October 1991

LU TP 91-25

Mass Reconstruction with a Neural Network

L. Lönnblad[1]

Deutsches Elektronen-Synchrotron DESY,
Notkestraße 85, 2 Hamburg 52, Germany

C. Peterson[2] and T. Rögnvaldsson[3]

Department of Theoretical Physics, University of Lund
Sölvegatan 14A, S-22362 Lund, Sweden

Physics Letters **B 278**, 181 (1992)

Abstract:

A feed-forward neural network method is developed for reconstructing the invariant mass of hadronic jets appearing in a calorimeter. The approach is illustrated in $W \to q\bar{q}$, where W-bosons are produced in $p\bar{p}$ reactions at SPS collider energies. The neural network method yields results that are superior to conventional methods. This neural network application differs from the classification ones in the sense that an analog number (the mass) is computed by the network, rather than a binary decision being made.

As a by-product our application clearly demonstrates the need for using "intelligent" variables in instances when the amount of training instances is limited.

[1] lonnblad@desyvax (bitnet) lonnblad@apollo3.desy.de (internet)
[2] thepcap@seldc52 (bitnet) carsten@thep.lu.se (internet)
[3] thepdr@seldc52 (bitnet) denni@thep.lu.se (internet)

Introduction

Artificial Neural Networks (ANN) have shown great promise for pattern recognition problems in particle physics. In particular, feed-forward classifiers have been used in off-line jet data analysis for quark/gluon separation and b-quark identification with remarkable success [1, 2, 3, 4, 5]. The principle is straightforward; a set of feature (classifier) units are parameterized as functions of kinematical data in a non-linear way by means of so called sigmoidal functions (tanh). A *training set* is used to fit the parameters (weights). The network then "interpolates" to predict answers when confronted with data it has never seen before (the *test set*). For a general introduction to this field with emphasis on particle physics applications we refer the reader to e.g. ref. [6].

When looking for new particles or resonances one almost always encounters the problem of computing invariant masses out of expected decay products. For example, in the case of the intermediate vector boson W (produced in $\bar{p}p$ collisions) in its hadronic decay channel, $W \to q\bar{q} \to$ hadrons, M_W is reconstructed relativistically from the momenta and assumed masses of the produced hadrons. A problem here is that the hadrons from the q and $\bar{q}$ jets are not necessarily the only ones produced in the collision; there are also remnants from projectile hadrons and hadrons originating from bremsstrahlung. In addition there is of course the problem with noise in the detector.

In this letter we devise a method where a neural network learns to compute a mass from calorimetric information of hadronic decay particles. This kind of neural network application differs from the commonly used classification ones in the sense that the network learns to compute a real number (the mass) and not a set of binary decision values. A variation of the standard back-propagation learning algorithm [7] is used together with an appropriate choice of architecture.

The Monte Carlo Data

The `PYTHIA` Monte Carlo [9] is used to generate data for the process

$$p\bar{p} \to W \to q\bar{q} \to hadrons \tag{1}$$

where the hadrons emerge in a calorimeter detector with variables η, ϕ and $E_\perp$, strongly resembling the CERN SPS UA2 calorimeter with $|\eta| < 2$ and a resolution of 20 and 24 cells in the η- and ϕ-directions respectively.

The simulation includes the modeling of the underlying event [9] with a set of parameters according to the tuning found in reference [11]. The selection of events is made by requiring two jets with $E_\perp > 10$ GeV in the $|\eta| < 1$ region.

Training Set

To prevent the network from learning just one number (M_W), we generate W-bosons with fictive masses for training. These masses are flatly distributed with 300 events per GeV in the interval [40,160] GeV, giving a total of $36 \cdot 10^3$ training examples.

Test Sets

Two different data sets are used for testing; one with 10^4 events using the fictive flat mass distribution between 50 and 150 GeV, and one with a normal mixture of W and Z^0 events with $M_W = 80.0$ GeV and $M_{Z^0} = 91.2$ GeV. The former sample is used for monitoring the generalization performance during training and the latter to get a more realistic impression of how the network would perform on real data.

