DD

DESY 95-061
April 1995

Application of Artificial Neural Networks in Particle Physics

H. Kolanoski
Institut für Physik, Humboldt-Universität, Berlin

ISSN 0418-9833

DESY 95-061
April 1995

ISSN 0418-9833

Application of Artificial Neural Networks in Particle Physics*

Hermann Kolanoski†
Humboldt-Universität zu Berlin, Germany

Abstract

The application of Artificial Neural Networks in Particle Physics is reviewed. Most common is the use of feed-forward nets for event classification and function approximation. This network type is best suited for a hardware implementation and special VLSI chips are available which are used in fast trigger processors. Also discussed are fully connected networks of the Hopfield type for pattern recognition in tracking detectors.

1 Introduction

The sensory and cognitive abilities of biological neural networks, like the human brain, are still not reached by even the most powerful electronic computer systems. The fascinating features of neural systems which are most distinct from the properties of the conventional 'von Neumann' computers are:

- the associative recognition of complex structures;
- fault tolerance: the data may be incomplete, inconsistent or noisy;
- the systems are trainable, i.e. they can learn as well as organize themselves;
- algorithms and hardware are inherently parallel.

There is an increasing interest in understanding the working principles of neural systems and to apply these principles to information processing. In this effort scientists from different fields, like biology, psychology, physics and mathematics, are working together. Inspired from the biological model neural and evolutionary algorithms have been developed and successfully used for the analysis of complex problems. Together with specific 'neural' hardware developments such algorithms now begin to find industrial applications for cognitive and sensoric tasks in image processing, robotics and process control. The industrial interest ensures that the hardware development will proceed, which is crucial for the most efficient use of neural algorithms.

Neural information processing is a relatively young field and one is still at the stage of evaluating where artificial neural networks have advantages over more conventional methods. It is certainly important that we now follow these developments and try to assess the possibilities

*Invited talk given at the Vienna Wire Chamber Conference 1995.

†Supported by Bundesministerium für Bildung und Forschung under contract numbers 056DO57P and 056DO50I.

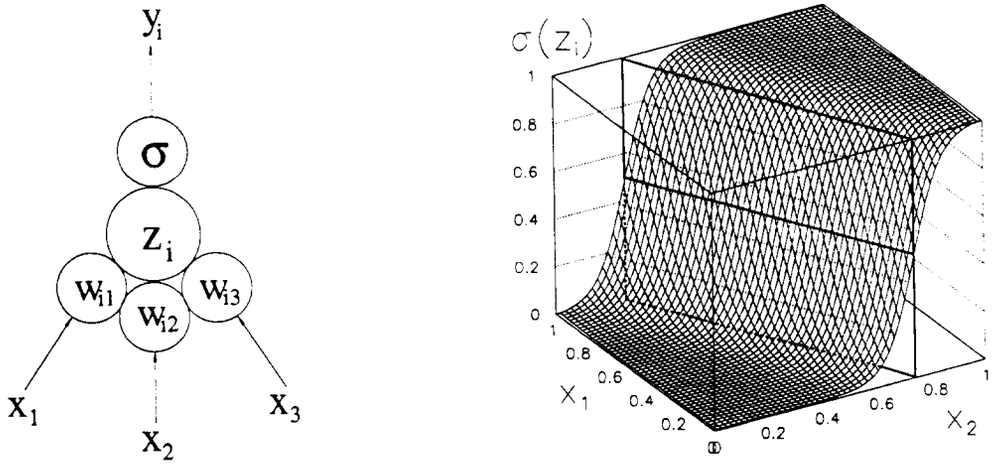

Figure 1: Left: structure of an artificial neuron; right: example for the output of a neuron with two input variables. The value $\sigma(z_i) = 0.5$ defines a hyperplane in the input space.

for applications in Particle Physics. For online applications in experiments we are particularly dependent on dedicated hardware. However, one should keep in mind that the available technology limits drastically the complexity of artificial neural networks – far below the level of the biological models we are looking at. To some extent this deficit is compensated by the higher speed of electronic circuits.

