Tina Memo No. 2010-006
Internal.

# Solving the Bias-Variance Problem during Network Training.

N.A. Thacker

Last updated
5/2/2018

Imaging Science and Biomedical Engineering Division,
Medical School, University of Manchester,
Stopford Building, Oxford Road,
Manchester, M13 9PT.

# Solving the Bias-Variance Problem during Network Training.

N. A. Thacker 12/12/09

## Abstract

*The following document outlines a theoretical approach to constructing an optimisation function for use in neural network training which could be used to solve the bias-variance dilemma, and thereby achieve optimal generalisation. The idea is rooted in quantitative use of probability and results in a cost function which embodies a form of orthogonal regression. A brief comment regarding biological plausibility is included. These ideas need development and testing, either as a modified EM algorithm or a neural network optimisation function.*

## Introduction

Artificial neural networks can be thought of as a high dimension interpolation function which maps an input vector $\mathbf{x_i}$ to an output $\mathbf{o}(\mathbf{x_i})$. Many supervised training algorithms are based upon optimising the difference between the output and some target for $N$ data samples.

$$\chi^2 \;=\; \sum_i^N \; |t_i \;-\; o(\mathbf{x_i})|^2 \qquad\qquad -(1)$$

and this is indeed the origins of training algorithms such as "Back-Propagation", which can be shown to minimise this quantity via a process of gradient decent. Although it is common to speak of the general validity of such measures for all data (BLUE, best linear unbiased estimator), strictly this measure is only *quantitatively* valid if our training data has the property of uniform independent random Gaussian noise. The cost function can then be directly interpreted as a log Likelihood. Although data can generally be linearly transformed if this is not already the case.

Alternatively networks can be trained using a "cross-entropy" function of the form

$$E \;=\; \sum_i^N \; t_i \; log(o(\mathbf{x_i})) \;+\; (1 - t_i) \; log(1 - o(\mathbf{x_i})) \qquad\qquad -(2)$$

The relationship of a chi-square function to Likelihood is well known, but what about equation (2)?

Generally the output function will be constrained by the network architecture to be smoothly varying over some volume of $\mathbf{x}$. Over this volume the output will be effectively constant $o$ for some subset of data $N_s$. A partial contribution to the overall cost function for this subset can be written as

$$E_s \;=\; \sum_i^{N_s} t_i \; log(o) \;+\; (1 - t_i) \; log(1 - o) \qquad\qquad (2)$$

We wish to understand how $o$ is related to the original training data $t_i$ and we can do this by determining the choice of $o$ which minimises $E_s$. Setting, $\sum_i^{N_s} t_i \;=\; N_c$ i.e. the number of times the target class label was 1 in the selected volume, we have

$$\frac{\partial E_s}{\partial o} \;=\; \frac{N_c}{o} \;-\; \frac{(N_s - N_c)}{(1 - o)} \;=\; \frac{N_c \;-\; N_s o}{o(1 - o)}$$

Assuming that $o(1 - o) \;\neq\; 0$ , this expression is zero ($E_s$ is a minimum) when

$$o = N_c/N_s$$

so that the overall cost function is minimised when, in all locations in the space, the output from the network is the local fraction of positive target labels; in other words the **conditional probability** $P(t|\mathbf{x_i})$. Interestingly, the same analysis can be made of a $\chi^2$ cost function and the same conclusion is reached. The binary targets $t_i$ can also be relaxed to be real or "soft" labels, in accordance with our definition for $N_c$.

By minimising either (1) or (2), assuming that the solution is reachable with the available degrees of freedom of the network, we are therefore ensuring

$$o(\mathbf{x_i}) \;\approx\; P(t|\mathbf{x_i}) \qquad\qquad -(3)$$

and if the output is used to attribute class assignment, it will attain Bayes optimal performance over a set of test data, provided the prior probabilities of classes are equivalent in both training and test data sets, and at all locations in the data space[1].

Finally, as $P(t|\mathbf{x_i}) \propto P(\mathbf{x_i}|t)$, when using equation (2) as with (1) we are also minimising a log likelihood of obtaining the data. The constant of proportionality, being fixed here, plays no role.

## Model Selection

The problem with neural networks is that the equivalent function $\mathbf{o}(\mathbf{x_i})$ can have arbitrary complexity, practically realised as a variable number of hidden nodes and connections. If all we seek to do is minimise a likelihood function (such as (1) or (2)) then we can eventually obtain a value of 0 for any finite quantity of data, if we are prepared use use enough free parameters (network weights). This has been referred to by the artificial neural network community as the bias-variance dilemma, as at some point we need to establish a trade-off between the residual bias on our attempts to predict $\mathbf{x_i}$ using $\mathbf{o}(\mathbf{x_i})$ and the stability (or variance) of any estimate.

