Neural Network Models

and their usage in Deep Learning

Harsh Sanjay Roniyar, 22B3942

Mentor:
Krisha Shah

Summer of Science, 2023

Table of Contents

Table of Contents

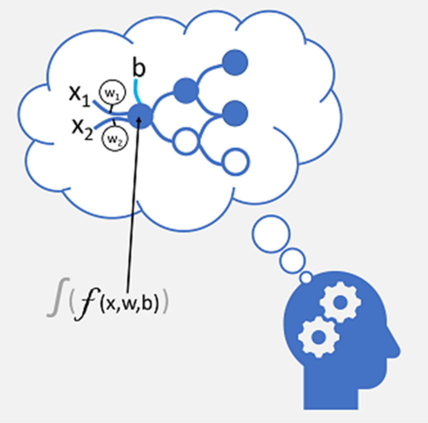
Deep Learning

Deep learning is an advanced form of machine learning that tries to emulate the way the human brain learns. In your brain, you have nerve cells called neurons, which are connected to one another by nerve extensions that pass electrochemical signals through the network.

When the first neuron in the network is stimulated, the input signal is processed, and if it exceeds a particular threshold, the neuron is activated and passes the signal on to the neurons to which it is connected. These neurons in turn may be activated and pass the signal on through the rest of the network.

Deep learning emulates this biological process using artificial neural networks that process numeric inputs rather than electrochemical stimuli.

Table of Contents



The incoming nerve connections are replaced by numeric inputs that are typically identified as x. When there's more than one input value, x is considered a vector with elements named x_1, x_2, and so on.

Associated with each x value is a weight (w), which is used to strengthen or weaken the effect of the x value to simulate learning. Additionally, a bias (b) input is added to enable fine-grained control over the network. During the training process, the w and b values will be adjusted to tune the network so that it "learns" to produce correct outputs.

The neuron itself encapsulates a function that calculates a weighted sum of x, w, and b. This function is in turn enclosed in an activation function that constrains the result (often to a value between 0 and 1) to determine whether or not the neuron passes an output onto the next layer of neurons in the network.

Table of Contents

Training a DNN

The training process for a deep neural network consists of multiple iterations, called epochs. For the first epoch, you start by assigning random initialization values for the weight (w) and bias b values.

Then the process is as follows:

1. Features for data observations with known label values are submitted to the input layer. Generally, these observations are grouped into batches (often referred to as mini-batches).
2. The neurons then apply their function, and if activated, pass the result onto the next layer until the output layer produces a prediction.
3. The prediction is compared to the actual known value, and the amount of variance between the predicted and true values (which we call the loss) is calculated.
4. Based on the results, revised values for the weights and bias values are calculated to reduce the loss, and these adjustments are backpropagated to the neurons in the network layers.
5. The next epoch repeats the batch training forward pass with the revised weight and bias values, hopefully improving the accuracy of the model (by reducing the loss).

Table of Contents

Loss vs Cost

Loss function computes the error for a single training example (discrepancy between $\hat{y}_i$ and y_i).

Cost function is the average of the loss function of the entire training set.

Loss Function

$$L\left(\hat{y}_i, y_i\right)$$

Cost Function

$$J(w, b) = \frac{1}{m} \sum_{i=1}^{m} L\left(\hat{y}_i, y_i\right)$$

As an example, let us analyse and understand the cost function for logistic regression.

Cost Function for Logistic Regression

Cost Function for Logistic Regression

$$J(w, b) = -\frac{1}{m}\sum_{i=1}^{m}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

Relation between output and input features

$$\hat{y}_i = \frac{1}{1 + e^{(-(Wx+b))}}$$

By choosing the cost function as defined above, we have ensured that it will have at most one minimum (strictly convex function).

Thank You !