In the following I want to review the application of Artificial Neural Networks (ANN) in Particle Physics [1] – sharing the enthusiasm of those engaged in this field but hopefully critical enough to see the current limitations of the method. The list of papers on ANN in Particle Physics is much too long that it could be covered in this talk. I also apologize if the selection of examples is biased towards the work which is better known to me.

2 Biological and Artificial Neural Networks

Biological information processing is characterized by a huge number of processors with highly complex connections. The processor units are the neurons which have a density of about 10^5 per mm^2 in the cerebral cortex. A simple model describes how neurons work: a neuron receives the signals from many other neurons via synapses which weight the signals. The neuron cell is activated by the sum of the weighted signals and if the activation exceeds some value the neuron sends a signal to the neurons it is connected to.

The structure of an artificial neuron is based on this simple model (fig. 1): The inputs x_j to the neuron i are multiplied by the weights w_{ij} leading to the activation z_i of the neuron. The activation function is usually a linear function of the inputs:

$$z_i = \sum_j w_{ij} x_j - s_i \tag{1}$$

where s_i is an offset or a threshold. The neuron activation determines the output y_i by means of the transfer function which could be just the step function $\Theta(z_i)$ or more often a differentiable, smoothed version of the step function ('sigmoid' function) like the 'logistic' function:

$$\sigma(z_i) = \frac{1}{1 + e^{-z_i}} \tag{2}$$

For the understanding how neurons and neural nets work it is helpful to realize that the fixed value $z_i = 0$ of the activation function defines a hyperplane which divides the input space

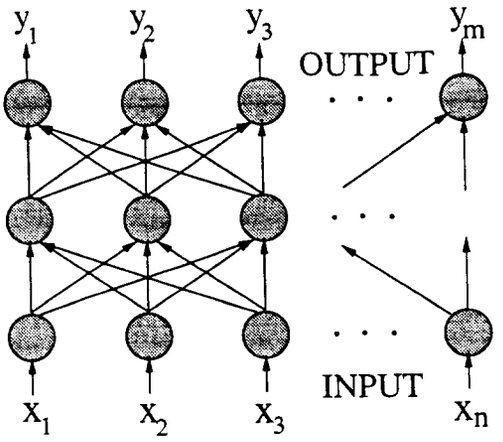

Figure 2: Example of a feed-forward network with one hidden layer.

into hemispheres with negative and positive activations (fig. 1). The neuron output $\sigma(z_i)$ is a measure for the distance from the hyperplane. For larger distances – the scale is given by the absolute values of the weights – the transfer function saturates, i. e. the output measures the neighbourhood of the hyperplane with higher resolution.

In general one has to enclose volumina in the input space. A volume can be approximated by a combination of several hyperplanes, i. e. by several neurons whose output is processed in a subsequent step by another neuron. This leads to a multi-layer network, in most cases so-called feed-forward nets meaning that the signals propagate only in one direction and in a fixed time sequence through the net (see sect. 3).

An important feature of neural networks is their trainability and the capability of self-organisation. Training should establish a functional dependence, i. e. correlate the input of a net with the output by changing the weights and/or the connectivity between the neurons. According to the 'Hebb rule' the learning and structure formation in the brain depends on the intensity and frequency of the stimulus acting on a neuron and on the correlation between the input and the output of the neuron. At the beginning of the training the input - output correlation is just random and then evolves with time, leading to a self-organized structure ('unsupervised learning'). However, in many practical applications of ANN the weights are adjusted according to a predefined output target ('supervised learning'), as for the feed-forward nets discussed in the following.

Self-organizing networks are often used for image processing, process control and robotics and they have also been investigated for applications in Particle Physics. However, in most cases studied in Particle Physics these networks were not found to offer any advantage over the feed-forward nets applied to the same problem. Given the restriction in the length of this write-up I will therefore not discuss self-organizing networks and the similarly interesting techniques of evolutionary formation of network structures.

3 Feed-forward Networks

Network Architecture and Backpropagation Training

From the large variety of neural net models the feed-forward (FF) nets are the most commonly used in Particle Physics. The network has a layered structure (fig. 2), with an input layer, an in principle arbitrary number of 'hidden' layers and an output layer. The signals are always

sent to the next layer above (feed-forward). In practice the number of hidden layers is mostly 1 or 2.

