Feed Forward Networks and the Cramér-Rao Bound

W. F. Schmidt and R.P.W. Duin

Pattern Recognition Group,
Faculty of Applied Physics,
Delft University of Technology,
P.O. Box 5046,
2600 GA Delft, The Netherlands

wouter@ph.tn.tudelft.nl

Abstract

In this paper the weight space of feed forward networks will be described by a probability density function were the probability is maximum for the optimal set of weights. This probability density function is given by a property of maximum likelihood estimators and the covariance matrix of this distribution is the Cramér-Rao lower bound, a well know theorem in parameter estimation.
For certain classes of problems the optimization of the mean squared error is equal to the maximum likelihood estimator. For these problems the probability density functions is closely related to the mean squared error criterion and therefor results derived from the probability density function hold for the mean squared error surface. An analysis of the probability density function provide some theoretical understanding of the error surface and learning dynamics.

Introduction

In the field of parameter estimation we can distinguish two primary classes of applications. One class deals with situations where a precise model of the observations is present. With the measured observations the parameters of the model have to be estimated. These parameters directly reflect properties with a well defined physical meaning. In these situations the goal of the parameter estimation is to measure the actual values as accurate as possible. An example is the model of weighted sums of exponentials for radioactive decay, where the parameters are the specific decay constants and the weights represent the individual component concentrations.
The second class of parameter estimation problems are sometimes referred by *curve fitting models*. Here the parameters are not of primary interest and are often loosely connected to the actual physical process, which means they do not have a clear physical meaning. The goal of this curve fitting is often to obtain a quantitative description of the data with a small number of parameters. An example is a linear or higher order fit to calibration data of a measurement device. The calibration curve is now described by only a few numbers and interpolation between to calibration points is a straight forward computation.

If we look at the application field of the feed forward networks we can conclude that from a parameter estimation point of view we primarily deal with the second class of problems. It is highly unrealistic that the observations (the learning data) are generated by a process where weighted sigmoids are involved (see equations 6). As proven by Funahashi[1] and Hornik[2] feed forward networks can be regarded as universal approximators (if sufficient hidden units are used) and can approximate any continuous function and are therefor a very powerful curve fitting model.
However in this paper we will reformulate the curve fitting problem in such a way that we can apply a well know theorem, the Cramér-Rao lower bound, to this model to obtain more understanding about the weight space of feed forward neural networks. As a result of this approach it is possible to formulate a probability density function in weight space.
As an example two theoretical cases will be analyzed, were the observations are generated by a process were sigmoids are involved. For this situation we can completely analyze the weight space with this probalistic approach and the results obtained from these special cases will be used to formulate hypothesis about the weight space for feed forward networks used as universal approximators.

The parameter estimation problem

In this paragraph the the basic theory is presented. A good introduction to parameter estimation can be found in Sydenham[5] where most of the issues found below are discussed. However we do assume that the reader is familiar to statistical concepts as probability densities, correlation etc.

0-7803-0559-0 /92 $3.00

Suppose that we have N observations $\mathbf{w} = (w(x_1),..,w(x_N))$[1], where x_i are points in space or time where a measurement is performed. The goal is to measure a vector of parameters ρ from these observations. The function E(w) which performs this estimation on the observations w is called the *estimator* of the parameters ρ. In formula, where r denotes the estimated value of ρ, this can be written as:

$$\mathbf{r} = E(\mathbf{w}) \tag{1}$$

The observations are assumed to be generated by a model $n(x, \rho)$ but the measurements w(x) are distorted by a process v(x). This distortion is v(x) is uncorrelated with the model data $n(x, \rho)$ and is added to the observation as follows:

$$w(x) = n(x, \rho) + v(x) \tag{2}$$

Because the observations w are stochastic variables and E() is a function of these stochastic variables we must conclude that the estimated parameters r are stochastic too. The stochastic variable r has a mean value and covariance. If the estimator E() is unbiased than the expected value of r must be ρ, in formula:

$$\mathcal{E}[E(\mathbf{w})] = \rho \tag{3}$$

The operator $\mathcal{E}[.]$ is the expectation operator. The covariance of the stochastic variable r is more complicated and is known as the Cramér-Rao inequality. This inequality states that for any unbiased estimator the following expression holds:

$$\mathrm{Cov}(\mathbf{r}, \mathbf{r}) \geq (-\mathcal{E}[\frac{\partial^2 \ln L(\mathbf{w}, \rho)}{\partial \rho^2}])^{-1} \tag{4}$$

