Confidence Measures for Neural Network Classifiers

Hugo Zaragoza, Florence d'Alché-Buc.
LIP6, *Université Pierre et Marie Curie*,
4, place Jussieu F-75252 PARIS cedex 05 (F).
Hugo.Zaragoza@lip6.fr

Abstract

Neural Networks are commonly used in classification and decision tasks. In this paper, we focus on the problem of the local confidence of their results. We review some notions from statistical decision theory that offer an insight on the determination and use of confidence measures for classification with Neural Networks. We then present an overview of the existing confidence measures and finally propose a simple measure which combines the benefits of the probabilistic interpretation of network outputs and the estimation of the quality of the model by bootstrap error estimation. We discuss empirical results on a real-world application and an artificial problem and show that the simplest measure behaves often better than more sophisticated ones, but may be dangerous under certain situations.

1 Introduction

Neural Networks (NNs) are commonly used in decision and regression tasks as non parametric estimators. Although a large number of studies have partly clarified their behavior as statistical learning tools, some aspects of their validation are still under question. For instance, in various contexts such as diagnosis, decision support or prediction, we need to know how reliable their estimations are. Diagnosis, for instance, is a classification task where a local criterion is needed to reject decisions that could be dangerous (i.e. costly). Combination methods that take into account the outputs of a pool of decision systems is another example where a local measure of confidence is needed. In this paper we focus on a study of different kinds of local confidence measures for neural networks for classification.

Review of the literature shows that most of the approaches devoted to local confidence measures of neural networks concern mainly the regression framework [1,9,13]. However, we argue here that the nature of confidence measures needed for classification is different in nature. Indeed, classification confidence measures should not be concerned with the uncertainty of the data, which is implicit to the probabilistic nature of the problem and is taken into account by statistical decision theory, but should focus on the accuracy of the model. We will see that, for this reason, confidence measures for classification can only be heuristic in nature. This does not mean that we should not use them but, rather, that they should be used with caution and with knowledge of the measure's underlying assumptions.

In Section 2, we formalize the problem of neural network classification within the framework of statistical decision theory. Section 2.1 briefly discusses the similar problem of confidence measures for regression and Section 2.2 presents some known convergence results of neural network classifiers that justify the application of statistical decision theory. In Section 3 we discuss classification confidence measures of two different kinds, those dealing with the probability of misclassification given the outputs (Section 3.1) and those dealing with the accuracy of the model's outputs themselves (Section 3.2). We will try to explicit, for each measure, the underlying assumptions and potential pitfalls. We introduce a novel confidence measure (Section 3.3) which takes into account the probabilistic nature of the decision problem and the confidence on our probability estimates. In Section 4, we test the different measures on a real-world application and an artificial problem.

2 Confidence Measures for Neural Network Classifiers

Classification can be best described within the statistical decision framework. Patterns are realizations of a random vector X. Classes are realizations of a random variable Y with K discrete unordered possible values. There exists a (fixed but unknown) joint probability law F(X,Y) linking the pattern space to the classes. A classifier is a function *h* from the pattern space to the set of classes {1,...,K}. We will see that the optimal local confidence measure is the probability that the class assigned to object *x* by our classifier, *h(x)*, be different from the true class, *c*, of the object, $P(h(x) \neq c|x)$. Since this probability cannot be exactly computed, we are interested in measures that are correlated to or give us some indication on this probability.

We see in Section 2.1 that, under such a framework, classification confidence is similar to its regression counterpart. However we consider here neural network classifiers and, more generally, classifiers which implement the function *h* by the composition of: i) an estimation of posterior class probabilities (the *estimation step*) and ii) the application of the Bayes decision rule (the *decision step*). Under this framework, the decision itself is done by an «optimal» deterministic function, but the quality of the result depends on the estimation step (see Section 2.2).

2.1 Local error probability in classification and regression.

The same framework described for classification is valid in regression, up to the decision step. Y is now a random variable with continuous values, the *desired* or *output* values. Output values are assumed to be the result of the addition of an unknown function f(x) and some centered Gaussian noise. This hypothesis implies that the distributions p(Y|X) is itself gaussian. Applying the maximum likelihood principle (i.e. minimizing a sum-of-squares error function), one can determine the parameters of the researched function *h(x)*, which is an approximation of the unknown function *f(x)*. Furthermore, *f(x)* is equal to the conditional expectation E(Y|X), so *h(x)* is an approximation to E(Y|X).

