March 1990

LU TP 90-3

Finding Gluon Jets with a Neural Trigger

Leif Lönnblad[1], Carsten Peterson[2] and Thorsteinn Rögnvaldsson[3]

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

Physical Review Letters **65**, 1321 (1990)

Abstract:

Using a neural network classifier we are able to separate gluon from quark jets originating from Monte Carlo generated e^+e^- events with $85-90\%$ accuracy.

PACS numbers: 13.65.+i, 12.38Qk, 13.87.Fh

[1] thepll@seldc52 (bitnet) leif@thep.lu.se (internet)
[2] thepcap@seldc52 (bitnet) carsten@thep.lu.se (internet)
[3] theptr@seldc52 (bitnet) denni@thep.lu.se (internet)

Motivation and Background

In this letter we demonstrate how to separate gluon and quark jets using a neural network identifier. Being able to distinguish the origin of a jet of hadrons is important from many perspectives. It can shed experimental light on the confinement mechanism in terms of detailed studies on the so-called string effect [1] and related issues. Also a fairly precise identification of the gluon jet is required for establishing the existence of the 3-gluon coupling in e^+e^--annihilation [2]. To date the gluon jet identification has been done by making various cuts on the kinematic variables ranging from just identifying the jet with smallest energy as the gluon jet [1] to more elaborate schemes [3, 4]. Such procedures are often based on the entire event rather than just a single isolated jet. It would be desirable to focus on the latter alternative given that in many situations "global" quantities like total jet energies are less well-known. One such example is jets produced in high p_T hadron-hadron collisions.

A straight-forward method for identifying the jets would be to find the functional mapping between the observed hadronic kinamatical information and the feature (quark/gluon). This reduces the problem from an expert's exercise to a "black box" fitting procedure. This is exactly what the neural network approach aims at. It has the advantage over other fitting schemes in that it is very general, inherently parallel and easy to implement in custom made hardware with its simple processor structure. The latter feature is very important for real-time triggering.

We confine our studies to Monte Carlo generated e^+e^- events using the Lund MC. To some extent this induces a "chicken-and-egg" effect to our studies; some of the physics one wants to study is already there. This effect can be minimized by limiting ourselves to kinematical quantities that are most model independent, e.g. considering the fastest particles only.

Although this paper is limited to the separation of gluon and quark jets it is clear that the methodology could be used in a variety of different triggering situations.

The Neural Network Learning Algorithm

The basic ingredients in a neural network are **neurons** n_i and connectivity **weights** ω_{ij}. For feature recognition problems like ours the neurons are often organized in a feed-forward layered architecture (see fig. 1) with input (x_k), hidden (h_j) and output (y_i) nodes. Each neuron performs a weighted sum of the incoming signals and thresholds this sum with a "sigmoid" function $g(x) = 0.5(1 + tanh(x))$. For the

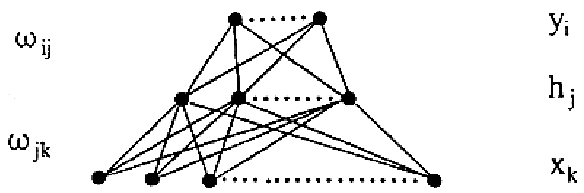

Figure 1: A feed-forward neural network with one layer of hidden units

hidden and output neurons one has

$$h_j = g(a_j/T) \tag{1}$$

$$y_i = g(a_i/T) \tag{2}$$

where the "temperature" T sets the slope of g and the weighted input sums a_j and a_i are given by $\sum_k \omega_{jk} x_k$ and $\sum_j \omega_{ij} h_k$ respectively.

