

Neural Networks with On-The-Fly Confidence-Estimation

Ingo Dahm

(Computer Engineering Institute, University of Dortmund, Otto-Hahn-Strasse 4, D-44221 Dortmund)

e-mail: ingo.dahm@udo.edu

Abstract: Artificial Neural Networks are an established solution for classification. Many Neural Networks consist of perceptrons (e.g. Multi-Layer-Perceptron Networks). In this paper, we propose a concept to extract reliability information of a perceptron-based Neural Networks decision. Therefore, we extend the conventional perceptron and suggest a simple training strategy.

Keyword: classification, neural network, perceptron, reliability, confidence, real-time constraint, signal-space

1 Introduction

Many classification problems can be solved by the use of Artificial Neural Networks. On the other hand, conventional (non-adaptive) classifiers provide additional features, particularly the extraction of so-called Soft-Information, a measure of the reliability of a decision. Postprocessing algorithms can profit from that extra information, as already discussed in the past [1, 2].

Thus, we suggest a concept to enhance a conventional Multi-Layer-Perceptron-based Neural Network by components that allow an on-the-fly confidence-estimation.

Thereto, the first section provides definitions and background information. Afterwards, we present the concept of reliability-estimation. Finally, we present experimental results and end with a conclusion.

2 Background and Theory

For the following, an object is given by M significant features which can be measured (e.g. by sensors). Thus, an observed object is represented by a so-called observed signalpoint \vec{r} in the M -dimensional feature-space \underline{S} . If all samples of the same class would be identical, all observed objects would be represented by fixed points \vec{c}_i (so-called admissible signal points) in \underline{S} .

Each \vec{c}_i represents a class i . Therefore, \underline{S} can be divided into subspaces, such that each section is uniquely associated to a certain class i . This partitioning can be done efficiently by hyperplanes that divide \underline{S} into Voronoi Regions.

A simple implementation of an M -dimensional hyperplane is the so-called Adaline-Element [3]. For the following, we approximate the Adaline by an Artificial Perceptron [4]. The so-called weight vector \vec{w} of the perceptron is scalar-multiplied by the input vector $(r_1, r_2, \dots, r_M)^T$. The result is weighted by a user-defined activation function A . Thus, the output of the perceptron is given by $A(\vec{w} \cdot \vec{r})$.

Steingrimsson et al. have presented a method to extract reliability information from a so-called Signal-Space Detector [5, 6]: An observed signal point, which is positioned close to an admissible signal point, is assumed to represent a confident decision. On the other hand, if \vec{r} is located close to a decision plane, then it marks an unreliable decision. That is, the distance from an observed signal point to the admissible signal points is a measure for the reliability of the detector output [2].

The consideration of this distance allows the calculation of individual probabilities for each possible perceptron output: Generally, \vec{r} can either be assigned to a certain class C ($\vec{r} \in C$) or not ($\vec{r} \notin C$). Assuming that the observed signal point is \vec{r} , the reliability of the classification i , which is called $L(i)$, can be calculated with the so-called log-likelihood ratio [1]:

$$L(i) = \ln \frac{P(i = C | \vec{r})}{P(i \neq C | \vec{r})} \quad (1)$$

Note, that a positive value $L(i)$ indicates that \vec{r} is classified to belong to C while a negative value means that \vec{r} is classified *not* to belong to C . The absolute value of $L(i)$ is the reliability of the decision.

Based on that approach, we developed a low-complexity architecture with soft-outputs [2]. Now we show, how this approach can be adapted to a generic Multi-Layer-Perceptron Network.

3 Classification with Confidence-Estimation

For the following, we define a generic Single-Layer-Perceptron Network (G-SLP) as a set of ψ perceptrons

in one layer. Each perceptron provides M input-weights. Thus, the M -dimensional signal-space is partitioned by ψ decision-planes. Moreover, this G-SLP transforms the M -dimensional feature-space into a ψ -dimensional auxiliary signal-space.

Further, a generic Multi-Layer-Perceptron (MLP) Network consists of λ G-SLP's which are connected among each other.

There are two ways to apply Eq. 1 to the MLP in order to estimate the confidence of its classifications. If it is used in the λ 'th layer, then it is applied to the ψ_λ -dimensional space only. This keeps the implementation costs low. On the other hand, unreliable decisions in the first $(\lambda-1)$ layers does not affect the computation of the overall confidence.

Thus, we suggest to apply Eq. 1 to every perceptron of the MLP. Obviously, this way is of higher complexity but a higher accuracy of the computed confidence can be expected.