Mathematically the network maps the n-dimensional input vector $\vec{x}$ unto the m-dimensional output vector $\vec{y}$ where the mapping is defined by the structure of the network and the weights (w_{ij}^k is the weight from the ith node in the kth layer to the jth node in the $(k+1)$th layer):

$$\vec{y} = \vec{y}(\vec{x}|w_{ij}^k) \tag{3}$$

The network is trained with N input patterns $\vec{x}^p$ $(p = 1, \ldots, N)$ to approximate the expected output $\vec{y}^p$ ('supervised learning'). This is done by adjusting the weights such that the error function

$$E = \frac{1}{2} \sum_{p=1}^{N} \sum_{i=1}^{m} (y_i{}^p - \hat{y}_i{}^p)^2 \tag{4}$$

is minimized. The minimization is done in an iteration loop by means of the gradient descent method $(0 < \eta < 1)$:

$$\Delta w_{ij}^k = -\eta \frac{\partial E}{\partial w_{ij}^k}. \tag{5}$$

The most common iteration procedure is the backpropagation algorithm which allows to 'back-propagate' the corrections from the output to the input layer. The algorithm is particularly well suited for implementation on computers because it is recursive and local which means that in any layer only the results from the previous layer are required. Since the training is in general very time-consuming the optimization of training algorithms is an intensive field of research.

Examples of public domain software for network training are the program packages JETNET, Aspirin/MIGRAINES, SNNS (Stuttgart Neural Net Simulator) and many others. Information on available software and a lot more can be obtained via World Wide Web (WWW: http://www1.cern.ch/NeuralNets/nnwInHep.html).

Classification with Feed-forward Nets

In Particle Physics FF nets have been mainly used to solve classification problems, such as for separating light quark jets from heavy quark jets, for particle identification or for rejecting background by a trigger decision. In such cases each event, characterized by its vector $\vec{x}$ in pattern space, is assigned to a class C_j which the net has to find out. The net is trained with events of known classification such that the expected value $\hat{y}_j$ of the jth output is 1 if the event belongs to C_j, else 0.

If the classes are well separated the trained net can assign the events to the proper classes with 100 % efficiency provided the net has enough degrees of freedom, i. e. enough neurons. The following is important for judging the necessary degrees of freedom of a network: By virtue of the non-linear transfer functions the combination of hyperplanes defined in the hidden layer yields smooth boundaries of the class volumes rather than mere polyeders. It should be emphasized that even with a smoothing transfer function the decision boundaries are sharp and well defined: FF nets are deterministic!

Bayes discriminator: If the classes have overlapping distributions it is in principle no longer possible to classify with 100 % efficiency. A FF net trained with target values 0 and 1 assigns each event to the class for which it has the highest probability. If the net was trained with n_i events of class i it can be shown that the network outputs approximate the Bayesian discrimi-

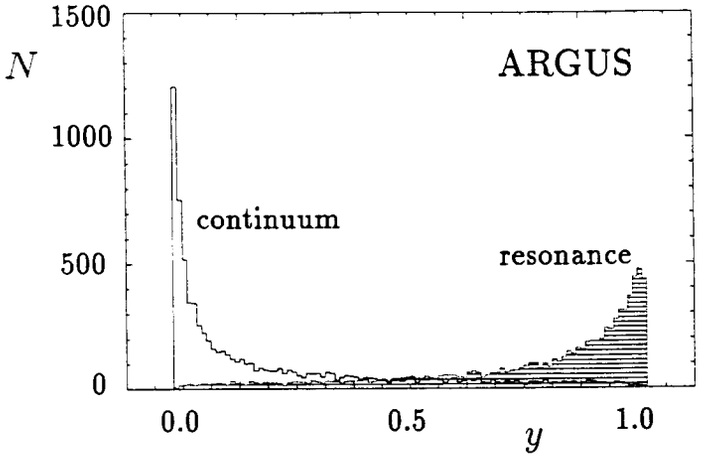

Figure 3: Network output for simulated $\Upsilon(4S)$ and continuum data.

nator:

$$y_i(\vec{x}) = \frac{n_i\, p_i(\vec{x})}{\sum_{j=1,m} n_j\, p_j(\vec{x})}. \tag{6}$$

The maximal y_i belongs to the class C_i for which the event has the highest probability. The advantage of ANN is that the probability densities $p_i(\vec{x})$ need not to be known explicitely because they are learnt by training using real or simulated data.

