D4.5

A PROBABILISTIC APPROACH TO THE UNDERSTANDING AND TRAINING OF NEURAL NETWORK CLASSIFIERS

Herbert Gish
BBN Systems and Technologies Corporation
Cambridge MA 02138

ABSTRACT

It is shown that training a neural network using a mean square error criterion gives network outputs that approximate posterior class probabilities. Based on this probabilistic interpretation of the network operation, we investigate information-theoretic training criteria such as Maximum Mutual Information and the Kullback-Lielbler measure. We show that both of these criteria are equivalent to the Maximum Likelihood (ML) estimation of the network parameters. ML estimation of a network allows for the comparison of network models using the Akaike Information Criterion and the Minimum Description Length criterion.

I Introduction

We first investigate the training of neural networks with the commonly used mean square error (mse) criterion and demonstrate a connection between neural network operation and the approximation of posterior class probabilities. This connection is an aid to understanding how the network will behave as a classifier and has implications regarding the use of the network as a probability estimator in a complex application. We pursue the probability connection of the neural network which leads us to consider, information-theoretic training criteria as alternatives to the mse criterion. In particular we consider network design using the Maximum Mutual Information (MMI) criterion. We then show that the neural network designed with this criterion is equivalent to a network designed by minimizing the Kullback-Liebler (K-L) distance to the posterior class probabilities. We finally investigate the estimation of the neural network parameters by the method of Maximum Likelihood (ML). Quite remarkably, we find that the ML estimate is equivalent to the MMI estimate and the K-L based estimate. A significant advantage of the ML approach is that its asymptotic properties enables us to compare the performance of neural networks taking into account the diffence in the number of parameters.

II Implications of Mean Square Error Training

II.1 Approximation of Posterior Class Probabilities

Let us represent the neural network by the function $f(x,\theta)$ where x is the input feature vector and θ represents the values of all the parameters that define the network. We will initially consider the two class problem for which the desired output of the neural network takes on the value a if x is in class C_1 or the value b if x is in class C_2. The criterion by which the performance of the network is measured is given by the mean squared error,

$$E = \frac{1}{N}\left[\sum_{x\epsilon C_1}[f(x,\theta)-a]^2 + \sum_{x\epsilon C_2}[f(x,\theta)-b]^2\right] \quad (1)$$

where N is the total number of training samples.

If we assume that N is large and the number of samples from each of the classes is in proportion to the *a priori* probability of class membership, we can approximate the summations in E by integrals:

$$E \approx \int[f(x,\theta)-a]^2 p(x,C_1)dx + \int[f(x,\theta)-b]^2 p(x,C_2)dx \quad (2)$$

where $p(x,C_i)$ $i = 1,2$, is the joint probability density function of the observations and that they are from class C_i. The previous equation can be rewritten as

$$\begin{aligned} E &= \int f^2(x,\theta)[p(x,C_1)+p(x,C_2)]dx && (3)\\ &- 2\int f(x,\theta)[ap(x,C_1)+bp(x,C_2)]dx \\ &+ a^2\int p(x,C_1)dx + b^2\int p(x,C_2)dx. \end{aligned}$$

If we note that the unconditional probability of an observation is given by

$$p(x) = p(x,C_1) + p(x,C_2) \quad (4)$$

and define the function

$$\begin{aligned} d(x) &= \frac{ap(x,C_1)+bp(x,C_2)}{p(x)} && (5)\\ &= aP(C_1|x) + bP(C_2|x) && (6) \end{aligned}$$

where $P(C_i|x)$ is the posterior probability of class C_i occurring given the observation x, we obtain, with $P(C_i)$ the prior probability of class C_i

$$\begin{aligned} E &= \int[f(x,\theta)-d(x)]^2 p(x)dx && (7)\\ &+ a^2P(C_1) + b^2P(C_2) - \int d^2(x)p(x)dx. \end{aligned}$$

Note that only the first term in the above equation depends on the parameters of the network and therefore *adjusting the parameters of $f(x,\theta)$ to minimize E is equivalent to minimizing the mean square error between the network output and $d(x)$.*

CH2847-2/90/0000-1361 $1.00 © 1990 IEEE

When we choose $a = 1$ and $b = 0$ as the desired network outputs we obtain from Equation 6

$$d(x) = P(C_1|x) \tag{8}$$

which is the posterior probability for class C_1, i.e., the probability that class C_1 has occurred given that x has been observed. From Equation 7 we see that this is what the network is being trained to approximate in a weighted mean square sense. The weighting being, $p(x)$, the unconditioned density of the feature vectors being classified. Note that the optimal (minimum average error) Bayes classifier assigns a feature vector to class C_1 if $P(C_1|x) \geq 0.5$ otherwise class C_2 is chosen.

