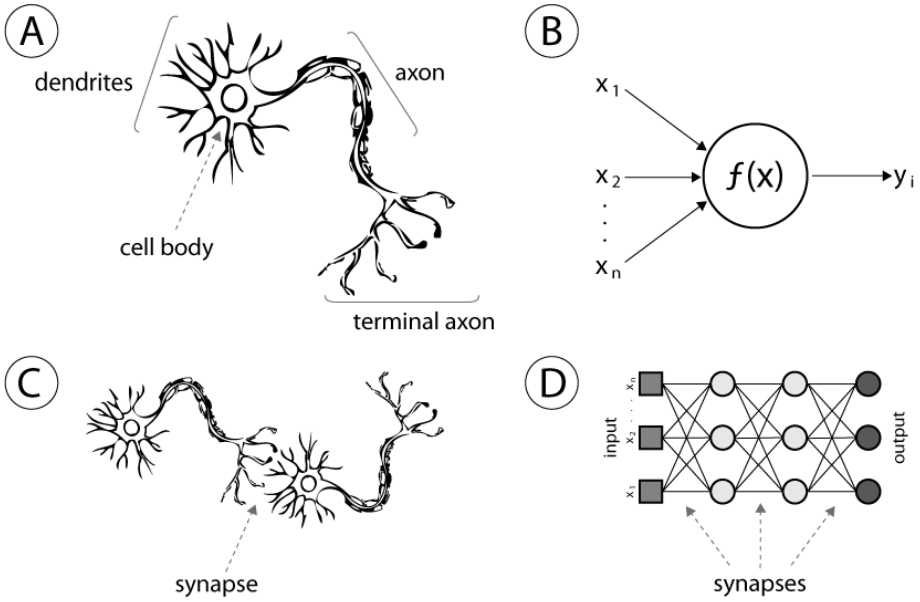
Ciro Donalek (Caltech)

Neural Networks

Sponsored by the Jet Propulsion Laboratory and by the Keck Institute for Space Studies, California Institute of Technology

Outline

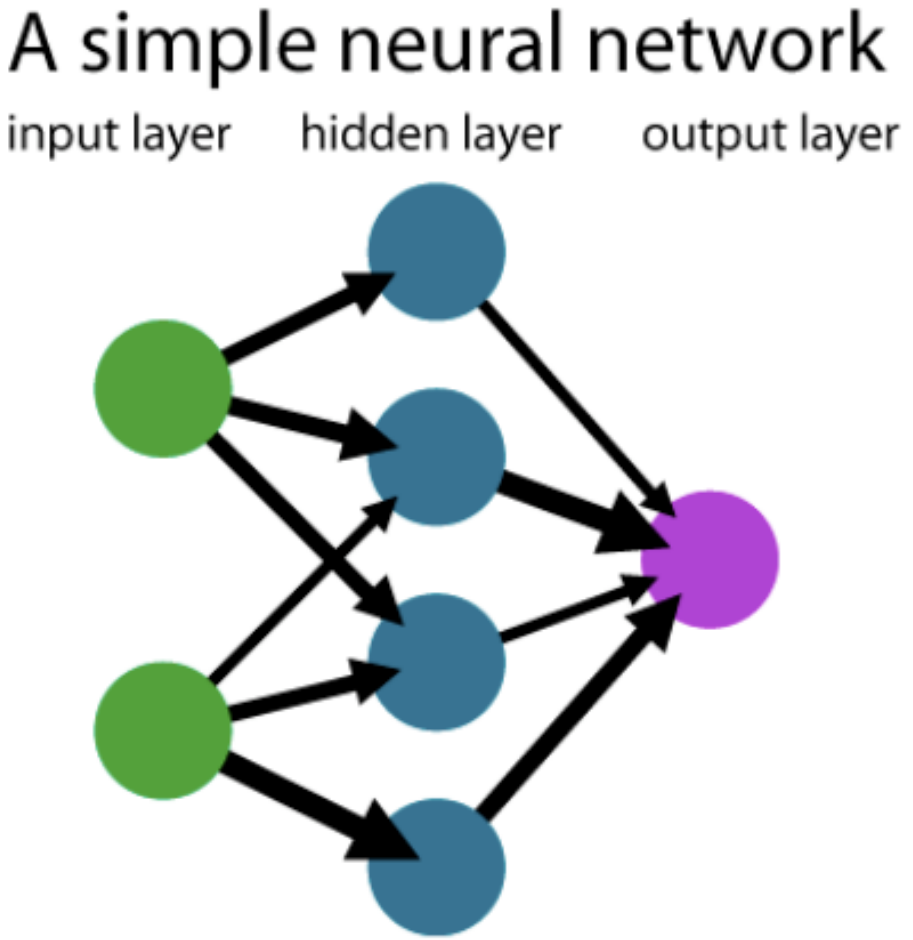
- Artificial Neural Networks
- Multilayer Perceptron (MLP)
- Learning Process
- Hidden Units
- Activation and Error Functions



