + 1)h$ adjustable parameters where ν , μ and h are respectively input, output and hidden layer dimensions. Let f be a function of multidimensional mapping from ν -dimensional input space to μ -dimensional output space $f: \mathbb{R}^\nu \rightarrow \mathbb{R}^\mu$. It was shown by Hornik et al. [4] that there exists a *best approximation* G_{w^*} of the function f approximation for a given architecture of the feedforward network¹. In practice, the goal is to create from a given set

*the current address is: the Institute of Mathematics and Informatics, Department of Neuroinformatics, Akademijos 4, 2600 Vilnius, Lithuania

¹It is an evident fact, that a feedforward network has multiple solutions due to the weights permutation and sign flips, but after the proper rearrangement of the network weights we can consider this as one, unique solution.

of samples Z_N the *optimal approximation* $G_{w^*|Z_N}$. This optimal approximation can never be better than the best approximation evaluated from the infinite learning set.

We define the general error measure (approximation error in general) as the difference between an arbitrary solution w and the function f as $E_a(w) = \int_X \|G_w(\mathbf{x}) - f(\mathbf{x})\|^2 dx$.

Representation error. The difference between the best available approximation G_{w^*} and the function f is defined as a representation error E_r , which informs us how accurately can a given network represent a given function when we have perfect knowledge (noiseless, infinitive learning set) about the function. According to the main result in [4], the representation error E_r of the approximation can be arbitrarily small if a sufficient number of the hidden units is available

$$E_r(w^*) = E_a(w^*) = \int_X \|G_{w^*}(\mathbf{x}) - f(\mathbf{x})\|^2 dx, \quad \lim_{N \rightarrow \infty} E_r = 0. \quad (1)$$

Generalization error. Since we always have a finite learning set, an extension of this learning set domain (generalization) yields an additional error

$$E_g(w^*|Z_N) = E_a(w^*|Z_N) - E_r(w^*) \quad (2)$$

named as the generalization error which goes to zero when the learning set increases.

Optimization error. Unfortunately, we are not able even to find $G_{w^*|Z_N}$, since no theory at all exists how the knowledge about the function can be explicitly transformed into the weights of a suitable neural network. The vehicle to set the weights optimally $w \rightarrow w^*$ is a numerical procedure of nonlinear optimization, which produces the *actual approximation* $G_{w^*|Z_N}$ ($w^*|Z_N \in W^* \subset W$), different from the $G_{w^*|Z_N}$ due to imperfect optimization. We define the optimization error as follows

$$E_{opt}(w^*|Z_N) = \int_X \|G_{w^*|Z_N}(\mathbf{x}) - G_{w^*|Z_N}(\mathbf{x})\|^2 dx, \quad (3)$$

which is a measure of the difference between the actual and the optimal solution evaluated over the whole domain of X . In general, the total error of the approximation $E_a(w^*|Z_N)$ involves implicitly the optimization error. It can be shown that

$$E_a(w^*|Z_N) \leq E_r(w^*) + E_g(w^*|Z_N) + E_{opt}(w^*|Z_N) \quad (4)$$

The equality holds only if $E_{opt} = 0$ ($w^* \equiv w^*$), so E_a can never be lower than the representation error.

Likelihood of valid generalization. The errors we defined above are characteristic for an *individual realization* of the approximation, rather than a given network architecture and a given number of samples. Only the representation error E_r is unique for a given network architecture. We can make estimates in sense of average over all possible learning sets Z_N and random initializations of the network by introducing the average optimization, approximation and generalization errors ϵ_{opt} , ϵ_a , ϵ_g as functions of h and N . Now we derive immediately from (4) a relationship between the average errors

$$\epsilon_a \leq \epsilon_g + E_r + \epsilon_{opt} \quad (5)$$

A necessary condition for valid generalization is *uniform convergence* of E_g when N becomes large, otherwise a single realization from W^* may have a poor generalization (large E_g value). In other words, we want the probability that there is some $w^* \in W^*$ such that $E_g(w^*|Z_N)$ differs significantly from ϵ_g be very small

$$Pr \left[\sup_{w^* \in W^*} |E_g(w^*|Z_N) - \epsilon_g| > \epsilon \right] \leq \delta(N), \quad \lim_{N \rightarrow \infty} \delta(N) = 0. \quad (6)$$

