Finding Gluon Jets with a Neural Trigger

Leif Lönnblad,[(a)] Carsten Peterson,[(b)] and Thorsteinn Rögnvaldsson[(c)]
Department of Theoretical Physics, University of Lund, Sölvegatan 14A, S-22362 Lund, Sweden
(Received 6 April 1990)

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

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 three-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 straightforward method for identifying the jets would be to find the functional mapping between the observed hadronic kinematical information and the feature (quark or 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 Monte Carlo model. 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 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 $\sum_j \omega_{ij} h_j$, 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 parameter $\mathbf{x}^{(p)}$ gives rise to an output (feature) value $\mathbf{y}^{(p)}$ that equals the desired output or target value $\mathbf{t}^{(p)}$. A frequently used procedure for accomplishing this is the *back-propagation* learning rule[5] where the error function

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

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

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

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

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

Correspondingly, for the input to hidden layers one has

$$\Delta\omega_{jk} = -\eta\sum_i \omega_{ij}\delta_i g'(a_j)x_k + \alpha\Delta\omega_{jk}^{\text{old}}\,. \tag{6}$$

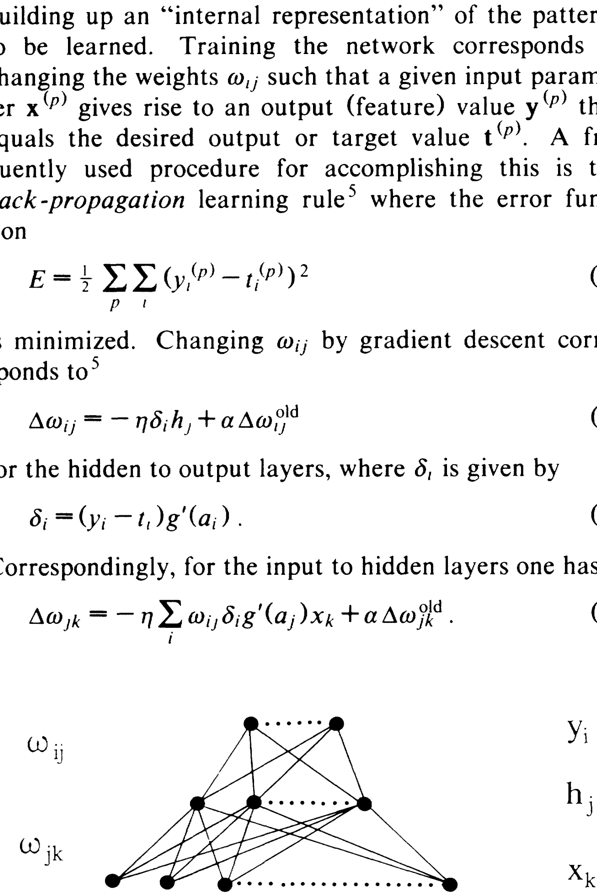

FIG. 1. A feed-forward neural network with one layer of hidden units.

In Eqs. (4) and (6) η is a learning strength parameter and we have also included so-called momentum terms $\alpha\Delta\omega_{ij}^{\text{old}}$ and $\alpha\Delta\omega_{jk}^{\text{old}}$ 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 (Ref. 6) and ARIADNE 3.1 (Ref. 7) programs. In the ARIADNE program, the partonic phase of the events is simulated using the color-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 color-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 antiquark, 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 and 92 GeV. For each energy we generated two different sets of events.

Forced three-jet events. Exactly three jets are required with angles between the jets larger than 30°. There is no cut in the clustering algorithm.

Multijet events. The cut in the clustering algorithm is 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.

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 $(\mathbf{p}_k, E_k)$ 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 1 output unit. In the second case we use a network with 6 input nodes, 6 hidden units, and 1 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 a 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

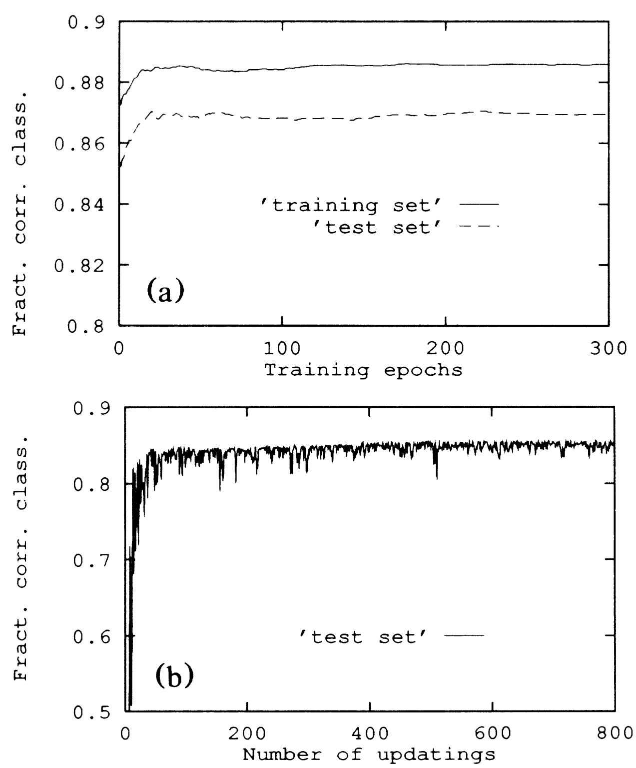

FIG. 2. (a) Network performance as a function of training epochs using a "multiple-jet" data set with heavy quarks and a center-of-mass energy of 29 GeV. The four-momenta of the leading four particles as input. (b) The first epoch displayed in detail. The network is updated for every 10 patterns. The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

TABLE I. Percentage correct classifications of jets using the four-momenta of leading four particles as input. The parameters used to obtain these results were $\alpha=0.5$, $T=1$, and $\eta=0.0001$. The network was run for 100000 learning passes.