Artificial Neural Networks

An Artificial Neural Network is an information processing paradigm that is inspired by the way biological nervous systems process information:

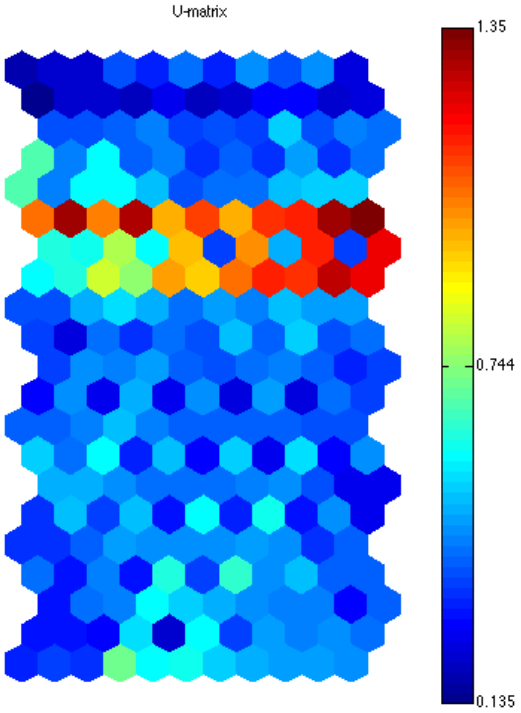
"a large number of highly interconnected simple processing elements (neurons) working together to solve specific problems"

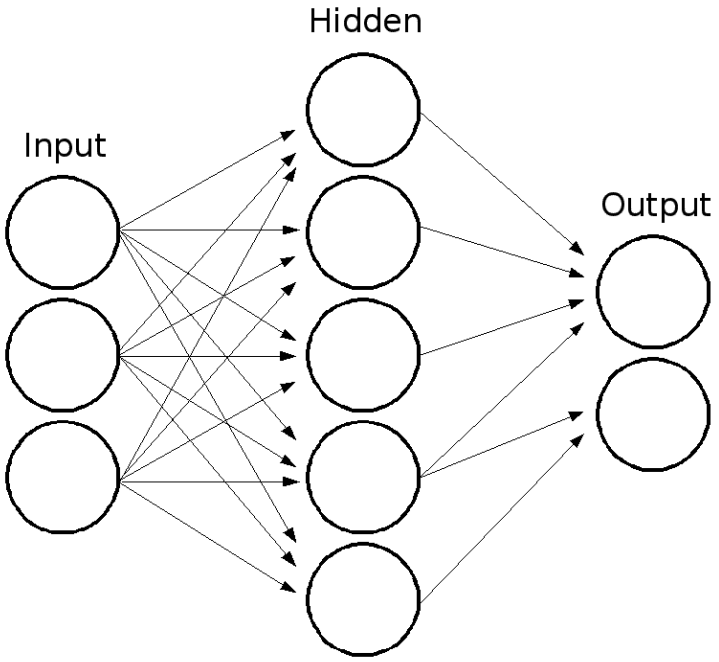


Neural Networks

A Neural Network is usually structured into an input layer of neurons, one or more hidden layers and one output layer.

