Artificial Neural Networks in High Energy Physics


Perceptrons and Feedback Networks

There were many attempts to construct artificial neural networks in the years following the introduction of the McCulloch-Pitts neuron model, and to study their properties theoretically.

![Diagram of artificial neural networks](image)

Figure 2.3: (a) Feed-forward and (b) feed-back architectures

Many of these attempts organized networks of neurons into layers with asymmetric connection strengths $w_{ij}$ taken to be non-zero only in the forward direction between one layer and the next. These networks modeled computational processes, such as the retrieval of stored information in response to a stimulus, using learning algorithms to set the values of $w_{ij}$. The input to the computation
was represented as the set of $v_i(0)$ of the first layer in the network at time $t = 0$. This input set would then feed-forward through successive layers, and the output of the computation would be represented by the set of $v_i(t)$ of the final layer.

Feedforward perceptron networks a fundamental limitation: it is not possible to find a learning algorithm to solve an arbitrary computational problem with a purely feedforward network. Unlike the more general McCulloch-Pitts network, perceptrons are not Finite state machines. McCulloch-Pitts networks allow bi-directional connections between neurons, and are therefore called feedback networks.

**A Simple One Layer Perceptron**

The simple perceptron has a layer of input neurons $x_j$ and a layer of output neurons $o_i$ with feedforward weights $w_{ij}$

$$o_i = g \left( \theta_i + \sum_j w_{ij} x_j \right),$$

where $g$ is a transfer function and $\theta_i$ are threshold factors.
The perceptron is trained on a set of \( P \) known input patterns \( \mathbf{x}^{(p)} \) and corresponding target output patterns \( \mathbf{o} = \mathbf{t}^{(p)} \) by finding a set of weights that minimizes the summed square error function

\[
E = \frac{1}{2} \sum_p \sum_i \left( o_i^{(p)} - t_i^{(p)} \right)^2.
\]

• Consider a simple 3-neuron perceptron with two input and one output neurons
• The transfer function is taken to be the unit step function \( g(x) = \theta(x) \)
• Choose weights \( w_{00} = w_{01} = 1.0 \)
• The following threshold factors implement logical AND and OR

\[
\theta_0 = \begin{cases} 
-1.5 & \text{implements logical AND} \\
-0.5 & \text{implements logical OR}
\end{cases}
\]

Table 3.1: The AND and OR functions.

\[
\begin{array}{ccc|ccc|ccc}
\hline
x_1 & x_2 & t & x_1 & x_2 & t \\
\hline
1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\hline
\end{array}
\]
**Application to W Boson Mass Reconstruction**


![Graph showing mass reconstruction](image)

*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).*

They use a feedforward neural network with one output node representing the mass $M_W$ of the $W$ boson.

The network has a 480-node input layer corresponding to detectors in the calorimeter. The input layer is feeds forward to a second hidden layer with 100 neurons, which feeds forward to the third hidden layer with 10 neurons, which feeds forward to the output node.
The network is trained using a “Manhattan” back-propagation algorithm using Monte Carlo data generated by the Pythia shower MC code.

The trained network is used to analyze experimental data.
Application to Parton Distribution Function Parametrization


The Neural Network Parton Distribution Functions NNPDF

The distributions are represented by a set of multilayer feed-forward neural networks with weights determined by a back-propagation learning algorithm.

The networks are trained on low $Q^2_0$ experimental data on inelastic lepton-proton scattering, which measures the quark, antiquark and gluon densities inside the proton.
The trained networks are coded in software for operation at any desired energy scale $Q^2$ to provide parton densities for use in high energy experiments and phenomenology.

Figure 3: Flow chart for the construction of the parametrization of structure functions