$\sqrt{s}$ (GeV)	Set type	Training set/ test set	With heavy quarks (%)	Without heavy quarks (%)
92	Forced three jet	12000/6000	86	87
92	Multiple jets	10000/8000	88	90
29	Forced three-jet	12000/6000	86	87
29	Multiple jets	11000/7000	87	88
55	Reference 3	12000/6000	89	...

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 $\frac{1}{2}$ of the total training set (see Fig. 2). The wrong classifications are approximately 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 I and II). Heavy quark jets in general are a little broader than light ones due to the decay of heavy 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 semileptonic 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 three-jet set. This is expected since the forced three-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. It is remarkable that it hardly mattered if the network had been trained on 92-, 55-, or 29-GeV events; it did well anyway, indicating that the *relative* signature of quark and gluon jets scales.

In order to investigate the model dependence of our results we have also considered two other Monte Carlo (MC) models,[11] JETSET 7.2 (parton shower)[6] and HERWIG (clustering).[12] It turns out that if the network is trained on a data set generated by one model and then tested on data from another model, at most 1%–2% in performance is lost. In other words, the procedure is very independent of the MC model used.

All exercises so far have taken place in an ideal world with no acceptance limitations, all particles detected, etc. In order to get a feeling for how our approach might work under real experimental conditions, the MC models have been processed through the DELPHI detector simulator.[13] We find only a modest degradation in performance ($\sim 3\%$), which is very encouraging.

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. The corresponding rate using assignment based on energy[1] with our data set is $\sim 65\%$.

TABLE II. 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: $\alpha=0.5$, $T=1$, and $\eta=0.0001$. The network was run for 100000 learning passes.

$\sqrt{s}$ (GeV)	Set type	Training set/ test set	With heavy quarks (%)	Without heavy quarks (%)
92	Forced three jet	12000/6000	86	87
92	Multiple jets	10000/8000	89	91
29	Forced three jet	12000/6000	87	87
29	Multiple jets	11000/7000	88	89
55	Reference 3	12000/6000	91	...

This amazingly simple nonexpert 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,[14] 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 under way.[11]

Also, an upcoming challenge is to apply our method to jets formed in hadron-hadron collisions.[11] Preliminary results here indicate a 70% generalization capability, which is lower than in the e^+e^- case. We interpret this performance difference as due to simple correlation between jet energy and quark or gluon identity from the e^+e^- matrix element which is absent in hadron-induced reactions.

[(a)]Electronic addresses: thepll@seldc52 (bitnet); leif@thep.lu.se (internet).

[(b)]Electronic addresses: thepcap@seldc52 (bitnet); carsten@thep.lu.se (internet).

[(c)]Electronic addresses: thepdr@seldc52 (bitnet); denni@thep.lu.se (internet).

[1]JADE Collaboration, W. Bartell *et al.*, Z. Phys. C **21**, 37 (1983).

[2]M. Bengtsson and P. Zerwas, Phys. Lett. B **208**, 306 (1988).

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

[4]L. Jones, Phys. Rev. D **42**, 811 (1990).

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

[6]T. Sjöstrand, JETSET 7.2 program and manual; see B. Bambah *et al.*, CERN Report No. CERN-TH.5466/89 (unpublished).

[7]L. Lönnblad, Lund University Report No. LU TP 89-10 (to be published).

[8]G. Gustafson, Phys. Lett. B **175**, 453 (1986); G. Gustafson and U. Pettersson, Nucl. Phys. **B306**, 746 (1988); B. Andersson, G. Gustafson, and L. Lönnblad, Nucl. Phys. B (to be published).

[9]B. Andersson and G. Gustafson, Z. Phys. C **3**, 223 (1980); B. Andersson, G. Gustafson, G. Ingelman, and T. Sjöstrand, Phys. Rep. **97**, 31 (1983).

[10]M. Bengtsson and T. Sjöstrand, Phys. Lett. B **185**, 453 (1987); Nucl. Phys. **B289**, 810 (1987).

[11]L. Lönnblad, C. Peterson, and T. Rögnvaldsson, Lund University Report No. LU TP 90-8 (to be published).

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

[13]O. Barring (private communication); DELSIM, Delphi event generator and detector simulator [DELPHI Report No. 89-67(68) (unpublished)].

[14]See, e.g., R. Duda and P. E. Hart, *Pattern Classification and Scene Analysis* (Wiley, New York, 1973).