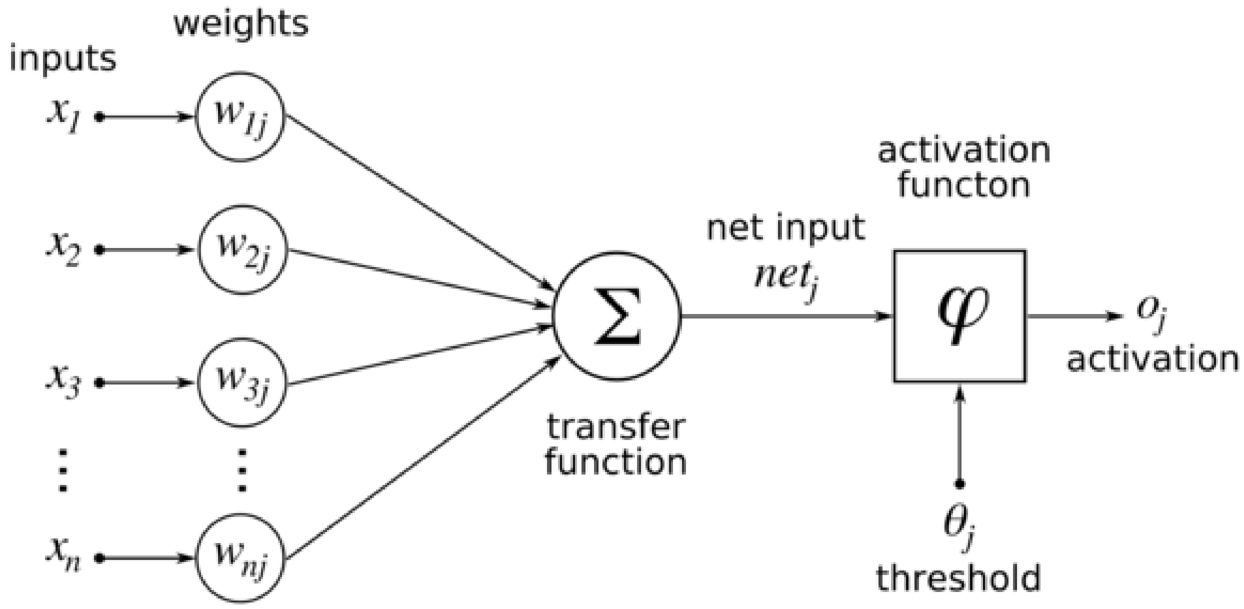
Artificial Neural Networks/Activation Functions

Contents

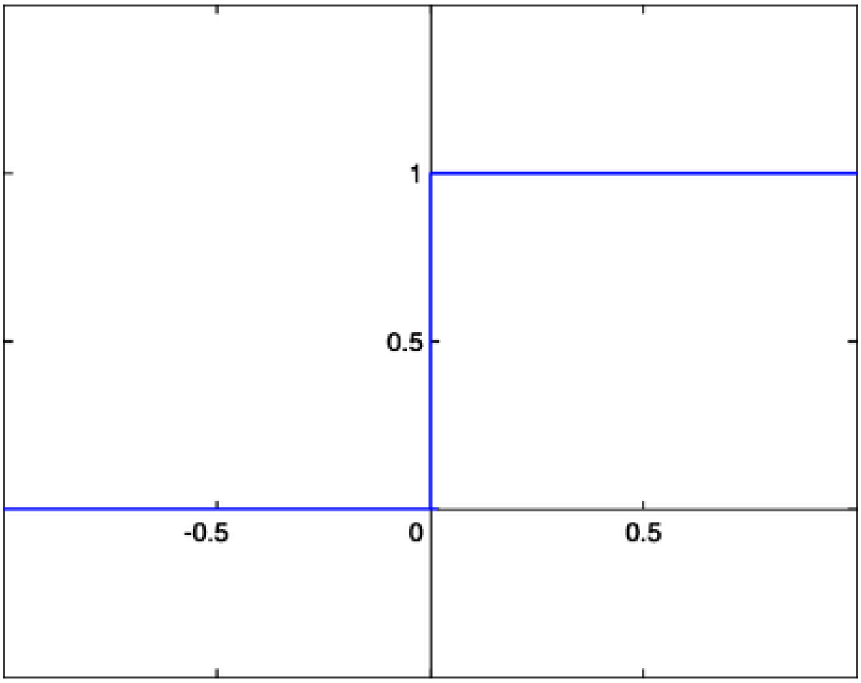
Activation Functions

There are a number of common activation functions in use with neural networks. This is not an exhaustive list.