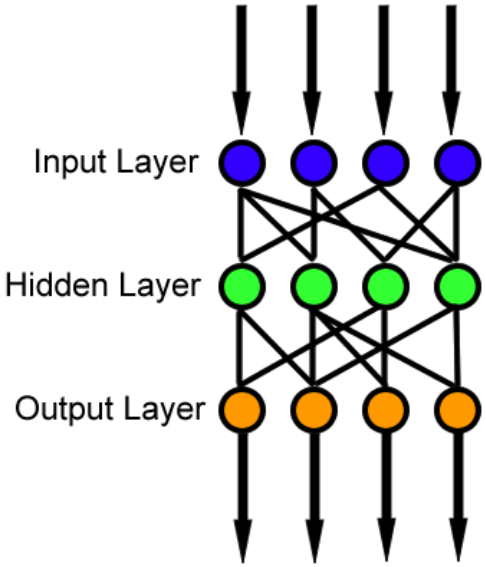
Neurons belonging to adjacent layers are usually fully connected and the various types and architectures are identified both by the different topologies adopted for the connections as well by the choice of the activation function.



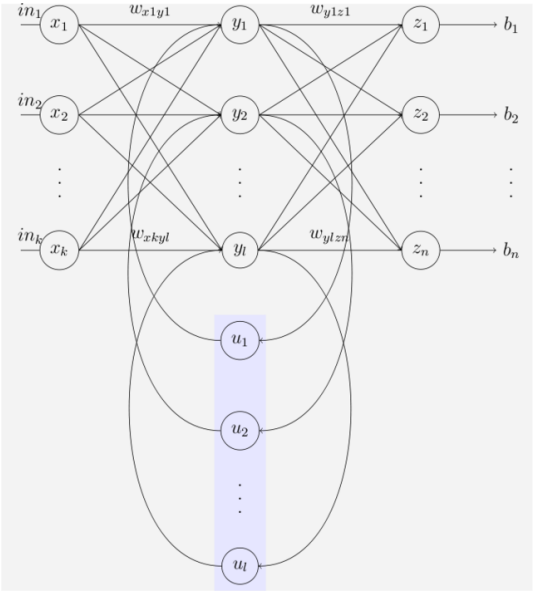


Neural Networks Types

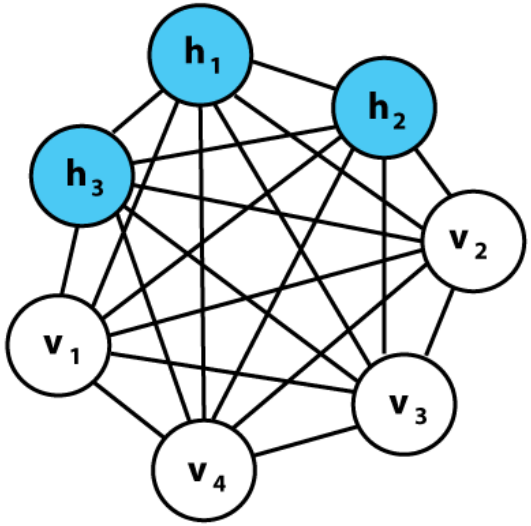
Feedforward: Single Layer Perceptron, MLP, ADALINE (Adaptive Linear Neuron), RBF.
Recurrent: Simple Recurrent Network, Hopfield Network, Elman.
Self-Organized: SOM (Kohonen Maps)
Stochastic: Boltzmann machines, RBM.
Modular: Committee of Machines, ASNN (Associative Neural Networks), Ensembles.
Others: Instantaneously Trained, Spiking (SNN), Dynamic, Cascades, NeuroFuzzy, PPS, GTM.