Applications of Feed-forward Nets

Selection of $\Upsilon(4S) \rightarrow B\overline{B}$ events: An example for an offline application of FF nets is the selection of $B\overline{B}$ events at the $\Upsilon(4S)$ resonance in e^+e^- reactions [3]. The background comes mainly from continuum $q\overline{q}$ events which have a two-jet topology while the $B\overline{B}$ events are more isotropic. However, at these relatively low energies the jets are not very pronounced and all kinematical variables (jettiness, particle multiplicities, number of leptons from semi-leptonic decays, ...) differ only slightly. Differences become only significant if the multi-dimensional correlations between the variables are exploited. In this case, a network with 20 inputs, 20 hidden nodes and 1 output, referred to as 20 - 20 - 1 net configuration, leads to a good separation of resonance and continuum events. The net output of both classes is shown in fig. 3. With the cut on the output the selection efficiency and background suppression can be tuned. For an efficiency of 80 %, e. g., the background is suppressed by a factor 4, yielding a signal-to-background ratio of 1:1 at the resonance.

The Level-2 Trigger of the H1 Experiment: As an example for a real-time trigger application employing dedicated hardware we discuss the level-2 trigger based on FF networks which is currently developed for the H1 experiment at HERA [4]. Using the data available at this trigger level (from calorimetry, tracking and others) several neural nets are trained to select the wanted physics events and reject the background, mainly from beam-gas and beam-wall interactions. The concept is to have one net for each physics process (e.g. charged current reaction, photoproduction etc.) and in addition a positive background identification ('background encapsulator'). The used hardware (based on the CNAPS chip, see sect. 4) suggests networks of up to 64 inputs and 64 hidden nodes.

Extensive studies have been performed to optimize the network architecture, the choice of variables and input data (real or simulated), learning strategies, efficiencies and many other topics [4]. Since the whole collaboration had to be convinced that a neural net is not a mysterious black box particular emphasis was put on understanding how the network works. Questions which have been tackled are, for instance, the importance of each input variable, the visualisation of the network decision, or efficiencies in 'kinematical corners' (which may be less frequently trained).

Analysis of shower clusters: There are many examples which demonstrate that neural nets can be very helpful for the analysis of calorimeter showers. The Crystal Barrel Collaboration, for instance, discriminates low-energy photons against 'split-offs', i.e. fluctuations from larger showers which look like single photons [5]. The ANN algorithm is better than the previously used algorithm as can be demonstrated by an improved π^0 signal-to-background ratio.

Another nice example is the use of ANN for the separation of photon and hadron induced cosmic air showers in the HEGRA experiment [6].

In the H1 collaboration work is in progress to tell electrons from pions using the energy deposition in the electromagnetic (Pb) and hadronic (Fe) sections of the liquid argon calorimeter [7]. At the same time (and with the same network) the deposited and incident energies are determined. With a 41 - 33 - 24 - 4 network the energy reconstruction is achieved without bias and with good resolution. Here it is particularly interesting that the resolution for pions is similar or better than with the standard H1 algorithm which aims at a software compensation of the strongly fluctuating loss in the observed energy in hadronic showers. The preliminary results suggest that one could build 'compensating calorimeters' with ANN processing.

Function approximation: The determination of the incident energy from the energies deposited in calorimeter cells is an example for the use of FF nets as function approximators. The network is trained to yield a continuous output. The goodness of the approximation is controlled by the number of contributing nodes. If a smoothing interpolation is desired the number of nodes has to be appropriately restricted.

Another example for the use of FF networks for function approximation is the determination of the W-boson mass from calorimetric information in a $p\bar{p}$ collider experiment [8]. The determination of a density function by network training is in fact also a function approximation. In these cases the functions are learnt from the sample events of a simulation.

4 Neural Net Hardware

The feed-forward network algorithm is well suited for a hardware implementation. The processes at each node are local and can be executed consecutively and in parallel for all nodes of a layer. Dedicated VLSI chips have been developed which are optimized for fast matrix multiplications as needed for the computation of the activation function (1) of each node.

