Introduction to Perceptrons

A Foundational Concept in Neural Networks (1.3)

Mayank Pratap Singh

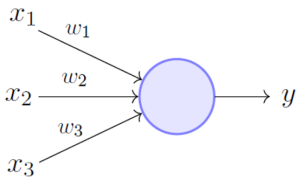
1 Perceptron: A Fundamental Building Block

A perceptron is a fundamental building block of artificial neural networks and a machine learning algorithm used for supervised learning of binary classifiers. It was first introduced by Frank Rosenblatt in 1957.

1.1 Basic Structure

- Inputs: $x_1, x_2, \ldots, x_n$
- Weights: $w_1, w_2, \ldots, w_n$
- Bias: b
- Linear combination: $z = w_1x_1 + w_2x_2 + \ldots + w_nx_n + b$
- Activation function: Applies a function to z to produce the output

$$z = \sum_{i=1}^{n} w_i x_i + b \tag{1}$$



Perceptron Model (Minsky-Papert in 1969)

Figure 1: Perceptron

1.2 Activation Function

An activation function introduces non-linearity into the model, allowing the network to learn complex patterns. One common activation function is the step function, defined as:

$$f(z) = \begin{cases} 1 & \text{if } z \geq \theta \\ 0 & \text{if } z < \theta \end{cases} \tag{2}$$

where θ is a specified threshold value.

1.3 Example Problem

Consider a dataset with student IQ, CGPA, and a binary label indicating whether they are placed:

IQ	CGPA	Placed
78	7.8	1
69	5.1	0

- Inputs: $x_1 = \text{IQ}$, $x_2 = \text{CGPA}$
- Weights: w_1, w_2
- Bias: b

Training involves finding the optimal values of w_1, w_2, and b. For example, if $w_1 = 1$, $w_2 = 2$, and $b = 3$, and a new student has IQ = 100 and CGPA = 5.1:

$$z = 1 \cdot 100 + 2 \cdot 5.1 + 3 = 113.2 \tag{3}$$

Since z is positive, the output is 1 (placed).

1.4 Geometric Interpretation

The perceptron essentially creates a linear decision boundary. For example, the equation $z = w_1 x_1 + w_2 x_2 + b$ can be seen as:

$$w_1 x_1 + w_2 x_2 + b = 0 \tag{4}$$

This represents a line in 2D, a plane in 3D, and a hyperplane in higher dimensions.

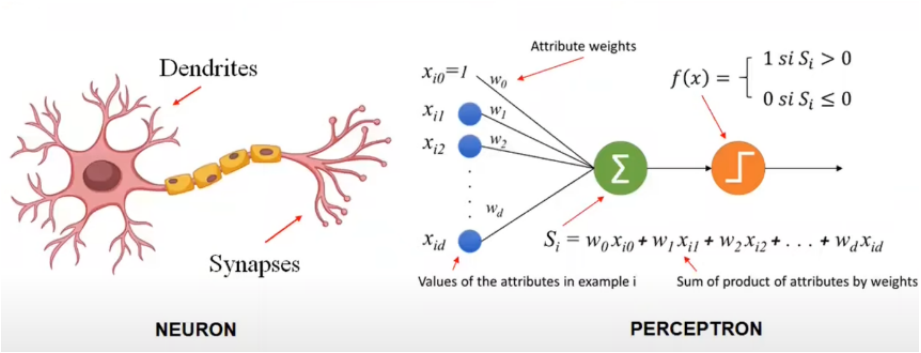


Figure 2: Neuron Vs Perceptron

Feature	Neuron	Perceptron
Definition	Basic unit of artificial neural networks mimicking biological neurons.	A specific type of artificial neuron used for binary classification.
Functionality	Can implement various activation functions, allowing for learning complex patterns.	Computes a weighted sum and uses a step function as the activation function for binary output.
Complexity	Can be part of complex architectures (e.g., deep neural networks).	Typically a single-layer network, limited to binary classification.
Learning Capability	Can learn non-linear relationships with different activation functions.	Can only learn linearly separable functions.
Architecture	Can be part of multi-layer networks (multi-layer perceptrons).	Often considered a single-layer network.
Output	Can produce continuous or binary outputs depending on the activation function used.	Produces binary outputs (0 or 1).

Table 1: Perceptron vs. Biological Neuron

2 Perceptron vs. Biological Neuron

3 Neural Networks and Neuroplasticity

Human neurons are complex and capable of neuroplasticity, meaning they can change over time. In contrast, connections in perceptron models remain fixed once trained. Weights in a perceptron indicate the strength of the connection between inputs and the node, with higher weights implying greater feature importance.

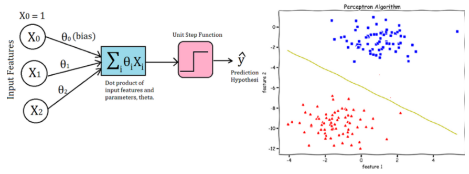


Figure 3: Perceptron

3.1 Geometric Interpretation of the Perceptron

The decision boundary created by the perceptron can be visualized as a line (2D), a plane (3D), or a hyperplane (higher dimensions). For instance:

$$z = w_1x_1 + w_2x_2 + w_3x_3 + b \geq 0 \tag{5}$$

4 Limitations of the Perceptron

The perceptron can only classify linearly separable data. For more complex classifications, multi-layer perceptrons (MLPs) or other neural network architectures are used.

5 Perceptron in Scikit-Learn

Listing 1: Perceptron Example in Scikit-Learn

```
from sklearn.linear_model import Perceptron

# Create the perceptron model
p = Perceptron()

# Fit the model
p.fit(X, y)

# Retrieve the weights and intercept
weights = p.coef_
bias = p.intercept_
```

6 Conclusion

The perceptron is a foundational concept in neural networks and machine learning. Despite its simplicity, it provides valuable insights into the workings of more complex models and algorithms.