The problem was solved conceptually by Vapnik and Chervonenkis [5] by introducing the concept of Vapnik-Chervonenkis (VC) dimension, a measure how fast the convergence is achieved (likelihood of generalization). This is of practical importance (especially in the approximation of the multidimensional mapping, when $\nu > 1, \mu > 1$) because this determines the number of examples and suitable architecture of a network needed to guarantee generalization within given tolerance parameters. As was shown by Baum and Hauser [1], in the case of neural networks VC dimension is closely related to the number of weights in the architecture.

3 The approach

As the basis of our approach we explored the relationship (5). Without loss of generality we suppose that the optimization is perfect ($\epsilon_{opt} = 0$).

Asymptotic expansion. Since the goal is to create an accurate function approximation we are interested in the behaviour of ϵ_a for $N \gg 1$ and $h \gg 1$. In such a case we can expand ϵ_a at the point (N_0, h_0) for any $N \ll N_0$ and $h \ll h_0$ as follows

$$\epsilon_a(N, h) = \sum_p \sum_{\alpha+\beta=p} \gamma_{\alpha\beta} \left(\frac{1}{N} - \frac{1}{N_0} \right)^\alpha \left(\frac{1}{h} - \frac{1}{h_0} \right)^\beta \approx \sum_p \sum_{\alpha+\beta=p} \frac{\gamma_{\alpha\beta}}{N^\alpha h^\beta} = \sum_p \epsilon_a^{(p)}(N, h), \quad (7)$$

where α, β, p are natural numbers and $\gamma_{\alpha\beta}$ are unknown parameters of the expansion, which satisfies the asymptotic conditions from the relationships (1-2). As we want to represent the asymptotic behavior of ϵ_a for $N \gg 1$ and $h \gg 1$, only few terms of order $p_{min}, p_{min}+1, \dots, p_{max}$ can be included. This truncation describes the asymptotical model of the error function (AMEF), different values of p_{min} and p_{max} yield different solutions.

Minimization of computational resources. In practice the knowledge about ϵ_a is of extreme importance, providing the possibility to find the optimal strategy so that the learning time would be minimal to obtain the desired accuracy. If we know additionally the computational complexity of the learning procedure which can be expressed as $r \sim N^k h^k$ as a function of N and h , where k is the order of complexity, it is possible to find an unique pair (N_0, h_0) resulting in the minimal computation time for a given precision of the approximation. Note that $k = 1$ for methods like *conjugate gradient* and *backpropagation*. The solution (N_0, h_0) is the minimum of r under the condition $\epsilon_a = \epsilon^*$.

Methodology to estimate γ parameters. The proper AMEF can be chosen by using the least-squares criterion. Note, the estimation of the average approximation error must be done by averaging over all different learning sets and over all random initializations of the network. Estimated parameters are needed to determine (N_0, h_0) .

4 Results

We applied the approach to approximate the camera-robot mapping for the adaptive robot control, for which the number of inputs $\nu = 5$ and the number of outputs $\mu = 3$. We estimated the approximation error ϵ_a at 64 points on the 8×8 grid in (N, h) framework ($50 < N < 400$, $5 < h < 40$) from an independent test set consisted from 4000 samples, and the network learning was limited to 5000 epochs. At each point the average from 10 realizations was computed. We used *conjugate gradient* procedure for the network weights adjusting.

