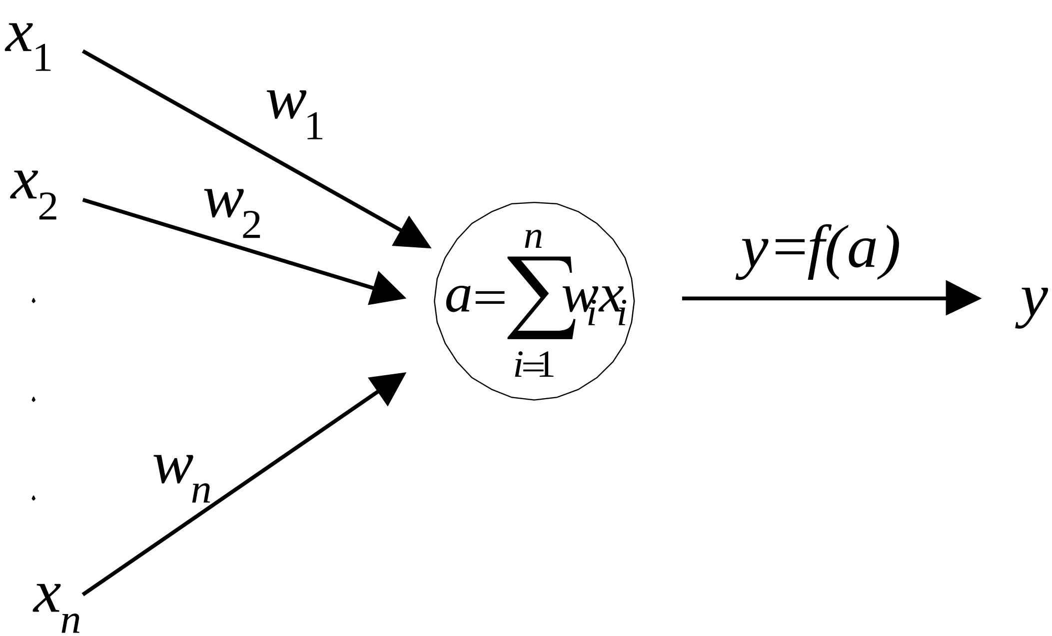
Back-Propagation Neural Network models (BpNN)

Introduction

- Neural Networks represents a class of computational models.
- In the context of analyzing time series special interest should be paid to these nonlinear neural networks models, which can be trained to map past and future values of a time series, thus extracting relationships governing the data.
- Such models could be described in the terms of traditional statistical methods as a “multivariate nonlinear nonparametric inference technique that is data driven and model free”.

The Artificial Neuron



The Artificial Neuron

- 1. Every neuron receives number of input signals $x_1, x_2, \ldots, x_n$. There is weight w_i, $i=1,\ldots,n$ associated with every input channel.
- 2. The neuron computes its generalized input .
- 3. The output value of the neuron is computed and may become input signal to a number of other neurons. The function f, called transfer function may be linear or hard-limit function in the simplest cases, but the sigmoid function is preferred in models aimed at finding non-linear dependencies in the input data.

Back-Propagation Neural Networks

- Back-Propagation Neural Networks (BPNN) term is often used to denote a class of forwardfeed neural networks, which are trained through the backpropagation algorithm for supervised learning. The same class of networks are also known as Multilayer Perceptrons (MLP).
- A subset of all neurons, called input neurons, receive input signals from the external environment. They typically do not process the signals, but transfer them to other neurons, i.e. for all input neurons.