The result that we have derived is valid for any network and just depends on the mse criteria. How small the mse can be made will, of course, depend on the structure of the network. However, if we allow the network to be arbitrarily complex, then we can have the error go to zero and the network output will converge to the posterior class probability. This special case is a result derived by Bourlard and Wellekens [1].

II.2 Absolute versus Relative Error

In speech processing, and other applications, the output of the neural network will not only be used to classify individual feature vectors but will also be used as a probability estimate of an event that is to be combined with other probability estimates of other events. These combinations are typically obtained by adding the logorithms of these probability estimates to form the log probability estimate of some joint event. Absolute errors in the log probability of the joint event will depend on the *relative* errors made in the estimation of the probabilities of the individual events that were combined. A relative error is the size of the error normalized by the value of what is being estimated. Also note that a small, absolute error, made in estimating a small probability can result in a large relative error. Unfortunately the mean square error criterion deals with absolute errors and not relative errors. However, we will see below that the Kullback-Liebler measure is a function of the relative error.

II.3 The Multi-Class Case

In many applications a neural network is designed to discriminate between M classes ($M > 2$). In this application the network will have M[1] outputs $f_i(x, \theta)$,$i = 1, \ldots, M$ where the output values are in the range [0,1]. The desired output of the neural network will be 1 for the class to which the input training sample belongs and 0 for all the other outputs. Following the derivation for the two-class case given previously, we find that minimizing the square error criterion is equivalent to minimizing

$$E = \sum_{i=1}^{M} \int [f_i(x, \theta) - P(C_i|x)]^2 \; p(x)dx \tag{9}$$

This shows that the parameters of network are being used to *simultaneously* approximate M different functions, the posterior class probabilities, such that the average of the squared errors is minimized, where the average is taken over all the training data.

Although the output of the network is approximating the posterior class probabilities it will not be a true probability since in general,

$$\sum_{i=1}^{M} f_i(x, \theta) \neq 1 \tag{10}$$

This presents a problem for any classifier that wants to use these outputs as probabilities, e.g., in combining the outputs to estimate the probability of a sequence of events, as in continuous speech recognition.

The problem of the network outputs not summing to unity requires that the optimization problem be changed to one of constrained optimization. Note that simply normalizing the outputs of the network to sum to one after training will result in outputs that are probabilities, but this will not be the least squares solution to the problem and is simply an ad hoc remedy. We can overcome this problem by altering the network in order to constrain the probabilities to be normalized. We simply must add a layer to the existing network to perform the normalization, i.e., if $f_i(x, \theta), i = 1, \cdots, M$ are the unnormalized outputs of the neural network, we instead have the outputs be

$$f_i'(x, \theta) = \frac{f_i(x, \theta)}{\sum_{j=1}^{M} f_j(x, \theta)} \tag{11}$$

It is $f_i'(x, \theta)$ that needs to be trained and this will result in network outputs that have the appropriate normalization. This modification to the network does increase the complexity of the training process and training of the network with the new normalized outputs amounts to a constrained least squares estimation of the network parameters.

III The Network as an estimator of $P(C|x)$

We have shown that a neural network trained with the mse criterion gives us a network that provides an approximation to the posterior class probabilities, $P(C|x)$. How good an approximation can be achieved depends on the training data and also on the structure of the network. In the remainder of the paper we will view the network as a mechanism for esimating posterior class probabilities. This primarily means that all network outputs are in the range [0,1] and the outputs sum to unity if the number of outputs equals the number of classes or sums to less than unity if the number of outputs is one less than the number of classes. For the sake of clarity we will only consider the two class problem although our results in the remainder of the paper can be readily extended to more than two classes. Below we consider some properties of a network viewed as a probability esitimator.

[1] For M > 2 classes it is more convenient to use M rather than M-1 outputs

III.1 Basic Modeling Properties of the Network

We can write the output of the network as a sigmoidal function of its input,

$$f(x,\theta) = \frac{1}{1+e^{-z(x,\theta)}} \tag{12}$$

where $z(x,\theta)$ is the input to the sigmoid at the output layer. If we invert the sigmoid we obtain

$$z(x,\theta) = \log \frac{f(x,\theta)}{1-f(x,\theta)} \tag{13}$$

Now let us write

$$f(x,\theta) = P(C_1|x). \tag{14}$$

From Equations 13 and 14 and the fact that $P(C_2|x) = 1 - P(C_1|x)$ we can write,

$$z(x,\theta) = \log \frac{P(C_1|x)}{P(C_2|x)} \tag{15}$$

$$= \log \frac{p(x|C_1)}{p(x|C_2)} + \log \frac{P(C_1)}{P(C_2)} \tag{16}$$

The above equations show that the neural network is in effect modeling the log likelihood ratio of the two classes. The direct modeling of the ratio allows the network to be efficient in its use of parameters since each of the pdfs is not modeled separately as is done with other types of classifiers. In the statistical literature this type of model is called logistic regression function and $z(x,\theta)$ is usually a linear function of the coordinates of the feature vector, i.e., no hidden layers.