The hyperplanes that partition the signal-space are given by their normal vectors \vec{n}^k – which are multiplied by the input vector \vec{r} – and a bias d^k , which represents the distance to $\vec{0}$ if $|\vec{n}^k| = 1$:

$$H : (n_1^k, n_2^k, \dots, n_M^k)^T \cdot \vec{r} + d^k = 0 \quad (2)$$

Such a hyperplane can be modeled by a conventional perceptron p^k : The weight vector $\vec{w}^k = (w_1^k, w_2^k, \dots, w_M^k)^T$ is dot-multiplied by the input vector and a bias w_{M+1}^k is added. If \vec{r} is used as input vector, then \vec{w}^k can be interpreted as the normal vector of a bounding hyperplane H . $H(\vec{r})$ is weighted by an activation function A . Thus, the perceptron's output $P := p^k(\vec{r})$ is given by

$$P : \vec{r} \rightarrow A \left[(w_1^k, w_2^k, \dots, w_M^k)^T \cdot \vec{r} + w_{M+1}^k \right] \quad (3)$$

The set of k_{max} hyperplanes is modelled by the same amount of perceptrons. The input vector of each perceptron is given by a combination of inputs r_i and other perceptron outputs p^k . To build an efficient constellation of hyperplanes, all weight vectors must be initialized and trained by a user-defined learning technique. We used the back-propagation algorithm [4]. This basic approach is easy to implement and a proper solution to train artificial neural networks. A sigmoid activation function as illustrated in Equation (4) should be used in order to simplify the learning rule [4]:

$$A : x \rightarrow [1 + \exp(c(t-x))]^{-1} \quad (4)$$

3.1 Training Techniques of the MLP

During the training phase, the weights are adjusted. Due to that, the hyperplanes are rotated by modified normal vectors and shifted by the changed bias'. The distances to the

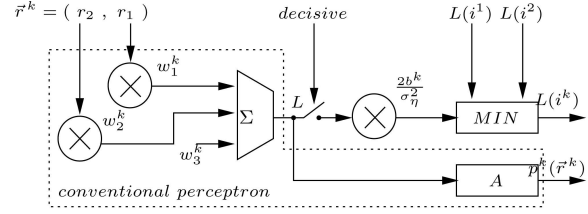


Figure 1: Low complexity implementation of a modified perceptron in order to compute reliability-information.

bounding hyperplanes are needed to extract the confidence information. Thus, we normalize \vec{w}^k to $|\vec{w}^k| = 1$ after each training step. Then, the distance from \vec{r} to H^k can be computed by applying the inverse function $A^{-1}(p^k)$:

$$A^{-1} : p^k \rightarrow t - \frac{1}{c} \cdot \ln((p^k)^{-1} - 1) \quad (5)$$

In practical implementations, it is unreasonable to calculate the distance by applying Eq. (5) since the computation of A^{-1} demands costly operations. Instead, the perceptron structure should be modified as illustrated in Fig. 1: The confidence-information $L(i^k)$ should be calculated simultaneously to the perceptrons output $p^k(\vec{r}^k)$. The use of the auxiliary value

$$L = L^k := \vec{w}^k \cdot \begin{pmatrix} \vec{r}^k \\ 1 \end{pmatrix} \quad (6)$$

which is calculated *before* applying A - helps to avoid the application of A^{-1} . Thus, the entire classification including reliability estimation is done in one processing step of the neural network.

3.2 Extraction of Reliability-Information

The concept of reliability-estimation is based on the work of Steingrimsson [5, 6]. It calculates the soft-outputs by the geometric distance of the observed signal point to the bounding hyperplanes. If a model for the distribution of observed objects in the feature-space is known, the position of all admissible signal points \vec{c}_i can be precalculated. With this information, an observed signal point $\vec{r} \in S_i$ can be modelled by adding noise η (with the variance σ_η^2) to \vec{c}_i :

$$\vec{r} = \vec{c}_i + \eta \quad \forall k \text{ with } (\vec{r}_k, \vec{c}_i) \in S_i \quad (7)$$

Then, the reliability $L(i^k)$ of the decision i^k is given by the normalized dot-product of \vec{r} and the normal vector \vec{w}^k of the hyperplane H^k after Eq. (8) [2]. Note, that b^k indicates the distance from \vec{c}_i to the hyperplane H^k .

$$L(i^k) = \frac{2b^k}{|\vec{w}^k| \cdot \sigma_\eta^2} \cdot \vec{w}^k \cdot \begin{pmatrix} \vec{r}^k \\ 1 \end{pmatrix} = \frac{2b^k}{|\vec{w}^k| \cdot \sigma_\eta^2} \cdot L^k \quad (8)$$

In many cases, the exact position of the admissible signalpoints, and consequently b^k , will be unknown. To handle this problem, we suggest to use the focal point of all observed signal points r_i in the same subspace S_i as an estimate for \vec{c}_i . This methodology has already been successfully applied to specific classification problems [7, 8].