The function $L(\mathbf{w}, \rho)$ is a probability density function which describes the probability that the observations w are generated by our model with the parameters ρ. Furthermore ln() is the natural logarithm. This equation states that any estimator of r will have a variance which is larger or equal than the right hand side of this equation. This equation gives a bound on the precision with which a certain parameter can be measured, independent from the function E(w) which is chosen.
Function $L(\mathbf{w}, \rho)$ is often called the *likelihood* function. The operator $E_{ml}(\mathbf{w})$ which calculates the parameters by selecting the r which maximizes $L(\mathbf{w}, \mathbf{r})$ is called a *maximum likelihood* (ML) estimator. For maximum likelihood operators a number of nice properties are known. An important property which will be used is this article is the following.

Property of Maximum Likelihood Estimators The asymptotic probability density function of a 'broad' class of maximum likelihood estimators is Gaussian (normal) with expectation ρ and covariance equals:

$$M=(-\mathcal{E}[\frac{\partial^2 \ln L(\mathbf{w}, \rho)}{\partial \rho^2}])^{-1} \tag{5}$$

Note about this property that maximum likelihood estimators reach the lower bound of the Cramér-Rao inequality. These estimators thus achieve the best possible precision for a given parameter estimation problem. M^{-1} is called the *Fisher-information* matrix.
A reference for the mathematical prove and the precise requirements (unbiased and maximum likelihood estimator) of this property can be found in Sydenham[5] vol 1, chapter 8. In the following discussion we assume that all these requirements are met and that we may assume the above property is true.

The parameter estimation problem applied to feed forward networks

In this paragraph the theory of the previous section is applied. Here we use as an example a feed forward network with only one input, one hidden layer (h units in that layer) and one linear output unit[2]. The squashing function F() can be any monotone increasing function. In mathematical formulation our model $n(x,\rho)$ becomes:

$$n(x,\rho) = \sum_{i=1}^{h}(\alpha_i F(\omega_i x + \beta_i)) + \tau \tag{6}$$

[1] Here the parameter x is a scalar, the discussion does not change if x is a vector quantity.
[2] It is obvious how to expand this discussion to multi-layer, multi-output feed forward networks, however this would needlessly complicate the description.

In this case $\rho = (\omega_i, \beta_i, \alpha_i, \tau)$. To continue the discussion we must know the stochastic process $v(x)$. Here we assume that all the $v(x_i)$ are independent and identically distributed (iid) with a normal probability distribution $N(0, \sigma_V)$. Hence the joint probability density of $v = (v(x_1),..,v(x_N))$ is:

$$f(v, \sigma_V) = \frac{1}{(2\pi)^{N/2}} \frac{1}{\sigma_V{}^N} \exp\{\frac{\sum_{i=1}^{N} v_i{}^2(x)}{-2\sigma_V{}^2}\} \tag{7}$$

Since we assumed that $w(x) = n(x) + v(x)$ we can use this to eliminate $v(x)$ from this equation by substituting $v(x) = w(x) - n(x)$.

$$f(\alpha_i, \tau, \omega_i, \beta_i, \sigma_V) = \frac{1}{(2\pi)^{N/2}} \frac{1}{\sigma_V{}^N} \exp\{\frac{\sum_{i=1}^{N} [w(x)-n(x)]^2}{-2\sigma_V{}^2}\} \tag{8}$$

This last probability function is the likelihood function for the situation that our stochastic processes $v(x_i)$ are independent, identically and normal distributed values. The corresponding maximum likelihood estimator which maximizes this function, is in this case equivalent to minimizing the following criterion:

$$MSE = \sum_{i=i}^{N} [w(x_i)-n(x_i)]^2 \tag{9}$$

This is the well known mean squared error criterion which is for example used as criterion function in the *back propagation* rule (see Rumelhart[4]). In the following discussion it is not important how this optimization is done, only that we have a method which is capable of finding the minimum of equation 9. However in practice it is very complicated to find the global minimum and in this article we will make some comments on the problems involved when using for example the back propagation algorithm.