Confidence measures in regression arise from the need to know more than just the expectation of the random variable, E(Y|X). Higher order moments, such as the variance, may be used as confidence measures. Considering that the function *h* provides an approximation of E(Y|X), the local probability of error P(|h(x) - y(x) | < e), can be modelled by the variance of Y given X, assuming a gaussian distribution of errors [1], or modelled by a second regressor [9].

Classification differs from regression in the nature of the random variable Y. The underlying posterior class probability distributions P(Y|X) are continuous valued functions as in the regression problem, and are also equal to E(Y|X), if we code appropriately classes as binary vectors. class vectors. However, probabilities themselves are not uncertain or noisy, and furthermore they are subject to the constraints proper to probability functions (they are positive and sum to one). In this case variance is not an issue; since there is no «noise» in the labels, variance of Y is implicit in its probability distribution, and taken into account in the decision step.

2.2 The Bayes Classifier and Neural Networks

Choosing the most probable class for every object corresponds to the Bayes decision rule:

$$\begin{cases} \arg\max_k P(k|x), & \text{if } \max_k P(k|x) > d \\ D, & \text{otherwise} \end{cases}$$

where is the decision (the chosen class), k is the index of classes, x the input pattern to be classified, D is the doubt or rejection class and d is the rejection threshold. The Bayes classifier is optimal in the sense that it leads to the smallest probability of misclassification. This classifier is correct with probability $\max_k P(k|x)$ and incorrect with probability $\min\{1 - \max_k P(k|x), 1 - d\}$. Note that when $\max_k P(k|x) \leq d$ the classifier rejects the input pattern, this classification not being considered either correct or incorrect.

We may interpret $\max_k P(k|x)$ as a confidence value and d as a confidence threshold. Since this leads to the Bayes classifier, which is optimal with respect to misclassification, we obtain an optimal classification rule, an optimal confidence measure and an optimal rejection criterion. These are of course only useful if

we know the posterior class probability distributions.

Neural Networks approximate class posterior probabilities under some general conditions [10]. That is, given a sample $\{(x_n,y0)\}$ where (x,y) are drawn from the fixed but unknown probability law, x is a real vector and y is a vector coding the class, the network's outputs (noted $\hat{P}(k|x)$) after training approximate the posterior class probabilities $P(k|x)$. Furthermore, it has been shown [11] that training minimizes:

$$\varepsilon^2(w) = \int_X [F(x, w) - g_0(x)]^2 \, p(x) \, dx \quad ,$$

where $F(x,w)$ is the NN output for input feature vector x and weight vector w, and $g_0(x)$ is the (optimal) Bayes discriminant function. This means that a NN trained by back-propagation is the best possible NN classifier in the sense that it will provide the best minimum squared error approximation to the Bayes optimal discriminant function. This is why it makes sense to choose the most activated output as the (most probable) class of an input pattern (note however that the local accuracy of this approximation is weighted by the data density p(x)).

Now, let and be the classes with highest true probability $P(k|x)$ and highest estimated probability $\hat{P}(k|x)$ respectively. In other words, is the class assigned by the Bayes classifier to input pattern x, and is the class assigned by our neural network classifier (NNC). If these two classes are the same the probability of misclassification of the NNC will be the Bayes error , regardless of the quality of the estimation. This value is unknown, but we may use as an approximation. On the other hand, if is different from , this approximation will be biased. This is why output activations (unlike true posterior probabilities) are not sufficient to compute or even to estimate the probability of misclassification.

In the 2-class problem, the probability of misclassification when and are different is (see [4] for a discussion on 2-class problems):

$$- P(C_{\vartheta \neq \theta}|x) = 1 - (1 - P(C_\theta|x)) = P(C_\theta|x),$$

This analysis is however difficult to extend to more than 2 classes. The accuracy of the approximated class posterior probabilities will depend on many factors, such as the size and fitness of the data or the model's complexity[1], and if there are more than two classes it seems difficult to estimate the error probability of misclassifed examples.

