Deep Neural Networks

An Engineer's Handbook

The Sarasvati Project

PUBLISHED BY PUBLISHER

BOOK-WEBSITE.COM

First printing, March 2022

Contents

List of Figures

List of Tables

1. Section heading

1.1 Perceptron

A perceptron is a fundamental building block in the world of deep learning! It's a simple yet powerful model that acts as a single artificial neuron within a neural network. Here's how it works:

Imagine a perceptron as a gatekeeper. It takes in a bunch of information (represented by numbers) as its input, like the features of an image or the words in a sentence. Inside, it performs a calculation on this information using weights and a bias. Think of the weights as adjustable knobs that control how much each piece of information affects the decision. The bias is like a threshold that the combined weighted information needs to cross to pass through the gate.

The key output of a perceptron is a binary classification: either 0 or 1. Based on its calculation, the perceptron decides whether the input belongs to one class or the other. For example, in image classification, it might decide if an image contains a cat or not.

Here are some key points about perceptrons:

- **They are linear classifiers:** This means they can only draw straight lines to separate data points into different classes. This limits their ability to handle complex data patterns.
- **They are the foundation of neural networks:** Multi-layer perceptrons, which are networks of interconnected perceptrons, can handle more complex data by combining the decisions of multiple layers.
- **They are simple and easy to understand:** This makes them a great starting point for learning about deep learning and how neural networks work.

1.2 Break down of Perceptron

1. **Input Layer:** Imagine a set of numbers being fed into the perceptron, each representing a different piece of information (like pixel values in an image or features of a data point). These are called inputs.
2. **Weights and Bias:** Each input is associated with a weight, which acts like a multiplier determining how much influence that input has on the final decision. Additionally, there's a bias term, which acts like a constant offset influencing the overall output.
3. **Weighted Sum:** Each input is multiplied by its corresponding weight, and all these products are summed up along with the bias. This gives us the weighted sum, which represents the combined influence of all the inputs.
4. **Activation Function:** The weighted sum doesn't directly tell us whether the input belongs to one class or another. So, we pass it through an activation function like the sigmoid function.

This function transforms the sum into a value between 0 and 1, which we can interpret as a probability of belonging to a specific class.

5. **Output Layer:** Finally, the output of the activation function determines the perceptron's output. If the value is above a certain threshold (usually 0.5), the output is considered 1 (indicating "yes" for the given class). Otherwise, the output is 0 (indicating "no").

Here's an analogy to help visualize: Think of the perceptron as a room with a light switch. Each input is like a switch knob, and the weights determine how much turning each knob contributes to the brightness of the room. The bias is like a starting brightness level, and the activation function acts like a dimmer switch that adjusts the final brightness based on the combined influence of all the knobs. Ultimately, if the room is bright enough, we declare it "daytime," and if it's dark, we call it "nighttime."