The Neural Network Architecture and Algorithm

We use a feed-forward architecture with one output node representing M_W, which is linear since it is not encoding a logical variable. Different number of hidden nodes and layers are explored. As expected, two or more hidden layers are needed to achieve peak performance. No further effort is made to optimize the hidden part of the architecture with respect to this task; our main goal is to provide a proof-of-concept for this novel application. Exploring different variants of input encoding and learning strategies for this problem turns out to be illuminating, which is discussed below.

Input Encoding

The following four different representations of the input calorimeter data are explored:

A: **"Raw" calorimeter**. A total of 480 input units, corresponding to the (η, ϕ) cells of the calorimeter, receive the $E_\perp$-values of each cell. These nodes are fully connected to a first hidden layer with $\mathcal{O}(100)$ nodes, which are fully connected to a second hidden layer with $\mathcal{O}(10)$ nodes, giving a total of $\mathcal{O}(10^4 - 10^5)$ weights in the network.

B: **"Raw" Calorimeter with receptive fields**. As above but with the difference that each hidden node in the first hidden layer only covers a certain part of the calorimeter in connectivity (8×8 *receptive fields*). Each receptive field overlaps its neighboring fields and the total number of such fields is 312. Weights from the same relative position within the receptive fields are *linked* [6, 7] such that they are updated identically, which is an effective way of encoding translational invariances. Each field is connected to 3 nodes in the first hidden layer, and the second hidden layer contains $\mathcal{O}(10-10^2)$ nodes. The number of effective weights is thus decreased to $\mathcal{O}(10^4)$.

C: **Tower representation** [3]. Take the $E_\perp$ of the leading cell in the 20×24 matrix and assign it to the first node x_1. Assign the η and ϕ coordinates to x_2 and x_3 respectively. Then take the second leading cell and assign its $E_\perp, \eta$ and ϕ to x_4, x_5 and x_6 and so on for the first 30 calorimeter cells. This gives 90 input units. The hidden layers contained $\mathcal{O}(10-10^2)$ and $\mathcal{O}(10)$ nodes respectively giving $\mathcal{O}(10^3-10^4)$ weights for the network.

D: **"Intelligent" variables**. The idea here is to use our physics knowledge of what may be useful variables for reconstructing the mass. The procedure we use is to first apply the `LUCLUS`[14] jet finding algorithm with parameters tuned such that there are always at least three jets being found. We then construct the invariant mass of the two largest jets (M_{12}), the three largest jets (M_{123}), and the four largest jets (M_{1234}) [4]. The idea is that the invariant mass of the three largest jets is a good estimate of the W-mass in case a gluon has been radiated from one of the quarks. Furthermore, for each jet three variables are constructed, each of which is known to be sensitive to whether the jet stems from a quark or from a gluon [10]. These variables are:

n_{90}: The minimum number of cells needed to account for 90% of the jets $E_\perp$. This is a real number; if 8 cells contain 85% and 9 cells contain 95%, n_{90} is set to 8.5.

n_c: The number of cells in the jet with $E_\perp > 1$ GeV.

s_c: The correlation of transverse energy deposited in neighboring cells in the jet, defined as

$$s_c = \sum_{i,j} \frac{|E_{\perp,i}^2 - E_{\perp,j}^2|}{(E_{\perp,i}^2 + E_{\perp,j}^2 + 1)} \tag{2}$$

where i and j are nearest neighbors.

These variables are also constructed for the whole calorimeter. In addition the $E_\perp$ of each jet as well as the total $E_\perp$ and the total invariant mass of all towers found in the calorimeter is presented to the net, making a grand total of 24 input variables and $\mathcal{O}(10^3)$ weights (see below).

[4] M_{1234} is set to 0 if less than four jets are found.

In all cases the the inputs variables have been rescaled to give an absolute value in the interval [0,1], in order to avoid scaling problems in the weights. We also tried to use more than two hidden layers which in some cases speeded up the training but no significant improvement of the final performance of the net was found.