Feedforward



Recurrent

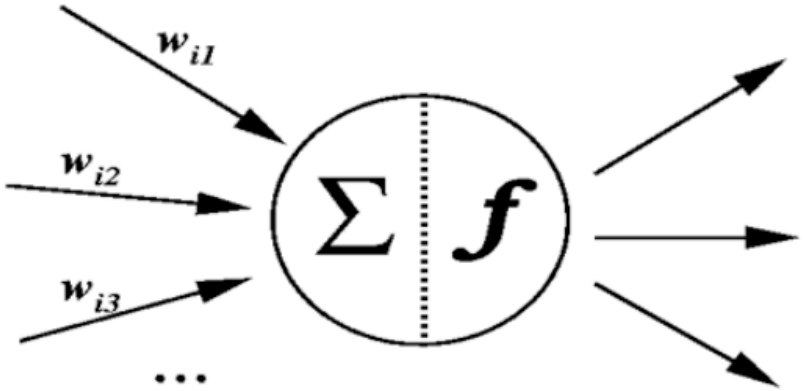


Stochastic – Boltzmann Machine

A simple Artificial Neuron

- The basic computational element is called “node” or “unit”.
- Receives inputs from some other units, or from an external source .
- Each input has an associated weight **w**, which can be modified so as to model synaptic learning.
- The unit computes some function of the weighted sum of its inputs:

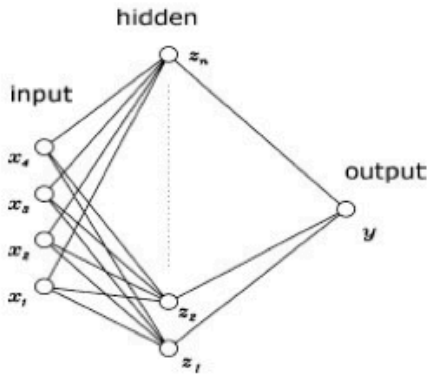
$$y_i = f(\sum_j w_{ij}\, y_j)$$



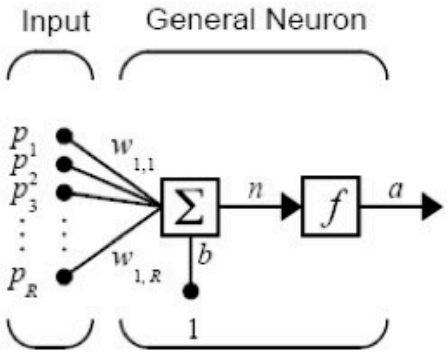
$$y_i = f(net_i)$$

Multi-Layer Perceptron

- The MLP is one of the most used supervised model: it consists of multiple layers of computational units, usually interconnected in a feed-forward way.
- Each neuron in one layer has direct connections to all the neurons of the subsequent layer.



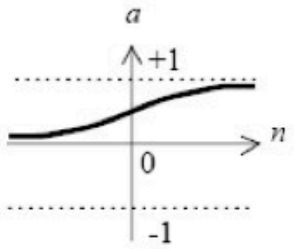
The architecture of a two layer MLP.



