Training a Neural Network

Nandinii Yeleswarapu

September 7, 2025

Introduction

In my last post titled "What Is A Neural Network", I talked about the basic building blocks of a neural network and how we can think of neural networks as a series of chained functions. But knowing *what* a network is isn't enough; the real magic lies in *how we train it*. This post walks through the core steps needed to train a neural network until its predictions are accurate enough for the task at hand.

The Training Loop

The training process can be broken down into four main steps:

- **Forward Pass:** Input data flows through the network, layer by layer, until it produces a prediction, often called scores.

- **Compute Loss:** We use a loss function to measure how far off the model's prediction is. High loss = big mistakes. Low loss = predictions are close to correct.

- **Backward Pass (Backpropagation):** Using calculus, the model computes how each layer's weight contributed to the error. By moving backward through the network layer by layer, we calculate how each weight should change. These adjustments make the next forward pass more accurate.

- **Update Parameters:** An optimizer (like SGD or Adam) takes those gradients and actually updates the weights of the network. This is the "learning" step where the model improves.

After updating, the cycle repeats: the new parameters are tested in another forward pass, the loss is computed again, and so on. Training continues until the loss stops improving significantly, which we call **convergence**.

Putting It Together

The training loop can be summarized like this:

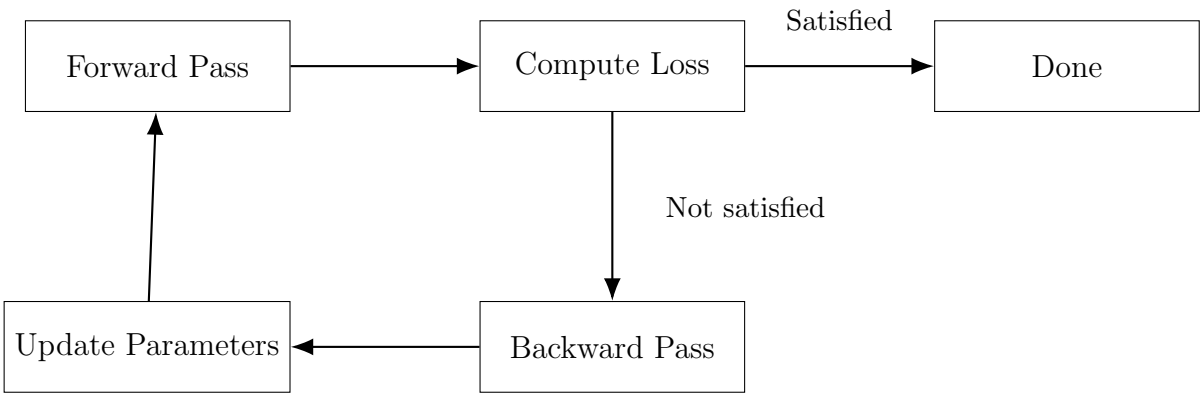

Figure 1: The training loop of a neural network.

In Summary

Training a neural network involves these three simple steps: run the network to get a prediction, check how wrong the prediction is, modify the models weights a little so we're less wrong next time. Repeat this process enough times, and what starts as random weights becomes a model that can recognize patterns, classify data, and power real-world applications.

References and Resources

Much of what I learned while building this foundation came from the excellent course: CS182: Deep Learning at UC Berkeley (Spring 2021).