The variants with no preprocessing (A and B) give poor performance for our application, in contrast to the tower representation (C) and the "intelligent" variables encoding (D). The reason is that raw information requires many weights (parameters) to process, which with limited training data gives rise to poor generalization performance on the test set [12]. In other words one has a situation with too many parameters as compared to training examples (over-fitting). More quantitatively, using raw calorimeter data (A and B) yields generalization results that are comparable to what one obtains with the conventional method described above. In what follows we stick to alternative D above, which gives the best results.

Output Encoding and Learning Strategies

We consistently use a summed square error measure for the back-propagation algorithm in our simulations.

$$E = \sum_p (o^{(p)} - t^{(p)})^2 \tag{3}$$

where $o^{(p)}$ is the output produced by the network for pattern p and $t^{(p)}$ the corresponding target value. It turns out that using the mass directly as target value ($t^{(p)} = M_t^{(p)}$) causes the network to systematically over-estimate the mass, since an error of 10% for a small mass is not as severe as it is for a large mass. What we really want to minimize is the width of the ratio between the reconstructed and the true mass, which can be achieved by instead using the logarithm of the true mass as target value for the network. The network then minimizes

$$E = \sum_p \left\{\log(M_o^{(p)}) - \log(M_t^{(p)})\right\}^2 = \sum_p \left\{\log(\frac{M_o^{(p)}}{M_t^{(p)}})\right\}^2 \tag{4}$$

which for small errors approximately gives the width of the distribution of M_o/M_t

$$E \approx \sum_p \left\{1 - \frac{M_o^{(p)}}{M_t^{(p)}}\right\}^2 \tag{5}$$

In both cases we normalize the target values to be between 0 and 1. In principle we could use a sigmoid response in the output layer. The idea would then be to start out with a low temperature (high gain), forcing the network to divide the patterns into high and low mass ones, then increasing the temperature gradually to fine-tune the net.

In the back-propagation algorithm the weights are changed by gradient descent. An alternative is the so-called *Manhattan* method [13] where the weights are changed by a fixed step size (the learning rate η) that is gradually decreased to 0 during training. The final result from the training is in principle independent of the strategy chosen but the latter has the advantage of being more tolerant to the choice of learning parameters than the former, especially when more than one hidden layer is used or when linear output nodes are used. In our calculations we have allowed the learning rate η to decrease geometrically,

$$\eta(t+1) = \eta(t)(1-\lambda) \tag{6}$$

where λ is a decay parameter.

The Conventional Approach

The "standard" procedure for reconstructing masses is to use a so-called cone algorithm to find two jets, defined as the two circular areas in the η, ϕ plane with radius $R^2 = \Delta\eta^2 + \Delta\phi^2$ with the largest $E_\perp$, and to calculate their invariant mass. In ref. [11] it is found that the optimum choice of R is 0.8. When we in the following compare with the "standard"approach, we use the cone algorithm in JETSET [14] called LUCELL. Although this is slightly different from the one used in ref. [11] it can safely be assumed that the choice of $R = 0.8$ as the optimum value still holds.

Results

Based on the considerations above we use the following architecture and learning strategy:

- **Inputs:** 24 input nodes according to coding strategy D above.
- **Hidden architecture:** Three hidden layers are used with 40 nodes in the first, 24 in the second and 10 in the third hidden layer.
- **Output:** One linear output node is trained to give $0.5(\log M_W - 3.5) \in [0, 1]$ for $M_W \in [40, 160]$ GeV.
- **Initialization:** All weights are initialized randomly in [-0.1,0.1].
- **Temperature:** The temperature for all nodes is set to 1.0.