$a = f(\mathbf{W}\mathbf{p} + b)$

Where

R = number of elements in input vector

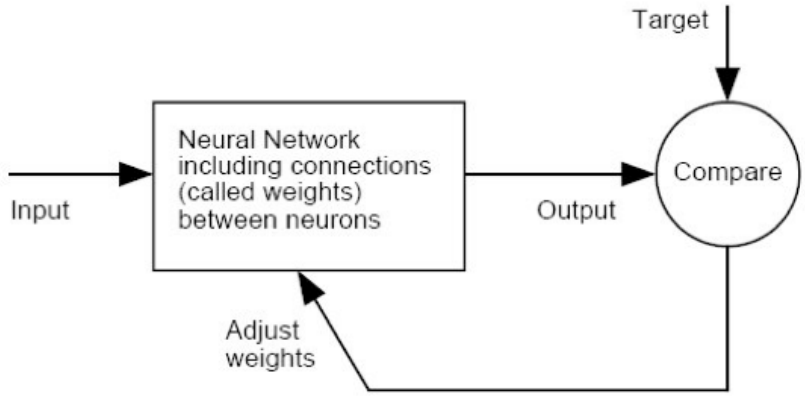


$a = logsig(n)$

Learning Process

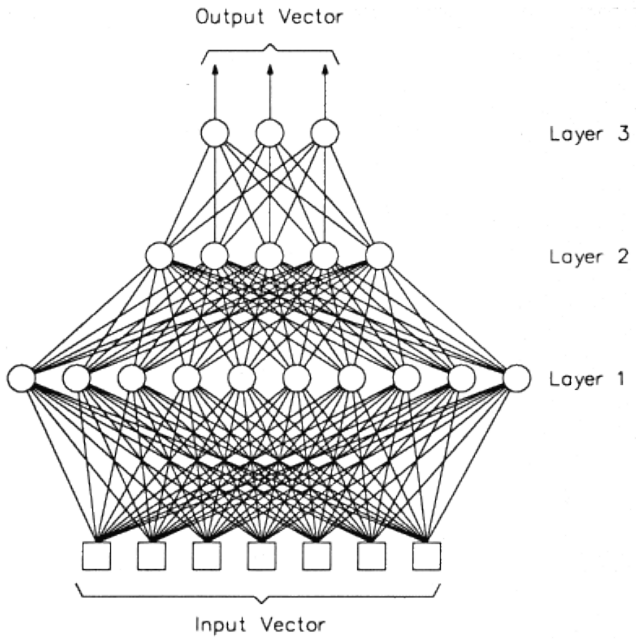
- Back Propagation
  - the output values are compared with the target to compute the value of some predefined error function;
  - the error is then feedback through the network;
  - using this information, the algorithm adjusts the weights of each connection in order to reduce the value of the error function.

