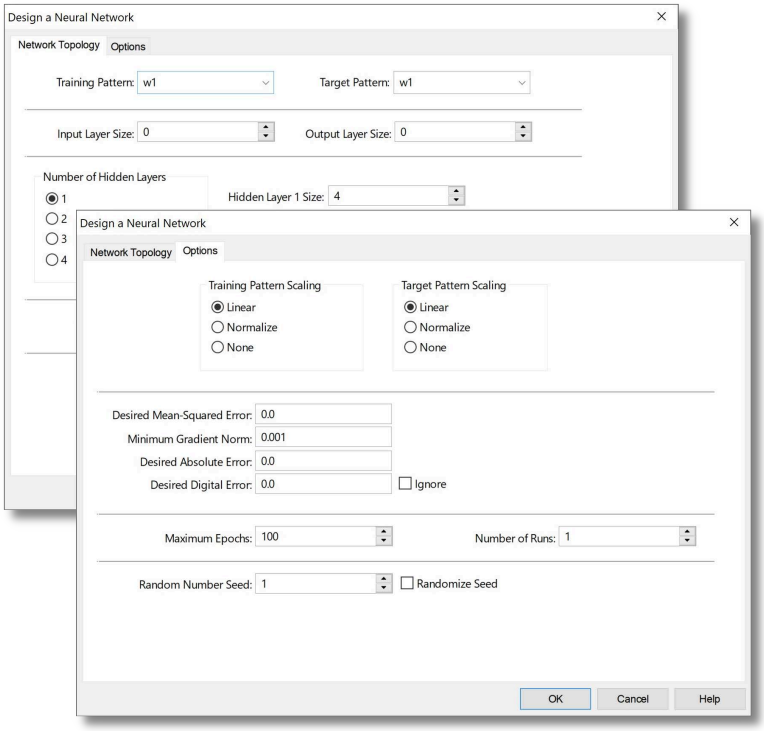
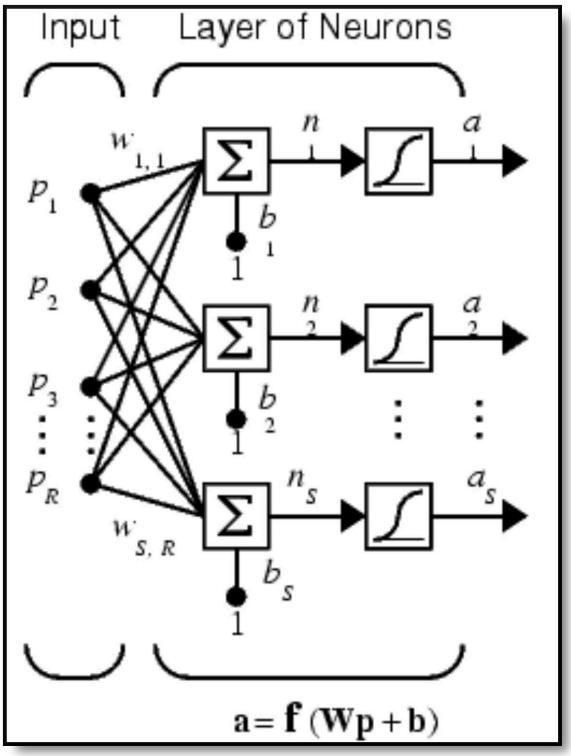
Neural Network Module

Neural networks are able to recognize underlying patterns and predict outcomes based on incomplete or inconsistent information. In contrast to expert systems, which use a static set of rules, neural networks are able to learn and adapt to changes in the environment. With DADiSP/NeuralNet, users can build their own artificial neural networks (ANNs) and apply them to achieve more accurate predictions and pattern classifications.





Neural Network Training

Neural networks resemble the human brain because they can learn. A back-propagation neural network develops its predictive capabilities by being trained on a set of historical inputs and known resulting outputs. The neural net applies random weights to each designated input variable. It then adjusts the weights depending on how closely the actual output values match the desired output values in the training set of historical data. Once the appropriate variable weights have been set that minimize the difference between expected and actual output from the neural net, the neural net can then be applied to new data for classification.

NeuralNet Functions

DADiSP/NeuralNet includes several functions to create, apply and analyze neural networks.

NeuralNet Functions

applynet	Apply a neural network to new input data
innorm	Normalize network inputs to +/- 1.0 using mean and variance
inscale	Linearly scale network inputs to +/- 1.0 using min and max
makecvnet	Create a neural network with cross verification
makenet	Create a neural network
nnoptrun	Extract neural network weights based on best error statistics
nnrun	Extract a particular set of post training data for a run
nnseeds	Extract neural network random seed values
nnweight	Extract neural network weights
outnorm	Normalize network output data using a reference series
outscale	Linearly scale network output data using a reference series