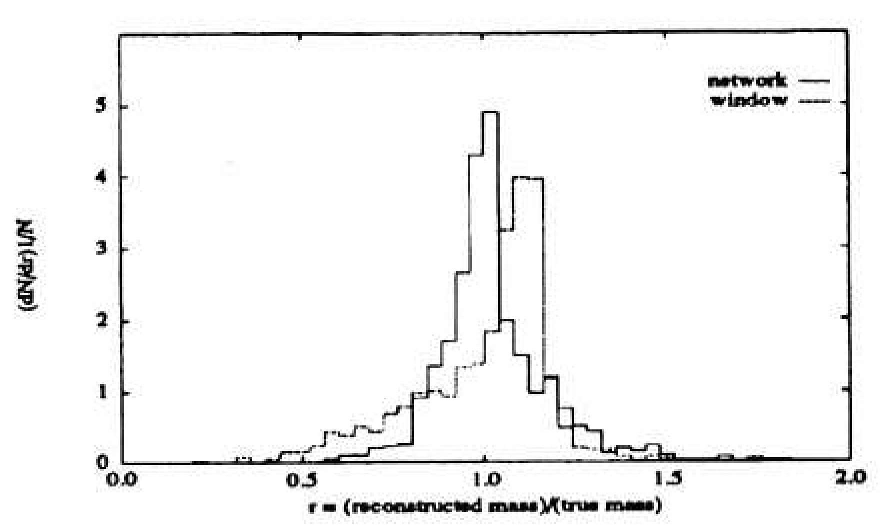

Figure 1: The reconstructed mass ($M_{W,Z}$) divided by the true mass ($M^0_{W,Z}$) using the neural network method (full line) and the conventional "window" method (dashed line) with $R = 0.8$.

- **Updating:** Patterns are selected randomly and the network is updated every 10^{th} pattern using the Manhattan algorithm, starting out with $\eta = 0.01$ and then lowering η with a factor 0.98 every epoch (= 36000 patterns) for 400 epochs.

The `JETNET 2.0` package [8] is used for all the simulations. The performance of the network is monitored using the test set of 10^4 W-events with a flat mass distribution, measuring the width of the distribution in reconstructed mass divided by true mass. The final value of this width for the network is

$$\sqrt{\langle (M_W^{net}/M_W^0)^2 \rangle - \langle M_W^{net}/M_W^0 \rangle^2} = 0.15 \tag{7}$$

as compared with 0.19 when reconstructing the mass with the `LUCELL` jet finding algorithm. In figure 1 this distribution is shown for the test set with a normal mixture of events with W and Z distributed according to their true masses and widths. For `LUCELL` the ratio has been rescaled so that $\langle M_{2jet}/M_W^0 \rangle = 1$. In figure 2 we show the reconstructed W and Z mass distribution for the same test set.

The main reason why the neural network approach does better than the conventional method is that it captures the gluon bremsstrahlung tails well and that it is more resistant to noise.

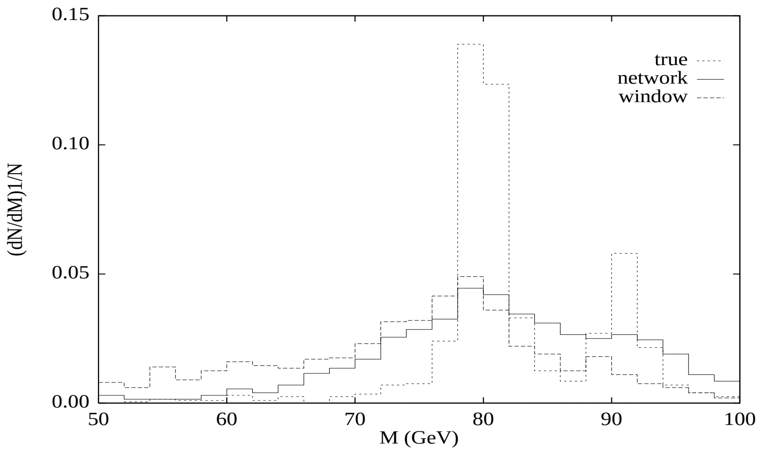

Figure 2: The W and Z^0 mass peak for the true mass spectrum (dotted line), the spectrum reconstructed with the neural network (full line), and the spectrum reconstructed with the conventional "window" method (dashed line).

Summary

We have devised an efficient special purpose analog machine to reconstruct an invariant mass given calorimetric information. Being a neural network it is of "black box" nature and hence requires very little expertise knowledge to use and should be fairly fault tolerant and resistant to detector inefficiencies. The latter are easily included in the training phase.