After repeating this process for a sufficiently large number of training cycles, the network will usually converge.



Hidden Units

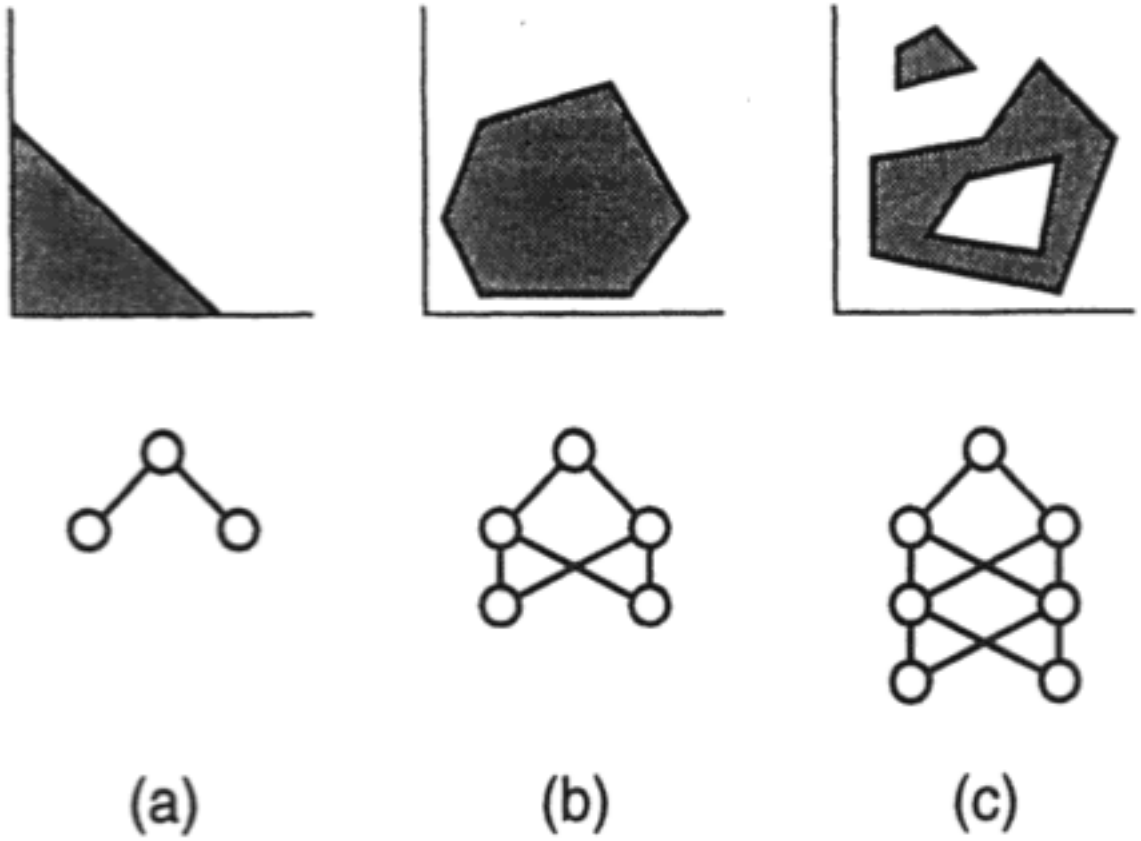
- The best number of hidden units depend on:
  - number of inputs and outputs;
  - number of training cases;
  - the amount of noise in the targets;
  - the complexity of the function to be learned;
  - the activation function, etc.