This community established long ago that the ultimate aim of a neural network is therefore not to simply optimise a likelihood, but prediction, specifically "generalisation" to unseen data. All trained neural network architectures are therefore tested on unseen data in order to establish utility. Networks with the best prediction capabilities are chosen for each application, thereby also determining the best architectures (and numbers of weights). However, this process is rather hit and miss. It would be better to be able to train the network so that it optimises generalisation directly. This has been seen as the "holy grail" for research in this area. Many researchers may have concluded that it is equally unattainable.

In order to stay close to the original work, which performed analysis of density function in the original data space $\mathbf{x}$, rather than considering a network classifier I will start by considering counter-prorogation. This has uses for tasks such as dimensionality reduction. When done properly it can have uses for finding low dimensional descriptions within complex data, and also noise filtering. It can also be used as a pre-cursor to classification.

For counter-propagation the training target is the $M$ dimensional input $\mathbf{x_i}$. This can be achieved by minimising a function of the form

$$E = \sum_i^N |\mathbf{x_i} - \mathbf{t}(\mathbf{x_i})|^2 \qquad -(4)$$

In previous work, we concluded that the best generalising function could be obtained by computing the overlap between the original data *pdf* $p(\mathbf{x})_n$, and an estimate of the *pdf* for prediction of unseen data. The process required several stages in order for the distribution of predicted outputs to be quantitatively valid. This incorporated both the expected measurement noise, the effects of interpolation and the instability in parameters due to the noise on the original data. Uncertainty due to functional interpolation was found to broaden the pdf in a manner equivalent to convolution with a unit variance distribution lying in the plane of the interpolation function $\otimes C_f$. Stability of estimated parameters was estimated using an analytic form of "leave one out" cross validation in order to obtain an unbiased estimate $C_p$. The level of agreement between noisy data $p_n$ and generalising function was evaluated in the original data space using the Bhattacharrya measure. Later work derived from first principles as the correct method for comparing probability densities. The overall comparison was therefore of the form

$$B = \int \sqrt{p(\mathbf{x})_n.(p(\mathbf{x})_n \otimes C_f \otimes C_p)}\, d\mathbf{x} \qquad -(5)$$

Where $p_n \otimes C_f \otimes C_p$ is an estimate of the probability density of the predicted output for an independent sample of data, drawn from the same distribution as the original data set. Notice that best agreement is achieved when contributions from $C_f$ and $C_p$ are zero ,ie: the functional interpolation predicts all dimensions of the data (rather than constraining a lower dimensional manifold) and the parameters are estimated using infinite quantities of data. Notice that this approach to model selection is based upon the use of quantitative probability, and not 'priors' (as would be found in Bayesian methods) or information theory. It therefore does not sit well with many of the popular theories regarding the origins of Likelihood.

## Analysis

Although the above theory was shown to be quantitatively valid and would estimate correctly the parameters which achieved optimal interpolation of polynomials, no work was done to extend the ideas to cover more conventional

[1] Whilst this sounds impressive it is also quite a significant restriction on optimal use.

estimation techniques such as (4). The functional form of (5) is not particularly convenient as the basis for an algorithm, with most of the difficulty arising due to the presence of the square-root. It is clear however, that we could attempt to replace the Bhattacharyya comparison with a more conventional cost function.

Making the observation that unseen data should be drawn from the same distribution which we are assuming for the noise process, we can say that it should be possible to construct an objective function from a set of perturbations of the target data, each of the form

$$E_s \ = \ \sum_i^N |\mathbf{x_i} \ - \ \mathbf{t}(\mathbf{x_i} + \boldsymbol{\delta})|^2 \qquad - (6)$$

where $\mathbf{x_i}$ is the noise free generator of the data and $\boldsymbol{\delta}$ is a random variable drawn from the expected distribution[2]. In this form $E_s$ is stochastic, but we can turn this into an analytic expression by evaluating the expectation of the cost function.

$$< E_s > \ = \ \sum_i^N \int p_{\boldsymbol{\delta}} |\mathbf{x_i} \ - \ \mathbf{t}(\mathbf{x_i} + \boldsymbol{\delta})|^2 \ d\boldsymbol{\delta}$$

with the normalised probability density

$$p_{\boldsymbol{\delta}} \ = \ \alpha \ exp(-|\boldsymbol{\delta}|^2/2)$$

To proceed further we need a specific form for $\mathbf{t}(\mathbf{x_i})$, however if the noise is expected to be small in comparison to the underlying non-linearity of the underlying interpolative function, we can write

$$\mathbf{t}(\mathbf{x_i} \ + \ \boldsymbol{\delta}) \ \approx \ \mathbf{t}(\mathbf{x_i}) \ + \ \nabla_{\mathbf{x}} \mathbf{t} \boldsymbol{\delta}$$

where $\nabla_{\mathbf{x}} \mathbf{t} \boldsymbol{\delta}$ is an element by element multiplication, ie: a first order Taylor expansion. Then

