1 Perceptron Theory

Perceptron is also known as a single neuron. It is used as a binary classifier, and is a type of supervised learning. After training on data, it is able to classify a given input to one of two classes.

It was introduced in 1943 by Warren McCulloch and Walter Pitts. They released a paper in 1958 with all the details of the perceptron https://psycnet.apa.org/doiLanding?doi=10.1037%2Fh0042519.

1.1 Requirements

There is limitation for using the perceptron:

- Binary classification only - i.e only two classes in the dataset.
- Training data must be labeled.
- Data has to be linearly separable.

1.2 Definition

The perceptron can be defined as a function $f(\vec{x})$, that take a feature vector $\vec{x}$:

$$
\begin{aligned}
f(\vec{x}) &= h(\vec{w} \cdot \vec{x} + b) && (1) \\
&= h(w_1 \cdot x_1 + w_2 \cdot x_2 + b) && (2)
\end{aligned}
$$

Where $\vec{w}$ is the weight vector with the two weights for the perceptron and b is the bias of the network.

Note that we use a activation function called *Heaviside step function*. The output of the activation function is either 0 or 1.

1.3 Why do we need a bias?

The bias is important to improve the flexibility of the model. Without a bias, the model will always go through origin. When we introduce a bias, it allows the model to pass thought the x-axis at different points.

1.4 Training

For the perceptron model: $f(x) = b + x_1 w_1 + x_2 w_2$, where b is the bias term.

1. Initialize the weights w_i and the bias b (usually to small random values or zeros).

2. Loop over each training instance until some stopping criteria are met (e.g., all examples are classified correctly or maximum iterations are reached).

3. For each instance, calculate the output:

$$y = \sigma(b + x_1 w_1 + x_2 w_2), y \in [0, 1]$$

4. Compare the target value, t, to the predicted output y.

5. If $t = y$, continue to the next instance. If not, update the weights and bias:

   (a) For each weight:
   $$w_i = w_i + \eta(t - y)x_i$$

   (b) Update the bias term:
   $$b = b + \eta(t - y)$$

   where η is the learning rate.

1.5 Perceptron Convergence Theorem

If the dataset is linearly separable, then the perceptron will eventually find a solution for the binary classification. Unless the training rate η is to high. It is important to note that there could be more than one solution.