Because the parameters are estimated by using the maximum likelihood estimator for ρ, we know (from the property of these estimators) that we have an unbiased estimator (mean value is ρ), with covariance matrix as predicted by Cramer-Rao and with known statistical distribution. The probability density function of **r**, the weight space, is now known. (Remember the property of maximum likelihood estimators.)
You can formulate this conclusion also as follows. Suppose we want to find out how all the possible weight combinations are distributed in weight space. From the function we want to approximate, we assume that it must be a continuous function. Funahashi[1] and Hornik[2] have proven that it is possible to approximate this with our model (equation 6) provided that we have a sufficient number of hidden units. Assume we know that we need H hidden units to do this.
At this point we propose that our observations are generated by a process which consists of a weighted sum of H hidden units. We want to approximate our function but with a certain freedom, namely it is not required that all the data points are approximated exactly, but they deviate from the true value with a given probability density. This probability is given by the probability density of equation 7 and we call this deviation $v(x_i)$. The deviation is controlled by the σ_V parameter. By using equation 7 we assume that the deviations at every point are not correlated with each other.
If we estimate the weights by optimizing the mean square error criterion we know that for this problem the likelihood function is maximized also. Now we can apply the maximum likelihood property which states that the weights are distributed with a multi normal distribution around a certain mean value with a covariance given by the matrix M. The set of weight combinations which may be a valid solution for our network, with a certain probability larger than p, all lay in one hyper ellipsoid. The size of this hyper ellipsoid is determined by the matrix M.

Deriving properties from the Cramér-Rao covariance matrix

In this section we will discuss some properties we can derive from the Cramér-Rao covariance matrix and the implications for the learning process. A full derivation of this matrix is omitted here because it can easily be obtained by calculating the Hessian matrix of the natural logarithm of equation 8.

Diagonal elements of M
The diagonal elements of matrix M are the covariances of the individual parameters and are measures for the interval in which every parameter will vary. These parameters do not directly influence learning performance.

The hyper ellipsoid volume |M|
The determinant value of M is a measure for the volume spanned by the probability function. A larger volume implies a larger volume in weight space, for all weight combinations that may have generated our data, with a probability larger than p (remember that matrix M is the covariance matrix of the probability density function in weight space). A large volume would suggest that a learning rule can more easily find an acceptable solution for the problem. This suggests that a certain problem can be learned faster.

Parameter correlation
The correlation between the parameters is defined as:

$$C_{ij} = \frac{M_{jj}}{\sqrt{M_{ii} * M_{jj}}} \tag{10}$$

The well known correlation coefficient measures the linear tendency between two parameters and $|C_{ij}|$ is approximately 1 if a linear relationship exists. A $|C_{ij}|$ value near 0 suggests that this relationship does not exist. If a large number of parameters are highly correlated, this is an indication that the solution space is 'banana' shaped and the back propagation method as a simple gradient descent method will follow a more and more zigzag path towards the minimum (see Press e.a[3] page 319 or Widrow and Stearns[6] chapter 4 and 5).

Eigenvalues of M
Matrix M specifies the sensitivity of the parameters with respect to the measurement procedure. In the case under investigation the parameters are calculated by optimizing the mean squared error and therefore a parameter with a small covariance value implies a large sensitivity with respect to the mean squared error. However a parameter with a large covariance implies a small influence to the mean squared error.
The matrix M is symmetric and all the eigen values and eigen vectors exists and are real, furthermore the eigen vectors define an orthogonal basis. This orthogonal basis defines the principal directions with respect to this sensitivity of the parameters and the eigen vectors point in the direction of all these directions. Because of the direct relationship between the mean squared error and the covariance matrix M the eigen values and eigen vectors of this matrix also define the principal direction in mean squared error space.