3 Measures of Confidence

3.1 Heuristic Measures of Confidence

A common measure of confidence used in NN classification is the strength of the most activated output unit. In doing so we assume that our model is reasonably good in approximating the real class posterior probabilities:

$$D_0(x) = \max_k \hat{P}(k|x)$$

Note that the network output's, $\hat{P}(k|x)$, are not however true probabilities, since there is no guarantee that their sum be one. A probability distribution may be trivially obtained by normalizing the outputs. A more statistically meaningful measure would therefore be the normalized version of D_0:

$$D_1(x) = \max_k P(\hat{k}|x) \quad , \text{where}$$

$$P(\hat{k}|x) = \hat{P}(k|x)\left(\sum_{l=1}^{K} \hat{P}(l|x) \right)^{-1}$$

These two measures suffer from the same problem discussed in the previous section: the true object class $\arg\max_k p(k|x)$, and the predicted class $\arg\max_k \hat{p}(k|x)$ may not be the same. In this case the two previous measures are not reliable. In fact, these measures are an approximation of the true probability of good classification when the class chosen is most probably correct, but not otherwise. They do not tell us the confidence on the choice of class. When output activations are close, small errors may easily change the class mostly activated without changing the value of the previously defined confidence measures. A confidence measure that takes this fact into account is the (negative) entropy of the

1.[Tumer & Ghosh 96] have shown that it is possible to estimate the probability of Bayesian error together with the approximation error using a combination of NNs or other approximators of Bayesian classifiers. However, their method cannot be applied to approximate the local probability of error.

normalized network's output [14]:

$$D_2(x) = -H(P(\hat{k}|x)) = \sum_{l=1}^{K} P(\hat{l}|x) \log P(\hat{l}|x)$$

This measure is zero when only one class is activated, and minimal when all classes are equally probable. Implicit to this measure is the belief that as approximated posterior class probabilities get closer, misclassifications are more probable, as explained before; note however that if a classifier strongly misclassifed a pattern, this confidence measure will be misleadingly high. This is not very probable, but may occur, for example, when classifying outliers, or for heavily under-parametrized networks.

Normalizing outputs may be dangerous if network outputs are not close to a true probability distribution. If, for example, the sum of the outputs of a network is close to 0, the outputs are far from a true probabilistic distribution, and it seems wrong to classify such patterns even if entropy is low (which could be the case after normalization). Since outputs should sum up to one for posterior class probabilities, a measure of distance between one and this sum may be used to test the validity of the network output as a probability distribution.

A different type of measure was proposed in [10]; prior class probabilities P(Y=k) may be obtained marginalizing the posterior class probabilities P(Y=k|x), and may at the same time be estimated by counting on a labeled set. The relative entropic distance of these two variables may be then used as a confidence measure:

$$D_3(C_i) = \sum_{l=1}^{K} Freq(C_l) \log \frac{Freq(C_l)}{Freq(y_l)},$$

where y_i is the class assigned by the classifier. This measure however is local to the class but independent of the input vector x (for this reason, it will not be considered further).

3.2 Error Estimates

Error estimates differ from the previous confidence measures in that they measure the accuracy of the output values, independently of the classification problem behind. They are a measure of the validity of the models, not of the difficulty of classification.

There exist several robust methods of error estimation with Neural Networks (see for example [3] Chapter 21). A comparative study of some of them is presented in [13]. We use here the Bootstrap pairs sampling algorithm [3] to determine if it improves on the previous heuristic methods.

The bootstrap estimate of the standard error of the prediction $\hat{P}(k|x)$ is:

$$\left\{ \frac{1}{B-1} \sum_{1}^{B} \left(\hat{P}_b(k|x) - \overline{\hat{P}(k|x)} \right)^2 \right\}^{1/2}$$

where $\hat{P}_b(k|x)$ is the estimate of the b-th bootstrap ensemble, B is the total number of bootstrap ensembles and $\hat{P}(k|x)$ is the mean of their estimate: $\sum_{b=1}^{B} \hat{P}_b(k|x)/B$. Since we are not interested in the true error estimate but only on the relative values, we define our measure more simply as the variance of the ensemble:

$$D_4(x) = \sum_{1}^{B} \left(\hat{P}_b(k|x) - \overline{\hat{P}(k|x)} \right)^2$$

This measure has been used as a measure of confidence in classification problems with bootstrapped ensembles [12]. The variance of an ensemble of networks built by crossvalidation has also been utilized as the figure of merit to choose input patterns for active learning in [6]. In [7], on the other hand, the minimal and maximal output values of the ensemble to display confidence intervals for human interaction.