Table 1 lists some properties of a selection of analog and digital chips which are of interest for applications in Particle Physics. The analog processing is potentially faster and the I/O bandwidth can be higher for a given number of connection pins. On the other hand analog circuits are more susceptible to instabilities. Also for general purpose neuro-computer systems only digital chips offer the required flexibility. Neuro-computers like the CNAPS II system or SYNAPSE-1 (based on the MA16) are optimized for fast neural network training and applications such as image processing.

Table 1: Characteristic properties of a selection of neuro-chips (PU = processor unit).

	ETANN	NeuroClassifier	CNAPS	MA16
arithmetic	6 bit analog	5 bit analog	16 bit fix	16 bit fix
PU per chip	4096	426	64	16
speed [MCPS]	1300	20000	1160	800
clock [MHz]	0.5	20 ns tot. delay	20	25

ANN Applications on the Second Trigger Level

Fermi-Lab activities with the ETANN chip: Neural network hardware has first been used for triggering in a high energy experiment at Fermi-Lab [9] employing the analog chip ETANN e.g. for real-time tracking, calorimetry triggers and recognition of isolated electrons.

When the neural net activity started at Fermilab the ETANN chip was the best choice. Today digital chips can perform similar tasks at the same speed while offering more stability and easier handling. Due to the fast development in electronics this situation could quickly change again.

The Level-2 Trigger of the H1 Experiment: The experiments at the electron-proton storage ring HERA have to cope with a bunch crossing rate of 10 MHz. The H1 experiment has a four-level trigger architecture. At the 2nd level the allocated decision time is 20 μs, too short for a standard programmable processor but offering a niche for a dedicated neuro-processor.

In sect. 3 we have discussed the general concept of this trigger [4]. The hardware which is designed to process feed-forward nets of a maximal size 64 - 64 - 1 is based on the CNAPS chip (table 1) available on a VME board environment and with very good software support. The trigger system will finally consist of about 10 VME boards, each of these 'pattern recognition engines' processing the decision for one specific physics channel. The system is scheduled to be implemented in the H1 experiment by summer 1995.

ANN Applications on the First Trigger Level

The first generation ANN applications for triggering experiments work on the time scale of some 10 μs and, frankly speaking, still on a level of quite moderate complexity. However, given the speed of electronic developments (at least if there is a commercial interest) we would not be surprised if a next generation of neuro-hardware would offer an order-of-magnitude improvement in speed and complexity for the same prize.

Of great interest are ANN applications for high rate, high multiplicity experiments where complex pattern recognition tasks (tracking, calorimetry) have to be performed on a first-level trigger time scale. Replacing the highly specialized hard-wired processors by standardized 'neuro-hardware' could facilitate the design of triggers and potentially also lower the costs. A first taste of such fast applications is given by the following two examples.

A tracking trigger for the CPLEAR experiment: Lacking still a suitable, fast enough neuro-chip standard ECL electronics has been employed for the design of a tracking trigger for the CPLEAR experiment [10]. The trigger is able to find four tracks in a cylindrical arrangement of tracking detectors. The 11 radial layers are divided into 64 azimuthal sectors. One network is assigned to each sector searching for tracks in a region of ± 2 sectors around

the central sector. This leads to 55 inputs for each network and a total of 64 networks with a 55 - 2 - 1 architecture. The decision time is only 75 ns.

The NeuroClassifier: In the CPLEAR experiment the requirement on the tracking trigger is not too demanding since the multiplicities and necessary granularities are quite low. For more complex tasks one certainly has to use integrated electronics. An interesting step in this direction is the development of the NeuroClassifier chip at the University of Twente [11] in close contact with Particle Physics groups at DESY and CERN.

The chip accepts 70 inputs at a bandwidth of 4 GByte/s, has 6 hidden layers and 1 output. The decision time is only 20 ns. The high speed is achieved by analog computing while keeping moderate precision (5 bit). A group in the H1 Collaboration is currently implementing this chip into the first level trigger for a fast analysis of a vertex trigger histogram.

