UDINE-92-09-AA
Fermilab Library
0 1160 0019481 5

UNIVERSITA' DI UDINE — ISTITUTO DI FISICA
Via Fagagna 208, I-33100 Udine (Italy)

Udine Report 92/09/AA

10 September 1992

Using Neural Networks for the Classification of the Decays of the Z^0 Boson

G. Cosmo, A. De Angelis, P. Eerola, J. Kalkkinen,
L. Lyons, N. de Groot, M. Los

Abstract

An effective classification of the hadronic decays of the Z^0 boson according to their native flavour can be obtained by using feed-forward neural networks. This possibility has been exploited by using the data collected by the DELPHI detector at LEP during 1991.

Presented at "Computing in High Energy Physics", Annecy (France), September 1992. To be published in the Proceedings.

Using Neural Networks for the Classification of the Decays of the Z^0 Boson

G. Cosmo, A. De Angelis

Dipartimento di Fisica dell'Universita' di Udine and INFN Trieste, Udine, Italy

P. Eerola, J. Kalkkinen

Research Institute for High Energy Physics, Helsinki, Finland

L. Lyons

University of Oxford, Oxford, UK

N. De Groot and M. Los

NIKHEF-H, Amsterdam, The Netherlands

An effective classification of the hadronic decays of the Z^0 boson according to their native flavour can be obtained by using feed-forward neural networks. This possibility has been exploited by using the data collected by the DELPHI detector at LEP during 1991.

The possibility of using a Neural Network (NN) for the classification of the hadronic decays of the Z^0 was explored in Ref. [1], in which the problem of separating Z^0 decays into $b\bar{b}$ pairs was considered. The result of this study was that, in the case of a perfect detector, a separation could be achieved with a higher efficiency than with respect to traditional separation variables. Further studies [2] demonstrated that, also in the presence of detector effects, NNs could be a useful tool for the classification of $b\bar{b}$ events, and preliminary results on data have been presented in 1991 [3].

The DELPHI collaboration has recently submitted a paper on this subject [4]. A reasonable stability in the classification with respect to the models used in the simulations, needed for training and testing the NN, has been achieved. It has been possible to measure the rates of the hadronic decays of the Z^0 into $b\bar{b}$ and $c\bar{c}$:

$$\Gamma_{c\bar{c}}/\Gamma_h = 0.151 \pm 0.008(stat) \pm 0.041(sys) \quad (1)$$

$$\Gamma_{b\bar{b}}/\Gamma_h = 0.232 \pm 0.005(stat) \pm 0.017(sys)\,. \quad (2)$$

In this paper, we will illustrate some technical points related to the network optimization, and show that the use of NN allows substantial improvements in the purity of the tagged samples of events with respect to traditional variables.

The Neural Network used as an input 19 variables, mainly coming from the shape of the event. Variables related to impact parameters and to the presence of leptons were also used, to reinforce the stability of the separation with respect to systematic effects.

One node in the input layer was associated with each of the input variables x_i. The input variables define the pattern space P. There were 25 nodes in the hidden layer. The outputs of the three output nodes, forming the vector $\vec{\Theta}$, belong to the feature space F. The components of the output vector were assigned to the three quark classes $u\bar{u}$+$d\bar{d}$+$s\bar{s}$ (unresolved), $c\bar{c}$ and $b\bar{b}$.

In principle, a single hidden layer is enough to perform any mapping of a continuous function between pattern space P and feature space F. It was checked that by adding a second

hidden layer the performance of the network did not improve, but the time needed to train the network increased substantially.

When the number of hidden nodes was reduced from 25 to 18 and further to 15, it was observed that the three output nodes of the NN gave increasingly similar output values, degrading the separation of the classes. It was also checked that by going much beyond 25 hidden nodes one begins to introduce useless nodes to the hidden layer. Therefore, the network with 25 hidden nodes was chosen. A single network with three output nodes was used instead of three single output networks to account automatically for correlations between classes.

In the structure chosen, each node performs a weighted sum of the output values from all the nodes of the previous layer. The node output is computed via a sigmoid function

$$g_T(x) = \frac{1}{1 + e^{-2x/T}}$$

at a "temperature" T. The output o_i of the i-th node of a layer (starting from the second) is then

$$o_i = g_T(\sum_j \omega_{ij} o_j) \, .$$

where the sum is made over the nodes of the previous layer.