3.3 Combining Classification Heuristics and Error Estimates

Error estimates are independent of the probabilistic interpretation we make of the network‘s output. Unlike their probabilistic counterparts, they do not take into account the classification decision that needs to be carried on the outputs. Using D_4, for example, low variance indicates high confidence on the accuracy of the output. On the other hand, using D_1, an output value near 1 indicates high confidence on the class assignment, taking the output activation at face value. These two types of measures are therefore complementary. That is the reason why we propose to combine them into a unique measure of the

type:

$$D_5 = f(D_0, D_4)$$

where f is some simple function which combines the risk of misclassificarion and the probability error estimate, may be used effectively. In particular, we try the simplest combination of these two quantities, their difference: $D_5 = D_0 - D_4$. This is an heuristic measure with no probabilistic interpretation. Intuitively, the activation (that is, the confidence, if posterior probabilities were correctly approximated) is moderated by the variance (interpreted as the quality of the approximation). Note that the variance is necessarily in the $[0, l^2/4]$ range, where l is the difference between minimal and maximal output activations (usually 1).

3.4 Observations on 2-class Problems

There are many applications where classifiers are trained to detect a particular class of objects. In many problems, for example, classes are not necessarily exclusive (that is, an object may belong to several classes simultaneously). It is typical in these cases to model each class independently, with a single NN of one output. The output of the network models the posterior probability of the class $\hat{P}(k|x)$, and $1 - \hat{P}(k|x)$ is then the posterior probability of the object not belonging to the class.We note that, for such cases, the confidence measures available are greatly reduced. The sum of the output probabilities is 1 by definition, rendering D_3 useless. Entropic measures, as well as normalization, become unnecessary since the probability of one class becomes completely determined by the probability of the other. Consequently, measures D_2 and D_1, in the two class problem, become equivalent to D_0 (up to a scaling function). Similarly, D_2 becomes equivalent to computing the difference between the real and the estimated class frequencies.

4 Results

We have seen that confidence measures introduced in this paper make some strong assumptions and are heuristic in nature; no single measure portrays completely the confidence (that is, the true probability of error, taking into account all factors). In this section we apply the measures introduced in Section 3 to a real-world application and an artificial classification problem, and discuss their behavior with respect to several figures of merit.

The real-world application discussed here is an automatic truck-motor failure diagnosis system. Trucks are represented by pattern-vectors of 35 real valued variables which define the state of a truck motor. Different types of failures must be identified; some failure subsets are exclusive but most are not. The most common class by large is the «No Failure» class. We will study two different problems within this application: a 2-class diagnosis problem and a n-class classification problem. In the first case, only one failure is considered; a truck must be classified as either «Failure A» (FA) or «No Failure A» (NF). The number of FA examples is one third of the number of NF examples. In the second case the are four exclusive failures (B, C and D) roughly equally represented

The reason we test these two different settings is the following. We noted that only D_0 and ensemble measures are applicable to 2-class problems (the rest being either equivalent or meaningless). Diagnosis is a typical 2-class application, were one of the classes is under-represented (in our example, the class A with respect to NF). Unfortunately, output activation measures of the D_0 type introduce a strong bias in the rejection procedure, since under-represented classes tend to have lower output activations. This leads to the systematic rejection of the under-represented class, which is optimal with respect to overall performance, but highly undesirable in cases were missing true failures is more dangerous than detecting false ones (i.e. when the risk of false alarms is low). We use a training and test set of 1500 and 500 patterns respectively for both experiments.

For the artificial classification problem we used the Breiman Waves [2]. It is a symmetrical three-class classification problem in 21 dimensions. Class prior probabilities are equal. We used a data-set of 1000 points for training and 3000 for testing. There are no under-represented classes in this problem, so there is no «privileged» class a priori; we monitor nevertheless both overall performance and the performance for one of the classes independently, as in the real-world problem.

In all experiments we use Multi Layer Perceptrons with sigmoidal hidden and output units [5] (the most

common neural network classifier architecture). All networks have 5 hidden units and are trained with the batch conjugated gradient algorithm (training is stopped when the learning quadratic error gradient decreases under 0.01).