The hidden nodes have the task of correlating and building up an "internal representation" of the patterns to be learned. Training the network corresponds to changing the weights ω_{ij} such that a given input pattern $\vec{x}^{(p)}$ gives rise to an output (feature) value $\vec{y}^{(p)}$ that equals the desired output or target value $\vec{t}^{(p)}$. A frequently used procedure for accomplishing this is the **back-propagation** learning rule [5] where the error function

$$E = \frac{1}{2} \sum_p \sum_i (y_i^{(p)} - t_i^{(p)})^2 \tag{3}$$

is minimized. Changing ω_{ij} by gradient descent corresponds to [5]

$$\Delta\omega_{ij} = \eta \delta_i h_j + \alpha \Delta\omega_{ij}^{old} \tag{4}$$

for the hidden to output layers, where δ_i is given by

$$\delta_i = (t_i - y_i) g'(a_i) \tag{5}$$

Correspondingly for the input to hidden layers one has

$$\Delta\omega_{jk} = \eta \sum_i \omega_{ij} \delta_i x_k + \alpha \Delta\omega_{jk}^{old} \tag{6}$$

In eqs. (4,6) η is a learning strength parameter and we have also included so-called momentum terms $\alpha\Delta\omega_{ij}^{old}$ and $\alpha\Delta\omega_{jk}^{old}$ in order to damp out oscillations. This procedure is repeated for each pattern p until the network has learned all patterns to a satisfactory level.

What is happening in the network is that sigmoids are being added up, which with alternating signs emulate 'bumps'. Adding 'bumps' is very similar to decomposing an arbitrary function into Gaussians, which is a well-known method for finding a functional mapping.

The Monte Carlo Data

The jets used to test our approach were taken from Monte Carlo generated e^+e^-events at different energies using the JETSET 7.2 [6] and ARIADNE 3.1 [7] programs. In the ARIADNE program, the partonic phase of the events is simulated using the colour dipole approximation [8], producing partons which are then fragmented into hadrons using the string fragmentation model [9] as it is implemented in the JETSET program. In JETSET the formed hadrons are also allowed to decay. All parameters in these two programs were set to their default values. This colour dipole approximation is dual to the parton shower approach [10], which is the default option in JETSET 7.2, and it generates the same physics.

We then used a clustering algorithm to find the jets in each event. In this algorithm, all particles that are "closer" to each other than a certain cutoff are clustered together into jets. The "closeness" can be defined in many ways. We have preferred to use the relative transverse momentum which is the default option in the algorithm LUCLUS in JETSET 7.2.

Having produced the jets we define those two which are "closest" (in the same respect as in the clustering above) to the initial quark and anti-quark respectively to be quark jets, and the rest of the jets in the event to be gluon jets.

We have studied events of two different center of mass energies; 29 GeV and 92 GeV. For each energy we generated two different sets of events.

- **Forced 3-jet events**. Exactly three jets are required with angles between the jets larger than 30°. No cut in the clustering algorithm.
- **Multijet events**. The cut in the clustering algorithm set to 2.5 GeV, requiring at least three jets.

For both alternatives we also required that each jet consisted of more than four particles and that their energy was larger than one tenth of the total center of mass energy.

In addition we generated one set of events at 55 GeV with the same clustering and cuts used in ref. [3] for more direct comparison with experimental situations.

Results

For each of the different data sets we use two different approaches of presenting the jets to the network. One is to show only the four-momenta $(\vec{p}_k, E_k)$ of the four leading particles in the jet. In this way we do not reveal too much about the structure of the fringe of the jets. Hence e.g. the string effect can be studied in a fairly model-independent way. In the other approach we show only the total energy and momentum of the jet together with the four-momentum of its leading particle. The model-dependency is thus reduced when studying details of the jet structure, such as asymmetries. The network sees only one jet at a time and knows nothing about the total number of jets in the event or their relative spatial orientation.

In the first case we use a three-layered (see fig. 1) feed forward network with 4×4 input nodes, 6 hidden units and one output unit. In the second case we use a network with 6 input nodes, 6 hidden units and one output unit. The network performance is not very sensitive to the number of hidden units. The output unit is used to code the jet identity; 1 for gluon and 0 for quark. We use strict middle point condition; if the output is > 0.5 the jet is considered to be a gluon jet and if the output is < 0.5 the jet is considered to be a quark jet.

Each data set is divided into two parts; one that is used for training the network (**training set**) and one that we use for testing the ability of the network to classify the jets (**test set**). We thus make sure that the network is tested on patterns that it has never seen before and that we really measure its ability to generalize. The weights in the network are initialized at random with values in the range [-0.1,0.1]. The network is then trained by taking (at random) 10 jet-patterns, half gluons and half quarks, from the training set. These 10 patterns are then run through the network whereupon the weights are updated and 10 new patterns are taken, and so forth.