The network training procedure fixes the values of the weights ω_{ij} associated with the node interconnections. The weights can be both positive and negative. If the value of the weight ω_{ij} is zero, then there is no connection between nodes i and j. The aim is to find a mapping of the input pattern space ($x_i \in$ P) to the feature space ($\Theta \in$ F), such that a good separation of events belonging to a class A from events belonging to the complementary class $\overline{A}$ is obtained. Each class is associated with one output node. Two symmetric target values (1 for class A and 0 for class $\overline{A}$) were used for each of the three output nodes.

In the back propagation learning algorithm the output feature values for the training input events are computed and compared with the desired target values. A squared error function E is computed to quantify the difference between the obtained output $\vec{\Theta}$ and the desired target $\vec{t}$,

$$E = \frac{1}{2} \sum_{events} (\vec{t} - \vec{\Theta})^2 \, .$$

This function is minimized by changing ("updating") the weights by an amount computed from the error function by the gradient descent method [5]. The process is controlled by the "learning strength parameter" η and the "momentum" α [5]. Each updating step in the space of weights, computed by gradient descent, is multiplied by η and added to the previous step, multiplied by α. To smooth out fluctuations, weights are updated using the cumulative error from a number of input training events.

For the training of the system, a set of 30,000 simulated events, generated with the JETSET Parton Shower Monte Carlo model (JETSET PS), was used. The number of independent events in each class was thus an order of magnitude larger than the number of weights in the network (550 in our case). The weights were updated every 10 events, chosen at random from the three classes $u\overline{u}$+$d\overline{d}$+$s\overline{s}$ (unresolved), $c\overline{c}$ and $b\overline{b}$, in such a way that, on the average, there was an equal number of events from each of the three classes[1]. In the following, this sequence of 10 events will be referred to as "update".

[1] It has been verified that equal sample sizes improve the performance of the network [2], and reduce biases.

Changing the parameters η and α during the training is convenient in order to allow for a fast movement in the space of weights in the early stage of training, and to obtain a controlled approach to the minimum at the later stage. For this reason, the learning and momentum parameters were decreased and increased respectively after every 3,000 updates (an "epoch") according to a rule:

$$\eta_t = \eta_{t-1} \times (\eta_{min}/\eta_{t-1})^{k_\eta}$$
$$\alpha_t = \alpha_{t-1} \times (\alpha_{max}/\alpha_{t-1})^{k_\alpha},$$

where η_{min} and α_{max} are the minimum (maximum) allowed values for the parameters, and subscript t $(t-1)$ refers to the epoch number. Exponents k_η and k_α were set to 0.05 and 0.14, respectively. Given the finite value of the weight change, the gradient descent method might lead to an occasional increase of the error value. In this case, the parameters were reset to their initial values.

The architecture of the network is summarized in Table 1, together with the parameters used in the training phase.

Nodes in the hidden layer	25
T	2.0
α (training)	0.4 - 0.9
η (training)	0.05 - 0.0001

Table 1: *Characteristics of the NN.*

At each step of the learning procedure, an indication of the network performance can be inferred from the error function. A more reliable evaluation is obtained by testing the response of the network on a set of input events independent of the training set. The behaviour of the error function with this new test set shows the error the network makes in generalizing to new data. When the generalization error starts to increase, the NN has 'overlearned' the training sample and its ability to generalize is degraded.

The test sample consisted of about 100,000 simulated events, generated by using JETSET PS. The number of events generated in each class corresponded to the Z^0 hadronic branching fractions in the Standard Model. By monitoring the behaviour of the generalization error, the training was stopped after 300,000 updates.

After the network has been trained, its performance can be judged in terms of signal efficiency ε_s (number of events correctly classified as belonging to class q over the total number of events in class q), and purity p (number of correctly classified events in class q over all the events classified as belonging to class q). For example, if the event is classified simply according to the highest output node, one obtains an efficiency for b identification of about 55%, and a purity of about 43%. The efficiency and purity for a single quark class can be improved with a dedicated single output network, but we have chosen to estimate the three quark categories simultaneously. As far as inclusive analyses using single variables are concerned, a better b quark event purity can be obtained by using prompt leptons with a strict cut on the lepton transverse momentum with respect to the jet axis, but the efficiency is limited by the semileptonic branching fraction.

