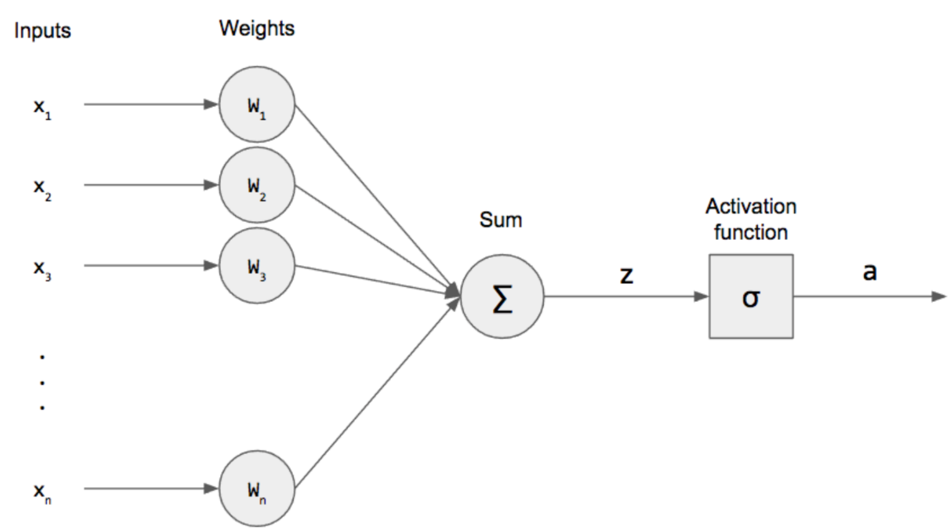
Perceptrons
AI/ML Guild Lab #2
January 18, 2021

1 Introduction

The *Perceptron* is a linear discrimant model which has an important place in Machine Learning as being the first "neural network"-style algorithm.

Neural Networks, which are pattern recognition systems somewhat based off of biological neurons, are incredibly popular today. Perceptrongs were created by Frank Rosenblatt, who was a phsychologist attempting to create a mathematical model of biological neurons (special cells found in brains.)

2 The *perceptron* model



There are n binary inputs given as a vector and exactly the same number of weights $W_1, \ldots, W_n$. These are multiplied together and summed. We denote this as z and call it the *pre-activation* stage of the perceptron.

$$z = \sum_{i=1}^{n} W_i x_i = W^T x$$

We can rewrite this as the inner product of W and x. The *inner product* is a way to multiply vectors (element-wise) with result of this multiplication being a scalar. For further discussion on why these are equivalent see this.

There is another term called *bias* that is a constant. We can incorporate it into the weight vector as element $x_0 = 1$ for all our inputs. The pre-activation stage then becomes

$$z = \sum_{i=0}^{n} W_i x_i$$

This bias term will make more sense when writing code.

Next, comes the non-linear *activation function*, σ.

$$\sigma(a) = \begin{cases} 1, & a \geq 0 \\ 0, & a < 0 \end{cases}$$

Uniting the pre-activation stage and the activation step gives us the mathematical model Rosenblatt created for modelling a single biologic neuron:

$$y(x) = \sigma(z) = \sigma(W^T x)$$

☺

3 Perceptron Power

The output of a perceptron is binary, so it can be used for *binary classification* e.g. that input belongs in one of two classes. But what constraints are there on the types of classes that Perceptrons can discriminant?

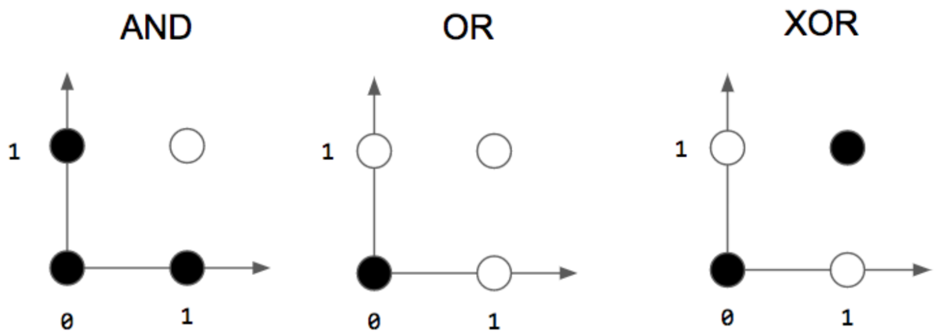
Look back at z with the bias term.

$$z = \sum_{i=0}^{N} W_i x_i + b$$

What does that look like?

3.1 Perceptron Limitations

Perceptrons can only discriminant classes that are *linearly separable*. This means there is a line (or plane) that can linearly divide up the two classes. See examples here.



However, the *perceptron convergance theorem* states that if there exists a linearly separatable solution then the perceptron learning algorithm is guaranteed to find an exact solution in a finite number of steps.