In most cases the energy for the leading particle in a gluon jet is lower than for a quark jet. The network sees this very quickly and is able to correctly predict 85% of the jets after seeing about 4000 patterns, only 1/2 of the total training set (se fig. 2). The wrong classifications are equally distributed among quark and gluon jets. Leaving out the jets originating from heavy c- and b-quarks makes it a bit easier for the network to separate quark jets from gluon jets (see tables 1 and 2). Heavy quark jets in general are a little broader than light ones due to the decay of heavy

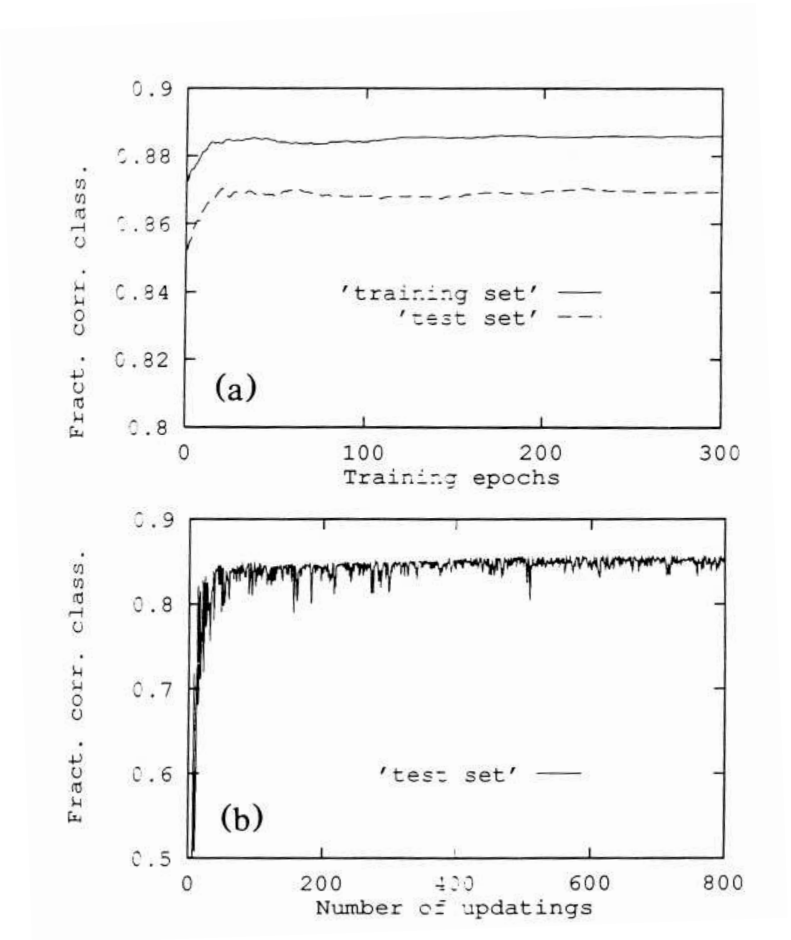

Figure 2: (a). Network performance as a function of training epochs using a "multiple jets" data set with heavy quarks and a center of mass energy of 29 GeV. The four-momenta of the leading four particles as input. (b). The first epoch displayed in detail. The network is updated for every 10 patterns. The parameters used are α=0.5, T=1 and η=0.0005.

mesons, they are therefore more difficult to separate from the gluon jets whose prime characteristic is that they are broader than quark jets. Heavy quark jets are, on the other hand, more easy to separate with other methods (looking for semi-leptonic decays etc.) so we do not feel that leaving them out introduces any severe handicap for our method. The network also performs better if it trains on a "multiple jet" set than if it trains on a "forced 3-jet" set. This is expected since the forced 3-jet clustering certainly produces both broad quark jets and heavy gluon jets.

We also used the network to tell which one of the jets in a three-jet event (from the 55 GeV jet set) that originated from the gluon (we chose the jet with output closest to 1 as the gluon jet). The network was correct in 87% of the events. Remarkable

$\sqrt{s}$	set type	training set/ test set	with heavy quarks	without heavy quarks
92 GeV	forc. 3-jet	12000/6000	86%	87%
92 GeV	mult. jets	10000/8000	88%	90%
29 GeV	forc. 3-jet	12000/6000	86%	87%
29 GeV	mult. jets	11000/7000	87%	88%
55 GeV	ref. [3]	12000/6000	89%	-

Table 1: Percentage correct classifications of jets using the four-momenta of leading four particles as input. The parameters used to obtain these results were α=0.5, T=1 and η=0.0001. The network was run for 100,000 learning passes.

