Neural Networks

Authors: Dylan Nikol, Joseph Niehaus, Alexandre Nicolaï. (PDF)

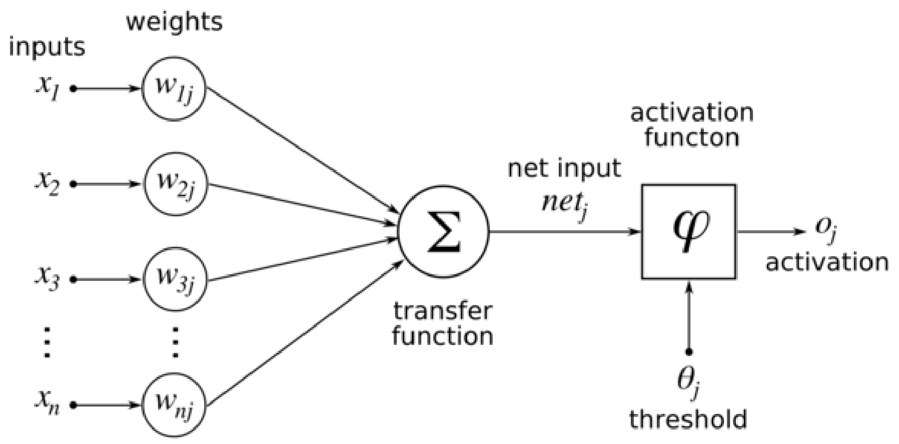


Figure 1: APM_Supp1

Depiction of a multi-layer perceptron

After identifying its advantages of using stochastic gradient descent over gradient descent, particularly when optimizing larger training sets, we shifted our discussion to multi-layered perceptrons (MLP). This is where the analogy to biology proved quite useful; a single neuron on its own is quite futile, but a network of neurons (a network of hundreds of billions of neurons in the case of the human brain), can be extraordinarily powerful. In the context of machine learning, a single-layer perceptron with just 2 layers of nodes (inputs and outputs), is only capable of learning linearly separable patterns, but when these layers are stacked, they can compute any function.

Depiction of a multi-layer perceptron

Activation Functions

An activation function is used to determine the output of the neural network; for example, a YES or a NO (a binary outcome). In this instance, it would be

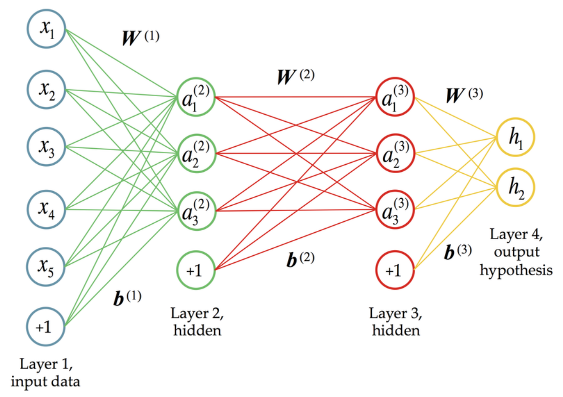


Figure 2: APM_Supp

acceptable to use a step function - one that classifies your output as a 0 or a 1, based on your selected threshold. It's value would be 1 (activated) when the value > threshold, and 0 (inactive) otherwise. However, this presents a problem when multiple neurons are activated, or when we move past evaluating binary options.

Linear Functions

A linear function allows us to evaluate outputs in a way that is proportional to inputs. In this situation, if multiple neurons are activated, we simply take the max (or softmax) as our output. A network consisting of only linear activation functions is very easy to train, but cannot learn complex mapping functions.

$$f(x) = x$$

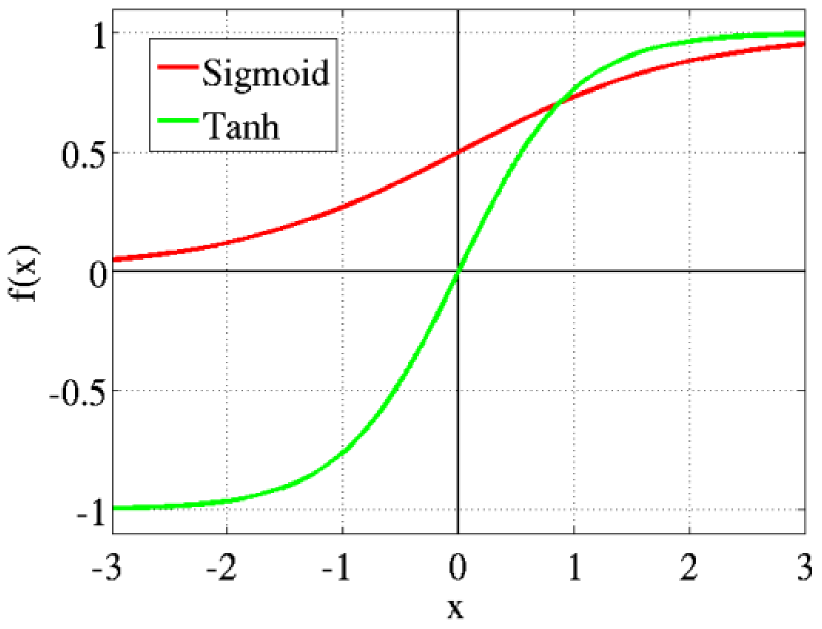
Non-Linear Functions

Sigmoid Sigmoid is a popular function as it allows us to limit our output between 0 and 1. This is especially useful when using our model to predict probabilities. One issue with sigmoid is that towards the outer bounds of the function, gradients can tend to be very small, leading to very high computational costs.

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

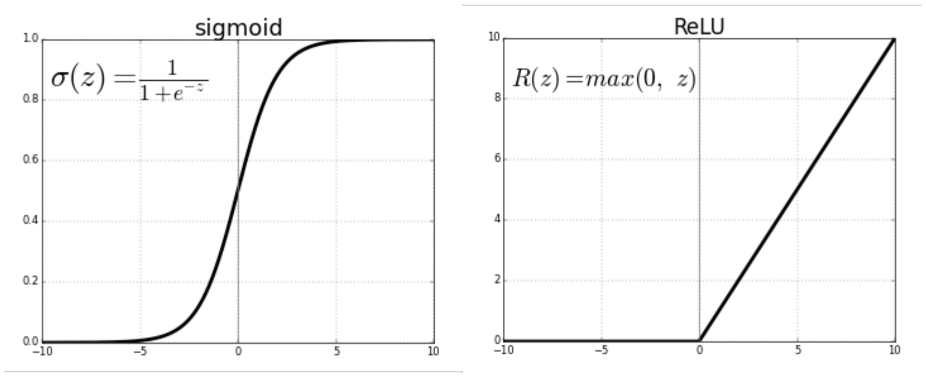
Tanh (Hyperbolic Tangent) Tanh is very similar to the sigmoid function, with a main difference being that its range is between -1 and 1, instead of 0 and 1. As the gradient for tanh is steeper than sigmoid, it can be helpful when running into shallow gradients, although vanishing gradients are still an issue.

$$tanh(x) = \frac{2}{1 + e^{-2x}} - 1$$



ReLU (Rectified Linear Unit) ReLU is a piecewise linear function that will output x if x is positive, and 0 if x is negative. ReLU has become popular because it overcomes the vanishing gradient problem, allowing models to learn faster and perform better. ReLU is also notably simple to implement, requiring only a max() function.

$$f(x) = \max(0, x)$$



Here are some resources we found useful:

How Does Back-Propagation in Artificial Neural Networks Work?

3Blue1Brown

Activation Functions in Neural Networks