In the following we will discuss results on the real world-problem and compare these with the artificial problem. Results will be presented as performance versus rejection ratio curves. In these curves, as we go from the left to the right we move from low rejection (were only highly unconfident patterns are rejected) to high rejection (were only highly confident patterns are classified). If confidence measures are adequately correlated with the probability of misclassification of patterns, the slope of the curves should be positive; a negative slope means that more hits than errors are being rejected. A horizontal line, on the other hand, indicates a non correlated confidence measure, equivalent to a random rejection scheme. The dotted line represents the performance of an «oracle» confidence rule which would assign a confidence of 1 to well-classified examples and 0 to wrongly-classified ones.

In the first experiment (Figure 1) we apply ensemble confidence measures to the 2-class real-world problem. The overall test classification rate is 72%, but note that performance on the FA class is only of 49%. With respect to overall performance (top graph) the two best confidence measures are clearly D_0 and D_5. With respect to class FA however (bottom graph) we see that D_0 is a dangerous confidence measure: its slope is negative. D_4, on the other hand, is a good rejection criterion for FA examples but, overall, is only useful in the low rejection range (or, alternatively, for high variances). The proposed measure D_5 effectively combines the strengths of both previous measures, yielding an overall performance gain, without penalizing the under-represented class. The same experiences were carried out with the artificial problem (Figure 2). For this problem all classes are equally represented so there is no «privileged» class, as it was the case in the previous experience. We see in fact that overall and single-class behaviors are quite similar. The ensemble variance (D_4) is clearly inadequate in this case, while D_0 and D_5 perform similarly. This is important since it shows that the proposed measure D_5 performs as well as output activation even in situations when variance is a poor indicator of confidence.

In Figure 3 we present the three heuristic confidence measures applicable to single classifiers in a multi-class classification problem. We can observe that D_0, which is simply the activation strength of the output (bounded by the sigmoidal activation function, but otherwise not normalized) proofs to be as good a confidence measure as the normalized output entropy D_2. Both of these are clearly superior to the normalized output D_1. With respect to the artificial problem, the only reasonable confidence measure is the output activation, the other two measures leading to surprisingly poor results.

Finally, in Figure 4, we compare the D_0 measure computed with a single network (SINGLE), the same measure computed by the average of the bootstrapped ensembles (BTSP), and the proposed D_5 measure, on the 2-class real-world problem and the artificial problem. For the 2-class problem (Figure 4 top) we have plotted both overall performance (top curves) and performance relative the under-represented class (bottom curves). For the artificial problem (Figure 4 bottom) we plotted overall performance (top curves) and percentage of points from one of the classes (bottom curves). We see that the quality of the three measures for overall classification is very similar. In the artificial problem SINGLE is not as good as BTSP or D_5 at high rejection rates; this may be due to the uneven rejection rate for one of the classes (as seen in the bottom D_0 curve). In the real-world problem, bootstrapped performance is slightly superior to single network performance (.03% overall and.09% with respect to the under-represented class) but confidence measures overall are similar (the slope of the curves being the same). In the 2-class problem, we see however that the quality of the three different measures is quite different with respect to the under-represented class. As seen already in Figure 1, the bootstrap mean output measure (BTSP), rejects on average more good classifications than misclassifications (relative to the under-represented class). The single network D_0 measure is a better confidence measure with respect to the under-represented class, although it leads to lower performances. Finally, the proposed measure D_5 produces the highest overall performance as well as the highest performance with respect to the under-represented class; it is therefore the only confidence measure in our experiences that leads to a consistent rejection criterion for both 2-class and multi-class neural network classification.