$$< E_s > \ \approx \ \sum_i^N \ [\int p_{\boldsymbol{\delta}} |\mathbf{x_i} - \mathbf{t}(\mathbf{x_i})|^2 d\boldsymbol{\delta} \ - \ 2(\mathbf{x_i} \ - \ \mathbf{t}(\mathbf{x_i} + \boldsymbol{\delta})) \ . \ \int p_{\boldsymbol{\delta}} (\nabla_{\mathbf{x}} \mathbf{t} \ \boldsymbol{\delta}) \ d\boldsymbol{\delta} \ + \ \int p_{\boldsymbol{\delta}} |\nabla_{\mathbf{x}} \mathbf{t} \ \boldsymbol{\delta}|^2 \ d\boldsymbol{\delta}]$$

Clearly

$$\int p_{\boldsymbol{\delta}} |\mathbf{x_i} - \mathbf{t}(\mathbf{x_i})|^2 \ d\boldsymbol{\delta} \ = \ |\mathbf{x_i} - \mathbf{t}(\mathbf{x_i})|^2 \int p_{\boldsymbol{\delta}} \ d\boldsymbol{\delta} \ = \ |\mathbf{x_i} - \mathbf{t}(\mathbf{x_i})|^2$$

$$\int p_{\boldsymbol{\delta}} (\nabla_{\mathbf{x}} \mathbf{t} \ \boldsymbol{\delta}) \ d\boldsymbol{\delta} \ = \ 0$$

and

$$\int p |\nabla_x t \ \boldsymbol{\delta}|^2 \ d\boldsymbol{\delta} \ = \ |\nabla_{\mathbf{x}} \mathbf{t}|^2$$

Therefore

$$< E_s > \ = \ \sum_i^N |\mathbf{x_i} \ - \ \mathbf{t}(\mathbf{x_i})|^2 \ + \ \sum_i^N |\nabla_{\mathbf{x}} \mathbf{t}|^2 \qquad - (7)$$

In words, this is equivalent to saying that we have corrected the original definition of the cost function to add back in the expected amount of variance which was lost within the manifold defined by the interpolating function $\mathbf{t}(\mathbf{x_i})$.

This is only a first order approximation, by considering this issue to second order we would obtain the rank of the correlation matrix for $\mathbf{t}(\mathbf{x_i})$ at $\mathbf{x_i}$, ie: the local dimensionality (the assumption of uniform homogenous noise simpifies things here). This is analogous to an Akaike style information correction to a chi-square cost function. However, an Akaike correction is a 'global' average estimated from all of the data and therefore unable to directly influence 'local' parameter estimation (see below). In the above approach such a correction is derived separately for each data point, which is required if the intrinsic dimensionality of the interpolation function can vary from location to location across $\mathbf{x}$.

Equation (7) defines an overall cost function which minimises an estimate of the variance in the data not accounted for by the interpolative model, ie: the random noise. It is therefore useful to consider the cost function as arising from two orthogonal components of variation. Those lying within the functional interpolation, and those perpendicular to this.

[2] Others have suggested using random noise on both input and output to boost the apparent quantity of training data, here we only add it during the forward pass of the network, in order to assess the degree of match to independent data.

# Discussion

Unlike the original work, the methodology thus far makes no attempt to account for parameter stability. We know that this process destabilises the parameters and thereby the predicted noise distribution in a direction perpendicular to the interpolation manifold. In fact, this is precisely the effect accounted for by a multiplicative factor of two in the Akaike correction.

For (7), the more complex the local model becomes, the lower the dimensionality of the noise component it can remove. This is present in the behaviour of (7) in a way that (4) does not capture. In fact, (4) will always miss variations within the functional constraint manifold, even though data cannot be usefully constrained (and noise removed) in these directions. This may already be enough to drive the estimation of parameters.

The above analysis suggests that it may be possible to design a neural network training algorithm which automatically compensates for the local intrinsic dimensionality of the data during training, in order to achieve optimal noise suppression and thereby optimal generalisation to any data drawn from the same noise distribution as the training data.

If this method has not already been published elsewhere, these ideas now need to be checked. The modified cost-function can be tested directly, but more theoretical work would be needed to derive a modified gradient based training algorithm. Although preliminary work could be done in Monte-Carlo, publication in a journal will probably require a suitable real-world problem with which to demonstrate practical benefit.

Note that in a biological system, (6) (the very definition of what we believe constitutes an un-biased estimate of performance) is implicitly defined via the process of introducing a temporal delay, so that any learning algorithm works so as to predict independent temporal samples of the same input data. This would simulate the effects of random perturbation via the process of repeated measurement. Static neural networks are an artificial concept, and real biological systems are stochastic and temporal, they will automatically get this type of training for free. A long standing problem in the field of Artificial Neural Networks may therefore simply not be an issue when computations are performed in neuronal tissue.