The curve giving purity of the selected sample versus efficiency for the selection of b quark pairs is given in Fig.1, and compared with the result obtained by using impact parameters only. The improvement is evident, and we want to stress, as a concluding remark, that results will be improved substantially by releasing the condition of a strict control of the systematics from detector and physics modelling.

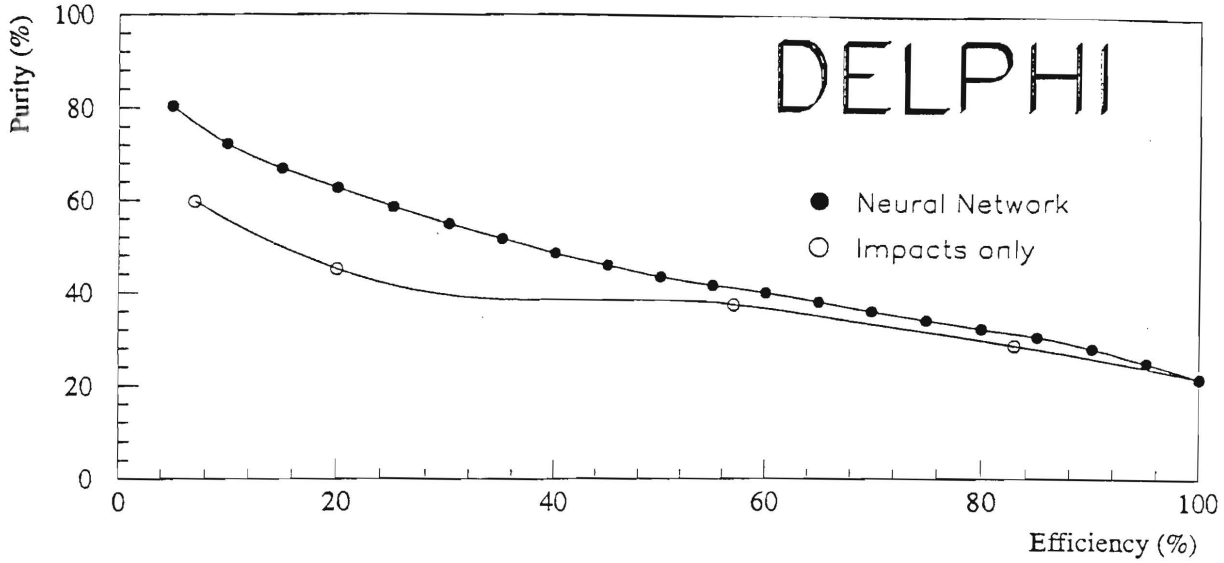

Figure 1: *Purity versus efficiency, from the Neural Network (closed points) and from the input variable related to impact parameters (open points).*

References

[1] L. Lönnblad *et al*, *Nucl. Phys.* **B349** (1991) 675.

[2] C. Bortolotto, A. De Angelis and L. Lanceri, *Nucl. Instr. Meth.* **A306** (1991) 459;
L. Bellantoni *et al*, *Nucl. Instr. Meth.* **A310** (1991) 618.

[3] C. Bortolotto *et al*, "A Measurement of the Partial Hadronic Widths of the Z^0 using Neural Networks", in O. Benhar, C. Bosio, P. Del Giudice and E. Tabet (Eds.), "Neural Networks: From Biology to High Energy Physics", p. 445, ETS Editrice, Pisa (1991). Proceedings of the Workshop held on Isola d'Elba (Italy), June 1991;
P. Henrard (ALEPH), presented at the 4th Symposium on Heavy Flavour Physics, Orsay, June 1991;
B. Brandl (ALEPH), Heidelberg preprint HD-IHEP 92-01, presented at the Second International Workshop on Software Engineering, Artificial Intelligence and Expert Systems for High Energy and Nuclear Physics, L'Agelonde, January 1992.

[4] P. Abreu et al. (DELPHI), "Classification of the Hadronic Decays of the Z^0 into b and C Quark Pairs Using a Neural Network", submitted to Phys. Lett. B.

[5] D.E.Rumelhart, G.E.Hinton and R.J.Williams, "Learning Internal Representations by Error Propagation", in D.E.Rumelhart and J.L.McLelland (Eds.), "Parallel Distributed Processing: Explorations in the Microstructure of Cognition (Vol.1)", MIT Press (1986).