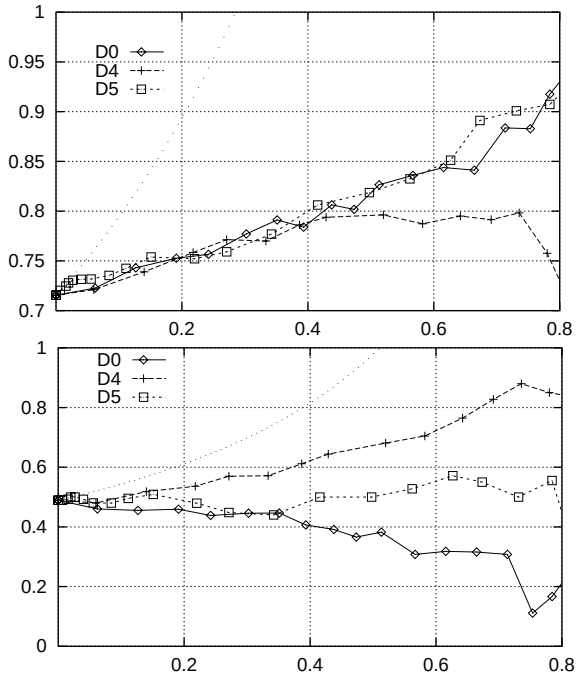

Figure 1: Ensemble Confidence Measures (Real World 2-class problem). Overall performance (top), should be contrasted with performance with respect to the under-represented class (bottom).

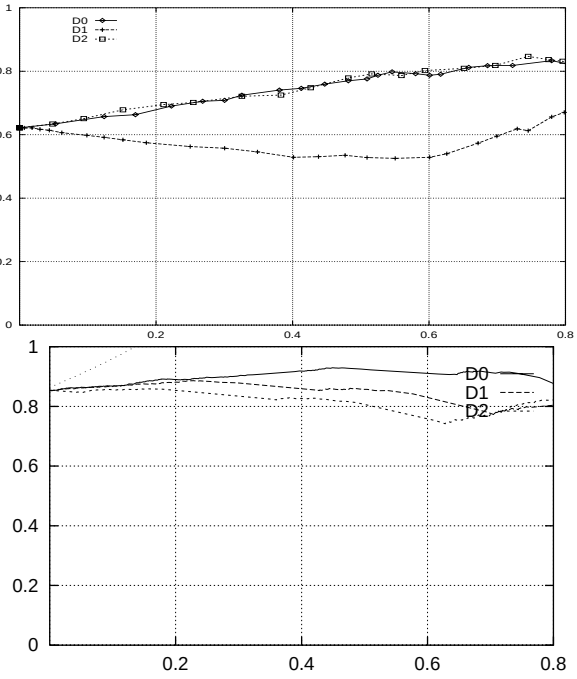

Figure 3: Heuristic Confidence Measures: output activation (D0), normalized (D1) and normalized output entropy (D2). Results are presented on the real-world 3-class problem (top) and the artificial problem (bottom).

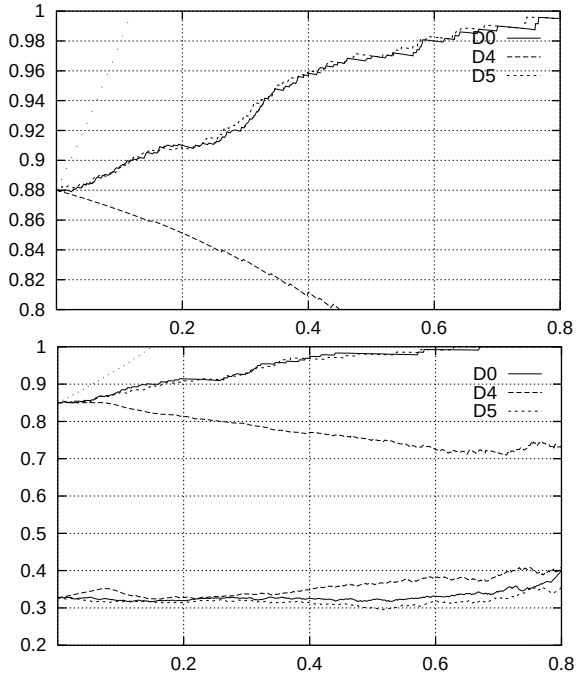

Figure 2: Ensemble Confidence Measures (Artificial problem). Overall performance (top graph) should be contrasted with performance with respect to a single (the most difficult) class (bottom graph).