It is know that steepest descent performs poor in very banana shaped valleys and the largest and second largest eigen value (λ) give an impression of the shape (see Press e.a [3]).

$$S_{rel} = \left| \frac{\lambda_{max}}{\lambda_{max-1}} \right| \tag{11}$$

Furthermore for stable learning the leaning rate of steepest descent is limited by the largest eigenvalue and the learning speed is limited by the smallest eigenvalue therefore the quotient of the largest and smallest eigen values is of importance with respect to the back propagation learning. (Note that a large value of S_{rel} or a large value of S_{abs} indicate a slow learning for gradient descent methods)

$$S_{abs} = \left| \frac{\lambda_{max}}{\lambda_{min}} \right| \tag{12}$$

Finally we want to mention that the eigen vectors itself are interesting. A number of these vectors will point to directions which are probably very insensitive with respect to the mean squared error.

Examples and Results

Because matrix H is dependent on the true value ρ, the type of squashing function used and on the values of x, we will investigate some quantitative properties of this matrix for a specific problem. In our example we use the following sigmoid model:

$$n_{example}(x) = F(x+1) - F(x-1) \tag{13}$$

Here the squashing function is the following sigmoid:

$$F(x) = \frac{1}{1 + \exp(-x)} \tag{14}$$

From our model 128 observations are measured. For the first example these values are selected from a x_i range between [-10..10] (see figure 1) and for the second example these values are selected between [-1..1] (see figure 2).

Now all the relevant parameters are set we can calculate the corresponding M matrix. The properties mentioned in the previous paragraph are calculated. Note that from α, ω, β the results of only one parameter is listed.
In the previous paragraph we mentioned that the hyper elliptic volume and the S_{shape} and S_{speed} parameters are important for the learning method. From table 1 we see that the hyper elliptic volume is much larger for the second example and therefore learning example 2 until a certain mean squared error is reached should be much easier.
However the S_{shape} and S_{speed} (table 2) show that gradient descent methods will have more problems because the error space has become more banana shaped, which results into slower learning. Combining these results we could make the hypothesis that the gradient descent method will take probably much longer to learn example 2 than example 1 due to the bad shape of the error surface. A higher order method, which uses more shape information, is probably much faster for example 2 because the volume of acceptable weight solutions is much larger.

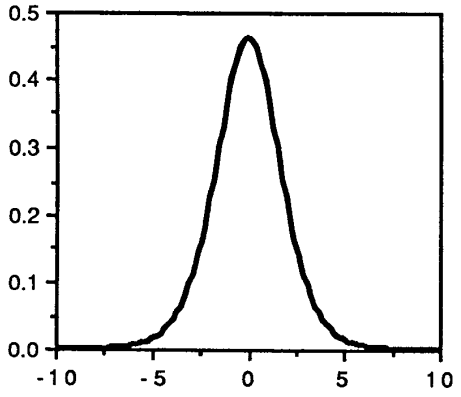

Figure 1 Curve of example 1

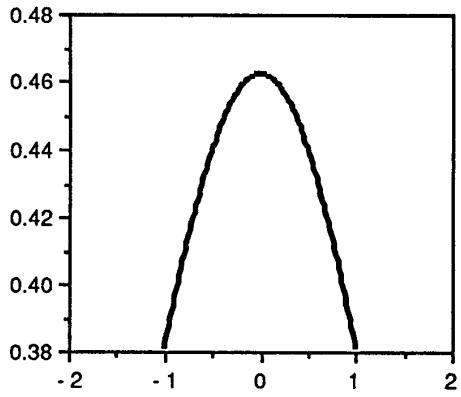

Figure 2 Curve of example 2

Table 1 Covariance, correlation and hyper ellipse volume calculated for both experiments

	Experiment 1	Experiment 2
Cov(α, α)	535 σ_v^2	123 10^6 σ_v^2
Cov(τ, τ)	0.04 σ_v^2	160 10^4 σ_v^2
Cov(ω, ω)	21.3 σ_v^2	132 10^5 σ_v^2
Cov(β, β)	736 σ_v^2	980 10^5 σ_v^2
Corr(α, τ)	-0.359	-0.985
Corr(α, ω)	-0.887	-0.991
Corr(α, β)	-0.977	-0.999
Corr(τ, ω)	0.532	0.996
Corr(τ, β)	0.407	0.986
Corr(ω, β)	0.954	0.997
Hyper ellipse volume	7.00 σ_v^{14}	1.92 . 10^{19} σ_v^{14}

Table 2 Speed and shape parameters calculated for both experiments.