Also note that the unconditional pdf of the observations is

$$p(x) = p(x|C_1)P(C_1) + p(x|C_2)P(C_2) \tag{17}$$

$$= p(x|C_1)P(C_1)\left[1+e^{-z(x,\theta)}\right] \tag{18}$$

where we have solved Equation 16 for $p(x|C_2)$ and substituted it in Equation 17. Since $p(x|C_1)$ has not been specified $p(x)$ is not constrained by the selection of θ. This point becomes important when we consider ML estimation of the model parameters.

IV Training with Information-Theoretic Criteria

In this section we rely on the interpretation of the network output as a true probability function to develop a design criterion for neural networks. We measure network performance by the amount of information it provides about the true class of the input feature vector using Shannon's mutual information. In training the network the goal is to maximize this mutual information. This is known as the Maximum Mutual Information criterion (MMI) (which has been used sucessfully in speech applications [2]; also see [3] for independently obtained, MMI-neural network results). We also look at the network performance in terms of the "distance" of the neural network estimate of the posterior class probability from the true class probability using the Kullback-Liebler information measure. Both criteria lead to the same network. We only consider the two-class problem but it readily generalizes to an arbitrary number of classes.

IV.1 The MMI Criterion

First let us consider, in general terms, what mutual information is measuring. The mutual information, $I(C,X)$, between two events C and X is a statistical measure of the amount of information the occurrence of each event contains about the other. One way of writing mutual information is

$$I(C,X) = H(C) - H(C|X) \tag{19}$$

where H represents entropy, which is a measure of uncertainty in the occurrence of a particular event. The above equation states that the mutual information between C and X is the uncertainty in C minus the amount of uncertainty about the occurrence of C when Y is already known. In the extreme cases, if observing X gives no information about C then $H(C|X) = H(C)$ and the mutual information between C and X is zero; if observing X gives complete knowledge about X, then $H(C|X) = 0$ and the mutual information between C and X is $H(C)$.

Consider the problem of classifying observations, x, into either class C_1 or class C_2. We are interested in the mutual information between the class and the observations which is given by

$$I(C;X) = H(C) - H(C|X) \tag{20}$$

where

$$H(C) = -\sum_{i=1}^{2} P(C_i) \log P(C_i) \tag{21}$$

and

$$H(C|X) = -\int p(x)\left[\sum_{i=1}^{2} P(C_i|x) \log P(C_i|x)\right] dx. \tag{22}$$

The first term in Equation 20 is the *a priori* uncertainty as to which class will occur and the second term, sometimes referred to as the equivocation, is the uncertainty as to which class occurred after the observation has been made. The greater the mutual information between classes and obsevations the more information we have about which class has occurred given an observation.

In the training of the neural network under the MMI estimation criterion we have the network outputs, $f_i(x,\theta)$ $i = 1,2$, with $f_1(x,\theta) = 1 - f_2(x,\theta)$, as our estimates of the posterior class probabilities $P(C_i|x)$ $i = 1,2$, respectively. With the network outputs as our estimates of the posterior probabilities we can form an estimate of the mutual information between the classes and the observations. It is this estimate that we try to maximize by adjustment of the network parameters, θ. Maximizing the mutual information, however, is equivalent to maximizing the negative of the equivocation term and therefore we only have to use the network outputs to estimate the negative of this term. The negative of the equivocation term is the negative of Equation 22 which (after noting that $P(C_i|x) = p(x|C_i)P(C_i)/p(x)$) can be expressed as

$$-H(C|X) = \sum_{i=1}^{2} p(x|C_i)P(C_i) \log P(C_i|x). \tag{23}$$

Observe that the above equation is the expectation of $\log P(C_i|x)$ over each of the classes. Now to obtain our sample estimate of the negative of Equation 23 we replace the $P(C_i|x)$ with $f_i(x,\theta)$ and the expectation is replaced by the appropriate sample average over all training samples. Therefore maximizing the sample MMI is equivalent to obtaining an estimate of the parameters $\hat{\theta}$ which yields

$$\max_{\theta} \frac{1}{N} \left[\sum_{x \epsilon C_1} \log f_1(x,\theta) + \sum_{x \epsilon C_2} \log\left[1 - f_1(x,\theta)\right] \right] \tag{24}$$

where N is the total number of training samples. The above expression assumes that the fraction of the training samples from each class is in proportion to the prior class probability. Note that maximization of the preceding expression can use the backpropagation algorithm since the expression is a simple function of the network output. We also note that the criterion will be sensitive to probability estimates near 0 because of the sensitivity of the logarithm for small values. This is in distinction to training with the mean square error criterion which didn't have this sensitivity.