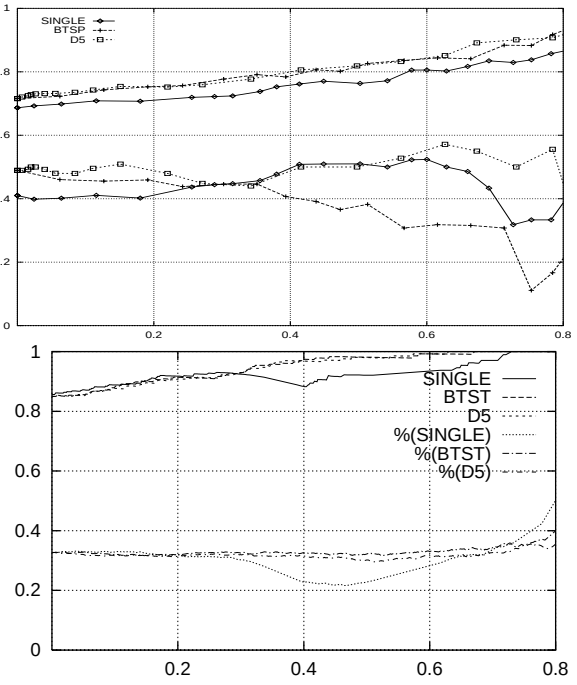

Figure 4: Comparison of Heuristic and Error Confidence Measures. SINGLE and BTSP are D0 measures computed by a single network and an ensemble respectively. For each of the two problems (top graph: real world 2-class problem, bottom graph: artificial problem) curves indicate overall performance (top curves) and class A performance (bottom curves).

5 Conclusions

We reviewed some notions from statistical decision theory that offer an insight on the determination and use of confidence measures for classification with Neural Networks. We have argued why an «optimal» confidence measure is not possible in most practical situations, and we have presented several heuristic measures for the determination of confidence values. We then proposed a simple measure which combines the benefits of the probabilistic interpretation of network outputs and the estimation of the quality of the model by bootstrap. A real-world application and an artificial problem were used to compare the different confidence measures for 2-class and n-class classification. Results show that the simplest confidence measure, output activation, is as good (and often better) as the rest of confidence measures proposed in the literature. One exception are 2-class problems with one under-represented class, in which output activation is in fact quite dangerous. The measure we have proposed was the only one which showed good behavior both for n-class and 2-class problems.

References

[1] Bishop C.M., *Neural Networks for Patterns Recognition*. 1995, Clarendon Press, Oxford.

[2] Breiman L., Friedman J.H., Olshen R.A., Stone C.J., *Classification and Regression Trees.* (1984) Wadsworth & Brooks, Cole Statistics/Probability Series, Pacific Grove, Cal.

[3] Efron B., Tibshirani R., *An Introduction to the Bootstrap,* 1993, Chapman and Hall, London.

[4] Friedman J.H. On Bias, Variance, 0/1 - loss, and the Curse-of-Dimensionality. *Data mining and Knowledge Discovery,* 1996, 1(1) 54-77.

[5] Hertz J., Krogh A. et Palmer R.G. (1991) *Introduction to the Theory of Neural Computation.* Addison-Wesley Publishing Company, Redwood City, CA.

[6] Krogh and Vedelsby. Neural Network Ensembles, Cross Validation, and Active Learning, in *Neural Information Processing Systems (NIPS 7),* 1995, p.234-238.

[7] Lippmann, Kukolich and Shahian. Predicting the Risk of Complications in Coronary Artery Bypass Operations using Neural Networks, in *Neural Information Processing Systems (NIPS7), 1996.*

[8] MacKay. The Evidence Framework Applied to Classification, in *Neural Computation,* 1992., vol.4, no.3, p.720-736.

[9] Nix and Weigend. Learning Local Error Bars for Nonlinear Regression, in *Neural Information Processing Systems (NIPS7), 1996.*

[10] Richard M.D. and Lippmann R.P. Neural Network Classifiers Estimate Bayesian a posteriori Probabilities, in *Neural Computation,* 1991, vol.3, p.461-483.

[11] Ruck et al. (1990) The Multilayer Perceptron as an Approximation to a Bayes Optimal Discriminant Function, in *IEEE Transactions on Neural Networks, vol.1 no.4 p.296-299.*

[12] Shimshoni Y. and Intrator N., Classifying Seismic Signals by Integrating Ensemble of Neural Networks, in *ICONIP'96.*

[13] Tibshirani R., A Comparison of Some Estimates for Neural Network Models, in *Neural Computation 8,* 1996, p.152-163.

[14] Wan (1993) Neural Network Classification: A Bayesian Interpretation, in *IEEE Transactions on Neural Networks,* vol.1 no.4 p.303-306.