The performance of the ANN algorithm is better than that of conventional ones; the reconstructed W-mass is more sharply peaked around the correct value and the resulting distribution is more symmetric, indicating that the algorithm handles gluon bremsstrahlung efficiently.

From experimenting with different architectures we strongly confirmed the fact that with limited sized training sets one needs to use preprocessed input data in order to avoid poor generalization (overfitting).

The method developed here could be of utmost importance when it comes to separating $H^0 \rightarrow W^+W^-$ produced at LHC/SSC energies from a background consisting of $W + jets$, $tt \rightarrow W + jets$ and $bb \rightarrow l + jets$. In this case one needs to reconstruct M_W.

References

[1] L. Lönnblad, C. Peterson and T. Rögnvaldsson, "Finding Gluon Jets with a Neural trigger", *Physical Review Letters* **65**, 1321 (1990).

[2] L. Lönnblad, C. Peterson and T. Rögnvaldsson, "Using Neural Networks to Identify Jets", *Nuclear Physics* **B 349**, 675 (1991).

[3] P. Bhat, L. Lönnblad, K. Meier and K. Sugano, "Using Neural Networks to Identify Jets in Hadron-Hadron Collisions", LU TP 90-13 (to appear in *Proc. of the 1990 DPF Summer Study on High Energy Physics Research Directions for the Decade, Colorado*), (1990).

[4] L. Lönnblad, C. Peterson, H. Pi and T. Rögnvaldsson, "Self-organizing Networks for Extracting Jet Features", *Computer Physics Communications* **67**, 193 (1991).

[5] I. Scabai, F. Czakó and Z. Fodor, "Quark and Gluon Jet Separation using Neural Networks", ITP Budapest Report 477 (1990).

[6] C. Peterson and T. Rögnvaldsson, "An Introduction to Artificial Neural Networks", *LU TP 91-23* (lectures given at the 1991 CERN School of Computing, Ystad, Sweden, to be published in the proceedings) (1991).

[7] D.E. Rumelhart, G.E. Hinton and R.J. Williams, in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* Vol. 1, D.E. Rumelhart and J.L. McClelland (Eds.), MIT Press, Cambridge (1986).

[8] L. Lönnblad, C. Peterson and T. Rögnvaldsson, "Pattern Recognition in High Energy Physics with Artificial Neural Networks - JETNET 2.0", *LU TP 91-18* (to appear in *Computer Physics Communications*) [Program and manual available via Email request] (1991).

[9] H.-U. Bengtsson and T. Sjöstrand, "The Lund Monte Carlo for Hadronic Processes: PYTHIA Version 4.8", *Computer Physics Communications* **46**, 43 (1987).

[10] J. Pumplin, "How to Tell Quark Jets form Gluon Jets", *Phys. Rev.* **D44**, 2025 (1991).

[11] J. Alitti et al., "A Measurement of Two-jet Decays of the W and Z Bosons at the CERN $\bar{p}p$ Collider", *Zeitschrift für Physik* **C 49**, 17 (1991).

[12] E. B. Baum and D. Haussler, "What Size Net Gives Valid Generalization", *Neural Computation* **1**, 151-160 (1989)

[13] See e.g. C. Peterson and E. Hartman, "Explorations of the Mean Field Theory Learning Algorithm", *Neural Networks* **2**, 475 (1989).

[14] T. Sjöstrand, JETSET 7.3 program and manual see e.g. B. Bambah et. al., QCD Generators for LEP, CERN-TH.5466/89 (1989);
T. Sjöstrand, "The Lund Monte Carlo for Jet Fragmentation and $e^+ e^-$ Physics - JETSET version 6.2", *Computer Physics Communications* **39**, 347 (1986);
T. Sjöstrand and M. Bengtsson, "The Lund Monte Carlo for Jet Fragmentation and $e^+ e^-$ Physics - JETSET version 6.3 - An Update", *Computer Physics Communications* **43**, 367 (1987).