5 Pattern Recognition with Hopfield Nets

The Hopfield Model: The high connectivity of biological neural systems is modelled by the Hopfield net. All neurons of the network interact with each other according to there activation and the connection strength as given by the weights. With two discrete activation states of a neuron, e.g. $S_i = \pm 1$, the system develops a dynamics similar to spin-glasses and can be characterized by an energy function:

$$E = -\frac{1}{2}\left(\sum_{i,j} w_{ij} S_i S_j - 2\sum_i \theta_i S_i\right) \tag{7}$$

In the Hopfield model the weights or interaction strengths w_{ij} are symmetric which guarantees stable minima of the energy function.

Associative memories: Hopfield nets can be used as associative memories. In the training phase the information will be stored into the weights by an algorithm which essentially accounts for the correlation between any two points of the pattern to be memorized. When offering in the memorizing phase a noisy or incomplete pattern the network will (most likely) settle into that stored pattern which is most similar to the presented sample. The fault tolerance of the system depends in a calculable way on the amount of stored information.

The recall proceeds by iteratively updating each neuron S_i according to the sign of the local field at the position i, which is the negative derivative of the energy function (7) with respect to S_i. It can be shown that for symmetric weights this update rule always minimizes the energy function.

The properties of the network suggest its application in experiments for storing and recalling, e.g., patterns in tracking devices. However, the feed-back, recursive architecture and the global minimization algorithm of such networks seems less suitable for hardware implementations than in the case of feed-forward networks.

Solving optimization problems with the Hopfield algorithm: The Hopfield algorithm has been generalized for solving optimization problems, in particular problems with high combinatorial complexity. A well known example is the 'travelling salesman problem' (TSP): How should N cities be connected for a round-trip so that the total path length is minimal. To solve this problem one could identify each link between two cities with a neuron and define a positive interaction between two neurons if they have a city in common. The additional constraints that

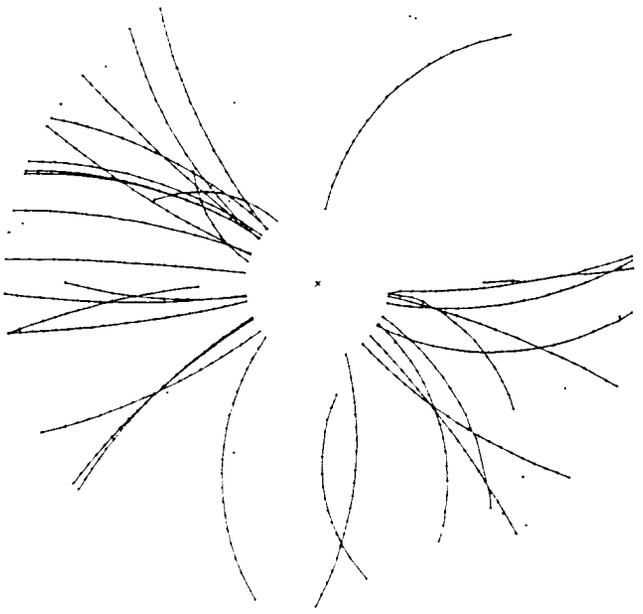

Figure 4: Tracks in the ALEPH TPC reconstructed with a Hopfield net [13].

the path have no bifurcation and that all cities be visited, i.e. that the number of on-neurons be N, can be included in the energy function by adding terms with Lagrange multipliers.

Stochastic extension of the Hopfield model: The energy function has to be minimized as described above by flipping each neuron spin S_i according to the sign of the local field $-\partial E/\partial S_i$. To avoid that the system settles in shallow local minima one can introduce thermal perturbations which cause random spin flips ('heat bath'). For the more complicated treatment of such stochastic systems one can utilize methods known from Statistical Mechanics, such as the 'Mean Field Approximation'.

Application to Tracking

The application of Hopfield networks for track reconstruction has been suggested by Denby and Peterson [12] independently. In their model a neuron is a link between two measured hits in a tracking detector. The weights are some power of the cosine of the angle between two links to ensure smooth curvatures of the found tracks. The energy function contains two additional constraint terms, one to suppress bifurcations of tracks, the other to ensure that all or most hits are used for a track. The importance of the constraint terms can be regulated by the Lagrange multipliers. For instance, the fraction of hits belonging to a found track depends on the noise level in the detector.