By using an estimated \vec{c}_i , it is easy to calculate σ_η^2 . Thereto, we assuming a Gaussian Noise distribution. The distance b^k to each adjacent hyperplane H^k is given implicitly by the dot-product $\vec{c}_i \cdot \vec{w}^k$ and can be calculated efficiently in the MLP-Network.

Some hyperplanes which does not influence the actual decision [2]. Thus, we must distinguish between bounding hyperplanes which contribute to the detector output (decisive hyperplanes) and those who do not (non-decisive hyperplanes).

To find out, if a perceptron represents a decisive hyperplane, we invert the result of the weight-function p^k of the previous perceptron by computing $(1 - p^k)$. Then, $(1 - p^k)$ is propagated through the rest of the neural network¹ If the final result changes, the hyperplane is claimed to be decisive. Only the decisive hyperplanes contribute to the calculation of the soft-outputs.

Finally, we have to merge all confidence information to an overall value. For our system it seems reasonable to apply the MIN-rule as suggested in [2]: We determine the confidence for each decisive perceptron. The overall confidence is given by the minimum of all calculated reliabilities. Geometrically spoken, we use the shortest distance to a decisive hyperplane as overall-reliability information. An implementation of this approach is illustrated in Figure 1.

4 Experimental Results

To evaluate the performance of the suggested approach, we have to observe three major characteristic properties:

1. correctness of the classification
2. speed of the reliability-extraction
3. quality of the extracted reliability information

Therefore, the performance of the suggested approach is evaluated by three different experiments.

¹To avoid costly computation, we recommend the use of a look-up table.

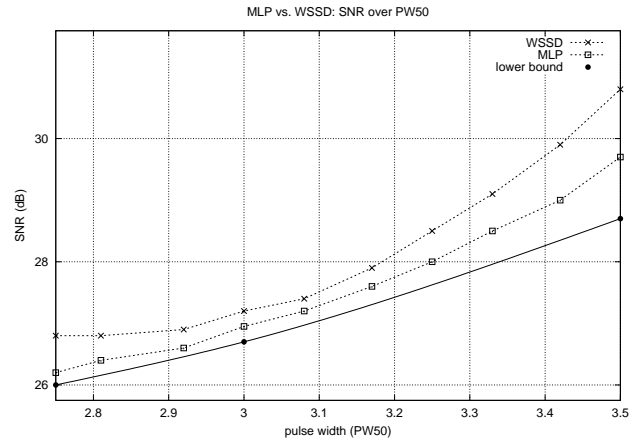


Figure 2: Performance of the suggested concept compared to a stationary WSSD-detector.

4.1 Correctness of Classification

First, we use the suggested classifier as detector of a read-channel in a magnetic recording (hard disk) environment. We compared the accuracy of the suggested approach with the performance of a conventional detector.

Thereto, we measure the performance of a so-called Signal-Space Detector with Noise Whitening (WSSD) on a magnetic channel. We measure the Signal-Noise-Ratio (SNR) which is needed to obtain a Bit-Error-Rate (BER) of 10^{-5} for a varying pulse width $2.75 \leq PW_{50} \leq 3.5$.

We observed, the WSSD reaches the illustrated lower bound (see Fig. 2), if the preprocessing unit - the so-called equalizer - is perfectly adjusted to the channel parameters. Under the influence of parameter noise, the performance of the WSSD worsens significantly (WSSD in illustration). On the other hand, the suggested approach compensates the additional noise (MLP in Fig. 2).

4.2 Speed of Reliability-Extraction

As already stated in the title - the reliability information can be extracted on-the-fly. The calculation is done by the dot-multiplication with the weight vector of every perceptron. Thereto, the weights must be normalized during the training phase, which slightly increases the training time compared to conventional approaches.

Further, the calculation of the minimal reliability of all prior depending decisions is needed. As the reliability of each decision is calculated simultaneously to the decision itself, the overhead of the whole calculation is given by the time for finding the minimum, which typically can be done in real-time.