Step Function

A step function is a function like that used by the original Perceptron. The output is a certain value, A_1, if the input sum is above a certain threshold and A_0 if the input sum is below a certain threshold. The values used by the Perceptron were $A_1 = 1$ and $A_0 = 0$.



These kinds of step activation functions are useful for binary classification schemes. In other words, when we want to classify an input pattern into one of two groups, we can use a binary classifier with a step activation function. Another use for this would be to create a set of small **feature identifiers**. Each identifier would be a small network that would output a 1 if a particular input feature is present, and a 0 otherwise. Combining multiple feature detectors into a single network would allow a very complicated clustering or classification problem to be solved.

Linear combination

A linear combination is where the weighted sum input of the neuron plus a linearly dependant bias becomes the system output. Specifically:

$$y = \zeta + b$$

In these cases, the sign of the output is considered to be equivalent to the 1 or 0 of the step function systems, which enables the two methods be to equivalent if

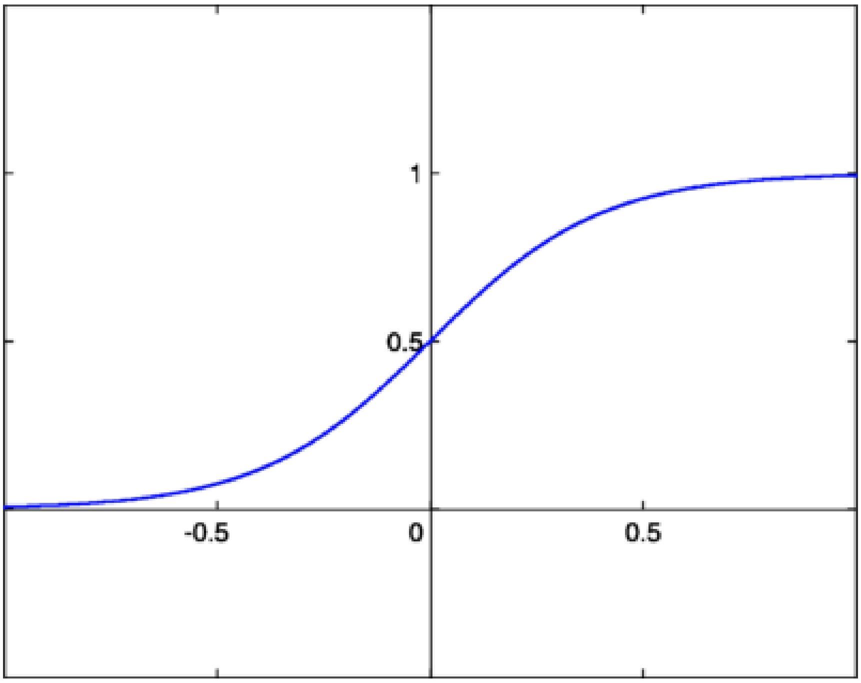
$$\theta = -b.$$

Continuous Log-Sigmoid Function

A log-sigmoid function, also known as a logistic function, is given by the relationship:

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

Where β is a slope parameter. This is called the log-sigmoid because a sigmoid can also be constructed using the hyperbolic tangent function instead of this relation, in which case it would be called a tan-sigmoid. Here, we will refer to the log-sigmoid as simply “sigmoid”. The sigmoid 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, although not exactly the same. Below is the graph of a sigmoid function.



Sigmoid functions are also prized because their derivatives are easy to calculate, which is helpful for calculating the weight updates in certain training algorithms. The derivative when $\beta = 1$ is given by:

$$\frac{d\sigma(t)}{dt} = \sigma(t)[1 - \sigma(t)]$$

When $\beta \neq 1$, using $\sigma(\beta, t) = \frac{1}{1 + e^{-\beta t}}$, the derivative is given by:

$$\frac{d\sigma(\beta, t)}{dt} = \beta[\sigma(\beta, t)[1 - \sigma(\beta, t)]]$$

Continuous Tan-Sigmoid Function

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

Its derivative is:

$$\frac{d\sigma(t)}{dt} = 1 - tanh^2(t) = sech^2(t) = 1 - \frac{(e^t - e^{-t})^2}{(e^t + e^{-t})^2}$$

Softmax Function

The softmax activation function is useful predominantly in the output layer of a clustering system. Softmax functions convert a raw value into a posterior probability. This provides a measure of certainty. The softmax activation function is given as:

$$y_i = \frac{e^{\zeta_i}}{\sum_{j \in L} e^{\zeta_j}}$$

L is the set of neurons in the output layer.

Retrieved from "https://en.wikibooks.org/w/index.php?title=Artificial_Neural_Networks/Activation_Functions&oldid=2778281"