- Too few hidden units => high training and generalization error, due to underfitting and high statistical bias.
- Too many hidden units => low training error but high generalization error, due to overfitting and high variance.
- Rules of thumb don't usually work.

Image Credits: statistics4u

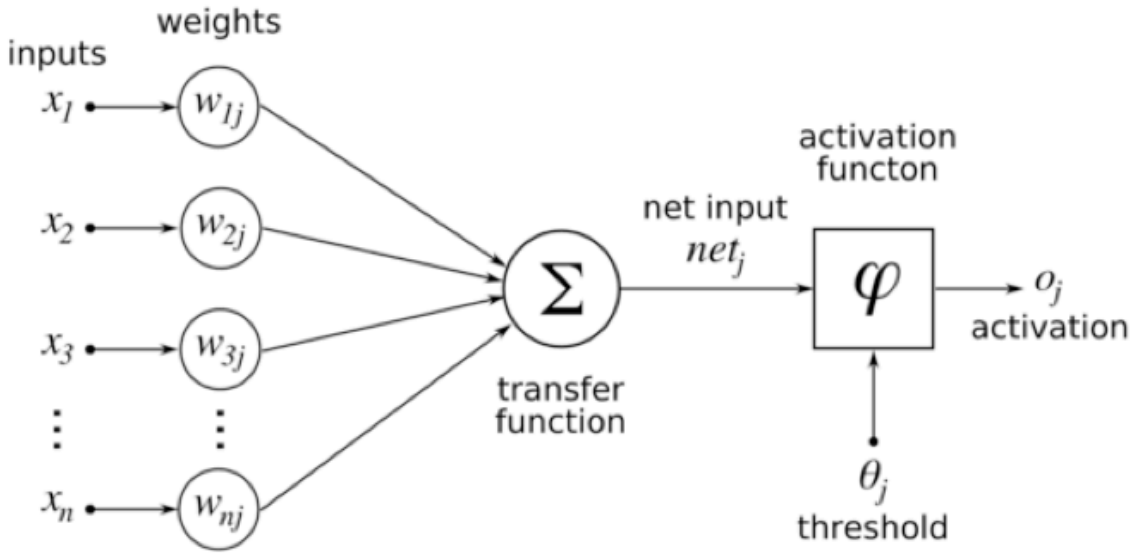
Decision Boundaries



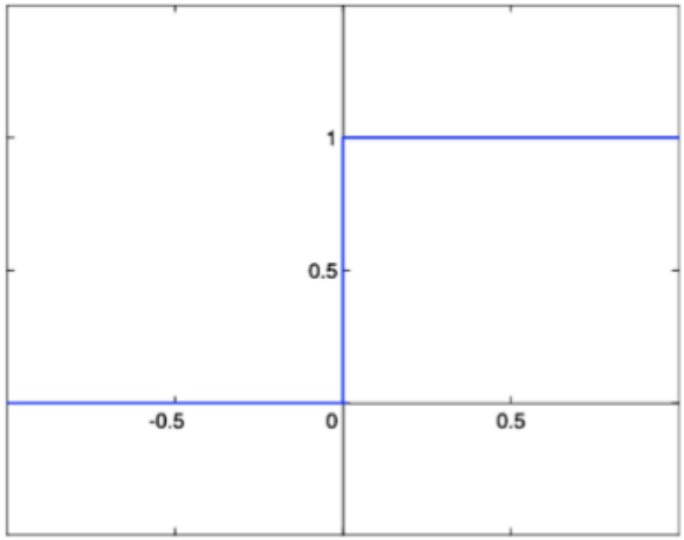
From: Bishop C., Neural Networks for Pattern Recognition

Activation Functions

- Activation Functions
  - used by most units to transform their inputs;
  - needed to introduce non-linearity into the network;
  - linear, logistic, tanh, softmax...



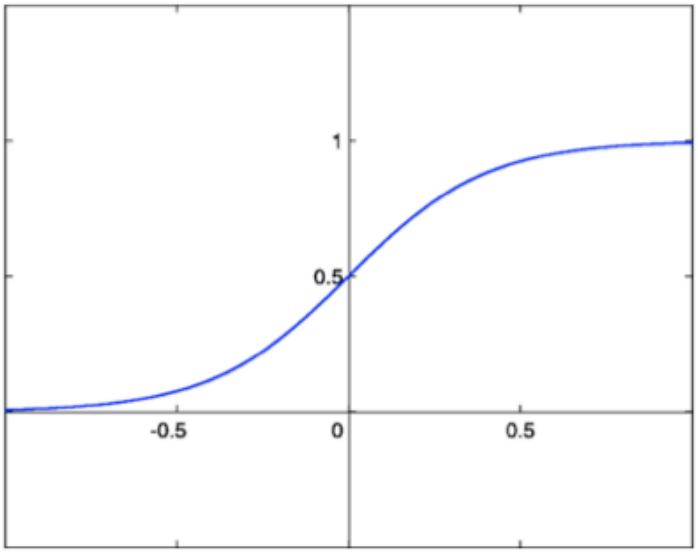
Activation Functions: examples



Step function

The output is a certain value A1, if the input sum is above a certain threshold and A0 if the input sum is below a certain threshold.

When we want to classify an input pattern into one of two groups, we can use a binary classifier with a step activation function.



Sigmoid function

Has the property of being similar to the step function, but with the addition of a region of uncertainty.

Sigmoid functions in this respect are very similar to the input-output relationships of biological neurons.