IV.2 Minimizing the Kullback-Liebler Measure

The Kullback-Liebler information measure [4] can be considered as "distance" between two probability density functions. In terms of the neural network whose outputs are $f_i(x,\theta)$, the estimates of the posterior class probabilities, the K-L measure is given by

$$\mathrm{I}_{\mathrm{K\text{-}L}}(P,f) = \int p(x) \left[\sum_{i=1}^{2} P(C_i|x) \log \frac{P(C_i|x)}{f_i(x,\theta)} \right] dx. \tag{25}$$

Since minimizing Equation 25 is equivalent to maximizing $\sum_{i=1}^{2} p(x|C_i)P(C_i)\log f_i(x,\theta)$ we can readily show, arguing as we did for MMI, that this will result in performing the maximization shown in Equation 24. Furthermore, if we write $f_i(x,\theta) = P(C_i|x) + \epsilon_i(x)$ where $\epsilon_i(x)$ is the error term, we can rewrite Equation 25 as

$$\mathrm{I}_{\mathrm{K\text{-}L}}(P,f) = \int p(x) \left[\sum_{i=1}^{2} P(C_i|x) \log \frac{1}{1 + \frac{\epsilon_i(x)}{P(C_i|x)}} \right] dx. \tag{26}$$

This shows that the K-L measure is a function of the *relative* error as opposed to the mean square error criterion which was a function of the *absolute* error.

V Maximum Likelihood Estimation

Maximum Likelihood (ML) estimation has many desirable properties which includes an understanding of its asymptotic behavior. The ML estimate of network parameters is that selection of the parameters of the pdf of the observations that maximizes the likelihood of these observations. We consider an observation to be a feature vector x_j and C^j, the class membership for the j^{th} observation for $j = 1, \ldots, N$. The likelihood of these observations can be expressed as

$$\lambda = \prod_{j=1}^{N} p(x^j, C^j; \theta) \tag{27}$$

where each term in the product can be written as $P(C^j|x^j;\theta)p(x^j)$ and where $P(C^j|x^j;\theta)$ is either $f_1(x^j,\theta)$ or $1 - f_1(x^j,\theta)$ depending on whether $C^j = C_1$ or C_2. Collecting terms based on the class of x^j gives

$$\lambda = \prod_{x \epsilon C_1} f_1(x,\theta) \prod_{x \epsilon C_2} \left[1 - f_1(x,\theta)\right] \prod_{j=1}^{N} p(x^j) \tag{28}$$

Recall from Equation 18 that $p(x^j)$ is not constrained by the choice of θ and maximizing Equation 28 with respect to θ just depends on maximizing the first two products in the equation. The log of these products is proportional to the MMI criterion give expression 24 and its maximization will give the same value for θ. Since we are not actually interested in $p(x^j)$ there is no need to parameterize this pdf and find its ML estimate.

Having the ML estimate of the neural network parameters enables us to utilize the asymtotic chi-square properties of the distribution of the log likelihood ratio, the Akaike Information Criterion [5] or the Minimum Description Length criterion [6] for comparing neural network models that have different numbers of parameters.

VI Discussion

We have explored some of the consequences of mean square error training, discussed some of the problems and have suggested alternative training criteria to overcome some of the problems and provide additional benefits. The equivalence of ML estimation and the information-theoretic approaches gives a single training criteria many benefits. A useful next step will be the practical application of the proposed training criterion.

Acknowledgment

The author thanks Rich Schwartz and John Makhoul of BBN for their helpful discussions of many of the issues presented in the paper.

REFERENCES

[1] H. Bourlard and C. Wellekens. Links between Markov models and multilayer perceptrons. In *Conference on Neural Information Processing Systems*, pages 502–510, Sept. 1988.

[2] L.R. Bahl, P.F. Brown, P.V. de Souza, and R.L. Mercer. Maximum mutual information estimation of hidden Markov model parameters. In *IEEE Int. Conf. Acoust., Speech, Signal Processing*. Toyko, Japan, April 1986.

[3] J. Bridle. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In *Conference on Neural Information Processing Systems*, 1989. To be published.

[4] S. Kullback. *Information Theory and Statistics*. J. Wiley, New York, NY, 1959.

[5] H. Akaike. Information theory and an extension of the maximun likelihood principle. In B.N. Petrov and F. Csaki, editors, *Proc. 2nd Int. Symp. Inform. Theory*, pages 267–281, Akademia Kiado, Budapest, 1973.

[6] J. Rissanen. Modeling by hortest data description. *Automatica*, 14:465–471, 1978.