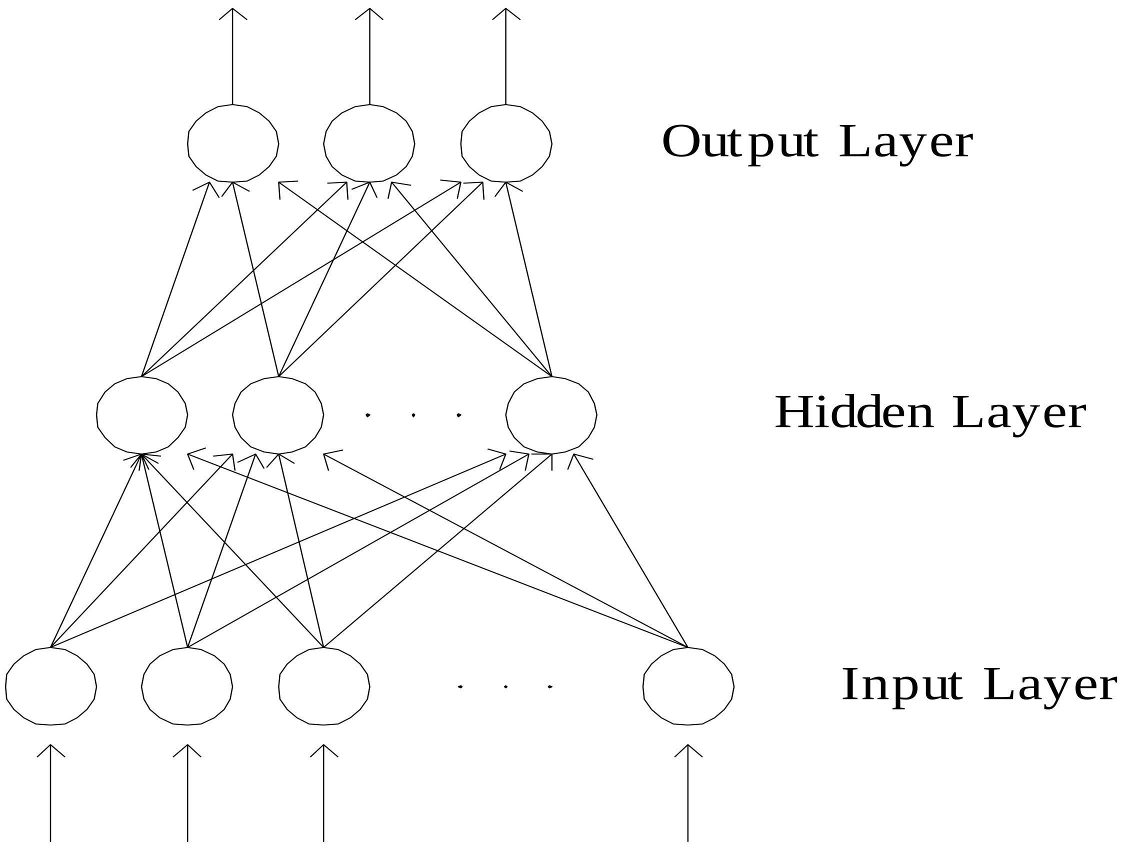
Back-Propagation Neural Networks

- A subset of all neurons, called output neurons, produce output values, which are not used as inputs for other neurons. Their outputs are interpreted by some external mechanism as result of the work of the network. The subset of neurons, which are neither input, nor output, are called hidden neurons.

Forward Processing

- The neurons in these networks are typically organized in layers, where all the neurons in one layer have connections to all the neurons in the next layer.
- The processing goes only in one direction, from the input layer to the output layer, i.e. there are no recursive links.

Forward Processing



Forward Processing

- The hidden neurons could be organized in one or more layers, but the typical BPNN architecture has only one hidden layer.
- It is mathematically proven, that when some general enough restrictions hold, for every neural network of this class, having hidden layers, there exists equivalent neural network having one hidden layer.

Supervised Learning

- The process of training the network is called supervised learning, because it is done on a set of training examples, which include as input vectors so the appropriate output vectors. In other words, during the training there is external factor (sometimes called "teacher"), which tells the network the correct answers.
- The most popular and mathematically well founded learning method is the Back-Propagation Algorithm, also referred as error backpropagation or back error propagation or the generalized delta rule.

The Back-Propagation Algorithm

Let $E(w)$ be a function (known as the energy or cost function) of the weights of the network, which it is desired to minimize

$$E^{total} = \sum_{\mu} \sum_{o} E(w)$$

$$E(w) = \frac{1}{2} [T - O(w)]^2$$

$$\frac{\partial E}{\partial O} = -(T - O)$$

where E^{total} is the total error, summed over μ patterns and o output neurons, *T* is the target and *O* is the observed output, a function of the network weights *w*.

The Back-Propagation Algorithm

Define a transfer function, *f(a)* with gain *g* and no shift, hence *f* has range [0, 1]:

$$a = \sum_{i=1}^{n} w_i x_i$$

$$O = f(a)$$

$$f = \frac{1}{1+e^{-ga}}$$

$$f' = g(1-O)O$$

The Back-Propagation Algorithm

Define network performance measure, for example the root mean square (RMS) error:

$$\varepsilon_{rms} = \sqrt{\frac{\sum_{\mu}\sum_{o}(T-O)^2}{N_o N_e}}$$

where N_o is the number of neurons in the output layer and N_e is the total pattern of patterns.

The Back-Propagation Algorithm

Initialize the weights randomly, drawing from a small distribution.

Present the first pattern to the input nodes of the network.

Propagate the values through the network until the output layer neurons have been reached and the network has a new observable output state.

Compare the observed output values O with the target values T and update the performance measure.

The Back-Propagation Algorithm

Calculate the δ_i value for each output neuron *i*, as follows:

$$\delta_i = -f'(a_i)\frac{\partial E}{\partial O}$$

$$\delta_i = g(1-O)O(T-O)$$

The Back-Propagation Algorithm

Backpropagate the deltas through the network to the preceding layer of neurons *j*, connected to the output neurons *I*, and calculate new delta values:

$$\delta_j = f'(a_j)\sum_i w_{ij}\delta_i$$

This process is continued back through the network until deltas have been calculated for each neuron.

The Back-Propagation Algorithm

The weight of connection from neuron *p* to neuron *q* is updated by:

$$\Delta w_{qp} = \eta \delta_q V_p$$

where V_p is the output of neuron p and η is the learning rate coefficient.

$$w_{qp}^{new} = w_{qp}^{old} + \Delta w_{qp}$$

All the weights are updated according to this rule.

Having completed the train set the error scores are evaluated and compared with the tolerance level.