The method has been applied to real data of the ALEPH TPC (fig. 4) and found to be comparable in tracking efficiency and computing time to the standard tracking algorithm of ALEPH [13].

A possible disadvantage is that the method does not produce an estimator for the goodness of the track reconstruction. In addition the algorithm seems not to be well suited for a hardware implementation since it is not inherently parallel and local. However, one should carefully investigate if this statement is biased by a too conservative notion of data processing.

Another application of a Hopfield net for tracking is described in [14]. The tracking device is the DELPHI Forward Chamber consisting of 3 double-layers. Each two-hit combination in a double-layer yields up to 4 possible track elements. The network selects the optimal set of track

elements to form a track consistent with the measured hits. Since the network has a better efficiency and requires less computing time than the previously applied method it became the standard method in the DELPHI reconstruction software.

6 Conclusion

The investigation of neural algorithms and their application in soft- and hardware is currently a very active field of research. The application of these methods in Particle Physics has found a wide interest which seems to be steadily increasing as can be inferred from the very large number of neural net contributions to the 'Artifical Intelligence' workshop in Pisa this year [1].

Feed-forward networks have proven to be valuable tools for data analysis (classification of events, particle identification, function approximation, pattern recognition). The advantages of feed-forward nets are: the highly parallel algorithm, the flexibility because of their trainability, the capability to solve high-dimensional problems and the deterministic behaviour. It is important to understand that, despite the word 'neural', decisions of feed-forward nets are clearly deterministic. The intrinsic parallelism of these networks make them well suited for hardware implementations. Specialized analog and digital VLSI chips for neuro-processing are available and have found applications for triggering. One can expect a fast development of the speed and complexity of the neuro-hardware over the next years opening the possibility to be used for first-level triggering in high-luminosity experiments.

The Hopfield algorithm appears very attractive for solving pattern recognition problems. Indeed very promissing studies using real tracking devices, e. g. of LEP experiments, have been carried out and partly included in the standard reconstruction software of the experiments. However, the state of the art in pattern recognition is highly developed which is extremely difficult to beat. In addition, the hardware implementation of Hopfield nets is not so straightforward as in the case of feed-forward nets.

Artificial neural networks are still very far away from the biological model concerning the complexity of problems which can be tackled. Given the activity in this field we can expect much progress in the future. The development of dedicated neuro-hardware is particularly essential for a full exploitation of neural algorithms.

References

[1] A good overview of neural network applications in Particle Physics can be obtained from the proceedings of the 'Workshops on Software Engineering and Artificial Intelligence for High Energy and Nuclear Physics (AIHENP)' which in this year takes place in Pisa (April 3-8, 1995). Previous workshops have been held at Lyon (1990), La Londe les Maures (1992) and Oberammergau (1993).

[2] Proceedings of AIHENP 1993, eds. K.-H. Becks and D. Perret-Gallix, World Scientific (1994).

[3] M. Joswig et al., in [2].

[4] C. Kiesling et al.; P. Ribarics et al.; A. Gruber et al.; T. Krämerkämper et al., in [2].

[5] T. Degener, in [2].

[6] S. Westerhoff and H. Meyer, contribution to the Pisa Workshop [1].

[7] M. Höppner, priv. communication.

[8] L.Lönnblad et al., Phys. Lett. B278 (1992) 181.

[9] For a review see: B. Denby, Proceedings of AIHENP 1993, ed. D. Perret-Gallix, World Scientific (1992).

[10] G. Athanasiu et al., Nucl. Instrum. Methods A324 (1993) 320; F.R. Leimgruber et al., contribution to the Pisa Workshop [1].

[11] P. Masa et al., Proceedings 'Fourth Intern. Conf. on Microelectronics for Neural Networks and Fuzzy Systems', Turin 1994.

[12] B. Denby, Comput. Phys. Commun. 49 (1988) 429; C. Peterson, Nucl. Instrum. Methods A279 (1989) 537.

[13] G. Stimpfl-Abele and L. Garrido, Comput. Phys. Commun. 64 (1991) 46.

[14] R. Frühwirth, Comput. Phys. Commun. 78 (1993) 23 and private communication.