Third, the estimation whether a perceptron is decisive to the actual classification has to be done at runtime, too. Without look-up tables, this slows down the classification

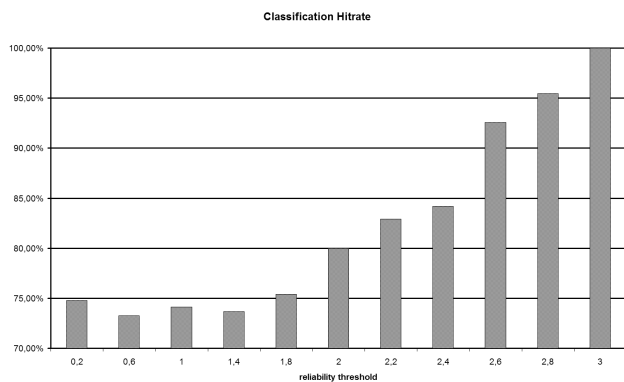


Figure 3: Hitrate of the classification depending on the reliability threshold. The number of correctly classified patterns rises with the computed confidence. This proves the relevance of the soft-information.

by factor $\frac{1}{2} \sum_{k=1}^{\lambda} \psi_k$. Therefore, we recommend the use of a look-up table when large MLP-networks are required.

4.3 Quality of Reliability-Information

The quality of the extracted reliability-information is analyzed by another experiment: We use the MLP to classify a set of high-dimensional objects that represent parameters of a robot walking engine[8]. An MLP is taught to forecast the speed of a given parameter using the backpropagation learning approach. We achieved a hitrate of about 75 percent.

Then, we added the reliability-information extraction module to the same neural network configuration. We forced the network to classify only that patterns, which let the reliability-level exceeded a given threshold. Then, we measured the hitrate of the performed classifications.

As illustrated in Figure 3, the hitrate increased significantly for that classifications with a high reliability-value. Thus, the extracted reliability-information is proofed to be useful for postprocessing modules.

5 Conclusion

In this paper, we suggested an approach to extend the conventional perceptron by an on-the-fly reliability-estimation. We have shown, how this approach can be applied to generic Multi-Layer-Perceptron Networks.

This novel classifier combines the flexibility of conventional MLP-approaches with the possibility to use additional information about the reliability of a classification. Although the performance of optimized static classifiers cannot be obtained under optimum conditions, that classifier performs notably better under the influence of time-

variant impairments or heavy non-stationary noise. The soft-information which is generated on-the-fly can be used to build powerful applications.

Unfortunately, the suggested module requires the use of large look-up tables. To equip small mobile devices with such classifiers, a low complexity implementation is essential. Thus, our future work will concentrate on complexity reduction in order to develop a classifier that can be used in mobile devices, too.

6 Acknowledgments

This work was supported by the Deutsche Forschungsgemeinschaft (DFG) under the program of emphasis SPP 1125. The author thanks Stefan Schmermbeck for his valuable ideas and discussions.

References

- [1] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," vol. IT-42, pp. 429–445, March 1996.
- [2] S. Schmermbeck, G. Stromberg, M. Hassner, and U. Schwiegelshohn, "Low-Complexity Signal Processing for ISI-Channels," in *Proceedings of the Globecom Conference 2003*, December 2003.
- [3] B. Widrow, *Self-Organizing Systems 1962*, ch. Generalization and Information Storage in Networks of Adaline Neurons, p. 435. Spartan Books, 1962.
- [4] S. Haykin, *Neural Networks: A Comprehensive Foundation*. No. ISBN 0-02-352761-7, Macmillan College Publishing Company Inc., 1994.
- [5] B. Steingrímsson, J. Moon, and T. Oenning, "Signal Space Detection for DVD Optical Recording," *IEEE Transactions on Magnetics*, vol. 37, pp. 670–675, March 2001.
- [6] B. Steingrímsson, "Soft Signal Space Detection for the Lorentzian Magnetic Recording Channel," *IEEE Transactions on Magnetics*, vol. 38, September 2002.
- [7] I. Dahm and J. Ziegler, "Adaptive methods to improve self-localization in robot soccer," in *RoboCup Symposium Fukuoka*, 2002.
- [8] I. Dahm and J. Ziegler, "Using artificial neural networks to construct a meta-model for the evolution of gait patterns of four-legged walking robots," in *International Conference on Climbing And Walking Robots (CLAWAR)*, November 2002.