$$\sigma(t) = \frac{1}{1 + e^{-\beta t}}$$

Two layer network

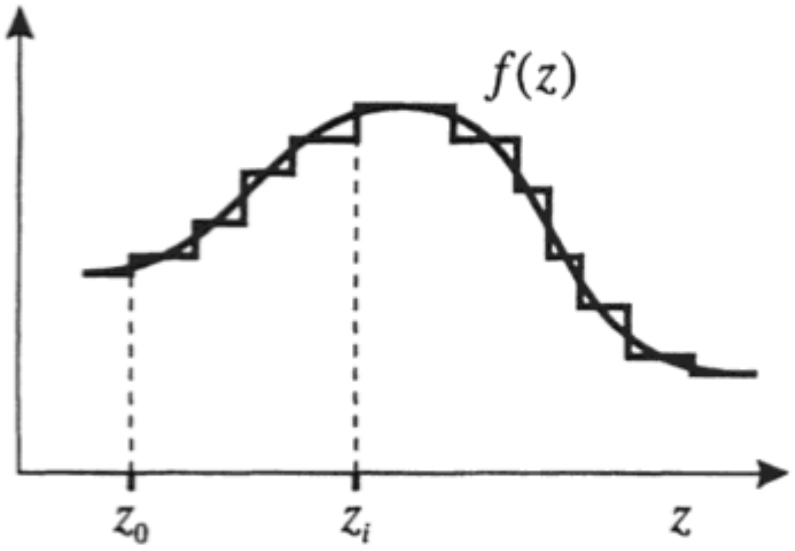


Figure 4.11. Approximation of a continuous function $f(z)$ by a linear superposition of threshold step functions. This forms the basis of a simple proof that a two-layer network having sigmoidal hidden units and linear output units can approximate a continuous function to arbitrary accuracy.

From: Bishop C., Neural Networks for Pattern Recognition

Error Functions

- Error Functions
  - most methods for supervised learning require a measure of the discrepancy between the netwrok output values and the target;
  - sum of the squared errors (SSE), cross entropy (CE), etc.

$$E_p = \frac{1}{2} \sum_j \left(t_j^p - y_j^p \right)^2$$

Activation and Error Functions

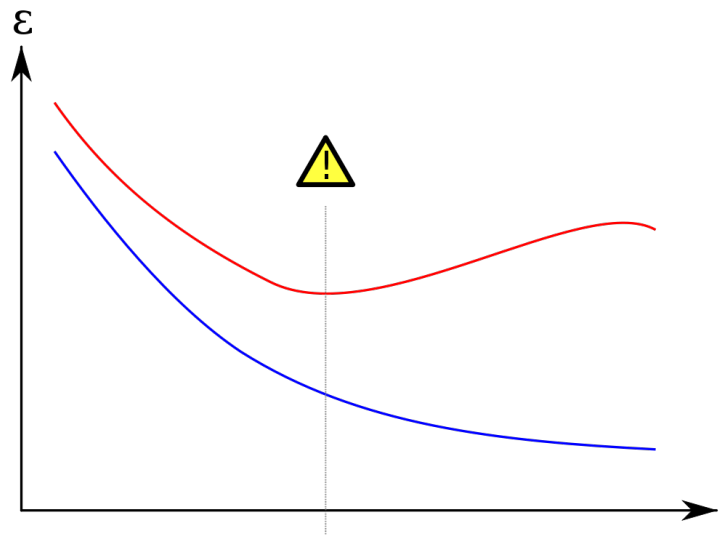
- Error Functions
  - most methods for supervised learning require a measure of the discrepancy between the netwrok output values and the target;
  - sum of the squared errors (SSE), cross entropy (CE), etc.

$$E_p = \frac{1}{2} \sum_j \left(t_j^p - y_j^p \right)^2$$

Using a Multilayer Perceptron with a softmax activation function and cross-entropy error, the network outputs can be interpreted as the conditional probabilities $p(C_1|\mathbf{x})$ and $p(C_2|\mathbf{x})$ where $\mathbf{x}$ is the input vector, C_1 the first class, C_2 the second class.

Early Stopping

- Form of regularization used to avoid overfitting when training a learner with an iterative method.
- Early stopping uses two different data sets: the training set, to update the weights and biases, and the validation set, to stop training when the network begins to overfit the data.
- Typically, the validation error initially decrease, but later on, when the model begins to overfit, the validation error starts rising.
- Idea: stop the learning process as soon as the minimum on validation error is reached.



Blue: training error
Red: validation error

Summary

- Multilayer Perceptron for classification and pattern recognition.
  - Training, Validation and Test Set
  - Choose topology
    - e.g., number of hidden layers, number of hidden nodes
  - Choose learning algorithm
  - Choose error and activation function
  - Train with early stopping
  - Use performance measure to assess the results
  - Deploy