	Experiment 1	Experiment 2
S_{shape}	34.1	629
S_{speed}	539 . 10^3	152 . 10^9

Table 3 Average number of training cycles to learn the examples until MSE is 10^{-4}.

Method	Example 1 (# cycles)	Example 2 (# cycles)
Back propagation	49	2.6 . 10^5

This hypothesis is verified with the following experiment. Example 1 and 2 are learned by standard back propagation (see Rumelhart[4]) by using a network mentioned previously in equation 13. The initial weights are chosen randomly from a uniform distribution between -0.01 to 0.01. The momentum term is set to 0.0 and a learning rate of 0.5 is used. The network is trained until a mean squared error of 10^{-4} is reached and the number of cycles needed to train this network is measured. This procedure is repeated 100 times and the average number of cycles is calculated. The same experiment is done for the second example. The results can be found in table 3. The results found in table 3 show that example 2 needs a significant larger number of cycles to learn the desired input - output relationship. For example 2 the volume of acceptable weights is predicted to be much larger but the shape parameters of table 2 show to be much worse for this example. This result shows that the shape of the error surface is much more important to the learning speed than the predicted volume, for back propagation learning.

Conclusions and Discussion

The main result from this analysis is that it is possible to formulate the approximation of a function by a feed forward network in such a way that we can apply the Cramér-Rao theorem. The Cramér-Rao theorem together with the properties of maximum likelihood estimators determine a probability density function in weight space. In this approach the optimal weights are determined by the maximum of this function.
From the Cramér-Rao theorem and the corresponding covariance matrix M we learn that the weights space shape is determined (at least near the global minimum) by the optimal weights, number and position in space or time of the data points and the squashing function used (evaluate matrix M). The influence of all these individual contributions are complex, because M is dependent in a highly non linear fashion to these contributions .

For real world problems this analysis is of limited use because the number of hidden units and the optimal weights are not known. A special case can be investigated, as example 1 and 2 in this article. From these examples we see that the volume measure (see table 1) has limited influence on the learning speed. The error surface shape largely dominates the learning speed for the back propagation algorithm. This is a well known property of gradient descent (see Press[3]) techniques.

Interesting for future research is the influence of a scaling of the inputs on the shape parameters S_{rel} and S_{abs}. Scaling the inputs implies that the parameters ω_i and β_i are scaled inversely to compensate for the input scaling. The influence of this scaling on the error surface is complicated to predict as concluded earlier. A similar discussion holds for an optional scaling of the outputs.
An other interesting aspect is the analysis of the principal directions as defined by the eigen vectors of M, with respect to the mean squared error. The relationship between a small deviation from the global minimum to one of these directions and the resulting change in the mean squared error is interesting to plot to see if there are redundant solutions is weight space.

Acknowledgement

This research is supported by the Dutch Government as part of the SPIN/FLAIR DIAC project.

References

[1] K. Funahashi, "On the Approximate Realization of Continues Mappings by Neural Networks.", Neural Networks, Vol 2 page 183-192, 1989.

[2] K. Hornik, "Multilayer Feedforward Networks are Universal Approximators", Neural Networks, Vol 2 page 359-366, 1989.

[3] W.H. Press, B.P. Flannery, S.A. Teulkolsky, W.T. Vetterling, *"Numerical Recipes in C"*, Cambridge University Press, 1988.

[4] D.E. Rumelhart, G.E. Hinton and R.J. Williams, "Learning internal representations by error propagation", Parallel Distributed Processing: *"Exploring in the Microstructure of Cognition"*, Vol 1, D.E Rumelhart and J.L. McClelland (Eds.), Cambridge, MA: MIT Press, pp 318-362.

[5] P.H. Sydenham, *"Handbook of Measurement Science"*, Vol 1, Chapter 8, John Wiley & Sons, 1982.

[6] B. Widrow and S. Stearns, *"Adaptive Signal Processing"*, Prentice Hall Inc, Englewood Cliffs, New Jersey, 1985.