We obtained a very good fitting for the solution

$$\epsilon_a^{(2)}(N, h) = \frac{700}{N^2} + \frac{711}{N^1} + \frac{702}{h^2} \quad (8)$$

(see Figure 1), where only the second term of the expansion was used. More detailed analysis showed that the approximation error can be described more exactly with the solution ($p_{min} = 1, p_{max} = 2$), and the relative contribution of these terms was respectively 26 and 74 percent. We found no practical gain to use more terms of the expansion, because the mean-square error decreases slowly with increasing of the regression order. These results also suggested that the optimal number of learning samples for these particular applications is very close to the number of the network weights.

For the solution $\epsilon_a^{(2)}$ it can be shown that there exists a lower boundary of the number of learning samples and hidden units which are necessary to obtain the desired accuracy of approximation

$$N > \sqrt{\frac{720}{\epsilon^*}}, \quad h > \sqrt{\frac{702}{\epsilon^*}} \quad (9)$$

where ϵ^* is the desired accuracy of approximation. Similar boundaries also exist for other solutions also. Probably, failures to approximate a function can be particularly explained as a lack of information about these boundaries.

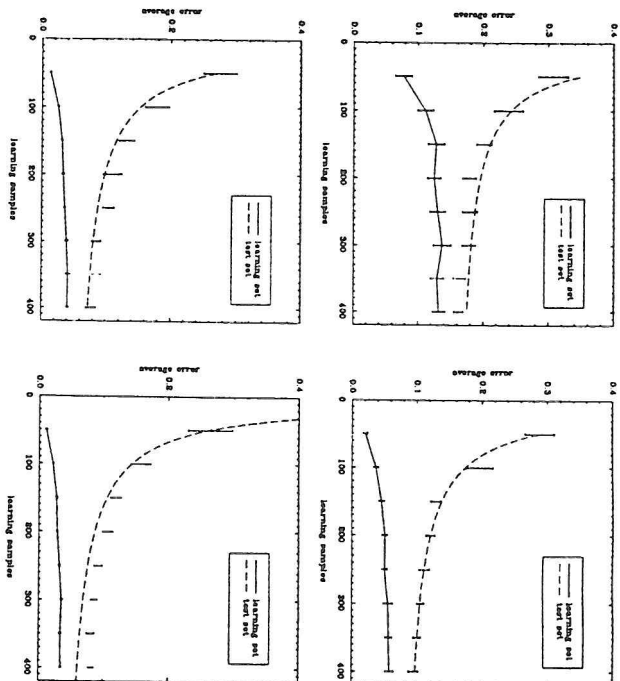


Figure 1: Comparison of the experimental results with the theoretical estimation (dashed line) for the network with 10, 20, 30, 40 hidden units. The learning error is represented by the solid line, error bars are indicators of a dispersion of the average value.

Concluding remarks. Future research will focus on a number of issues concerning the model and applying it to a number of different problems. We did not include in our approach the factor of the limited number of iterations needed to obtain the desired accuracy, because this was in our case not the most prominent source of error. The optimization error can be incorporated by adding extra-term as a function of N and h . A study will be carried out on the relationship between the architecture of a network and the number of terms in the expansion (7). Also the influence of the noise in the learning samples in the model needs some further attention.

References

- [1] Baum E.B., Hausler D. "What Size Net Gives Valid Generalization?", *Neural Computation* 1, 1989, pp. 151-160.
- [2] Cybenko G. "Approximation by Superpositions of a Sigmoidal Function", *Math. Control Signals Systems*, 2, 1989, pp. 303-314.
- [3] Funahashi K. "On the Approximate Realization of Continuous Mappings by Neural Networks", *Neural Networks*, vol. 2, 1989, pp. 183-192.
- [4] Hornik K., Stinchcombe M., White, H. "Multilayer Feedforward Networks are Universal Approximators", *Neural Networks*, vol. 2, 1989, pp. 359-366.
- [5] Vapnik V.N., Chervonenkis A.YA. "On the uniform convergence of relative frequencies of events to their probabilities", *Theory of Probability and its Applications*, vol. XVI, no. 2, 1971, pp. 284-280.