$\sqrt{s}$	set type	training set/ test set	with heavy quarks	without heavy quarks
92 GeV	forc. 3-jet	12000/6000	86%	87%
92 GeV	mult. jets	10000/8000	89%	91%
29 GeV	forc. 3-jet	12000/6000	87%	87%
29 GeV	mult. jets	11000/7000	88%	89%
55 GeV	ref. [3]	12000/6000	91%	-

Table 2: Percentage correct classifications of jets using the total energy and momentum of the jet plus the four-momenta of the leading particle as input. Parameters: α=0.5, T=1 and η=0.0001. The network was run for 100,000 learning passes.

is that it hardly mattered if the network had been trained on 92 GeV, 55 GeV or 29 GeV events, it did well anyway, indicating that the signature of quark and gluon jets scales.

Summary and Outlook

We have successfully demonstrated how neural network techniques can be used to distinguish between gluon and quark jets in $e^+e^- \rightarrow hadrons$. After learning the network successfully identifies $85-90\%$ of jets it has never seen before. This amazingly simple non-expert procedure thus yields a world record score on this problem! However, it should be stressed that a similar order of magnitude performance presumably could be obtained with another kind of expansion. But, we reiterate the opinion that the neural network approach has a clear edge given its general structure, parallelism and closeness to simple hardware. The question of how well the network in principle can perform given a certain data set naturally arises. The upper limit of performance is given by the *Bayesian limit* [11], which is obtained from the minimal overlap between the two multidimensional distributions. An estimate of this limit can be obtained numerically within realistic CPU consumption by reducing the accuracy of the kinematical variables. Work in this direction is underway [12].

It will be interesting to see whether our successes survive if as training and test sets we use mixtures of events created by different Monte Carlo programs [12], e.g. JETSET [6] and HERWIG [13].

Also, an upcoming challenge is to apply our method to jets formed in hadron-hadron collisions [12]. Future work will also analyze the role of the hidden units in terms of generating rules for the decisions.

References

[1] JADE Collaboration, W. Bartel et. al., *Z. Phys.* **C21** (1983) 37.

[2] M. Bengtsson and P. Zerwas, *Phys. Lett.* **B208** (1988) 306.

[3] Y. K. Kim et. al.,*Phys. Rev. Lett.* **63** (1989) 1772.

[4] L. Jones, "Towards A Systematic Jet Classification", University of Illinios preprint ILL-(TH)-90-01.

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

[6] T. Sjöstrand, JETSET 7.2 program and manual. See B. Bambah et al., QCD Generators for LEP, CERN-TH.5466/89.

[7] L. Lönnblad, "ARIADNE-3, A Monte Carlo for QCD Cascades in the Colour Dipole Formulation", Lund preprint LU TP 89-10.

[8] G. Gustafson, *Phys. Lett.* **B175** (1986) 453;
G. Gustafson, U. Pettersson, *Nucl. Phys.* **B306** (1988) 746;
B. Andersson, G. Gustafson, L. Lönnblad, "Gluon Splitting in the Colour Dipole Formulation", Lund Preprint LU TP 89-9 (to be published in *Nucl. Phys.***B**).

[9] B. Andersson and G. Gustafson, *Z. Phys.* **C3** (1980) 223;
B. Andersson, G. Gustafson, G. Ingelman, T. Sjöstrand,
Phys. Rep. **97** (1983) 31.

[10] M. Bengtsson and T. Sjöstrand, *Phys. Lett.* **B185** (1987) 453; *Nucl. Phys.* **B289** (1987) 810.

[11] See e.g. R. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons (1973).

[12] L. Lönnblad, C. Peterson and T. Rögnvaldsson, work in progress.

[13] G. Marchesini and B.R. Webber, *Nucl. Phys.* **B310** (1988) 461;
I.G. Knowles, *Nucl. Phys.* **B310** (1988) 571.