Building Neural Networks from Scratch

Luca WB

2026-01-08

Table of contents

0.1 Brief summary

The goal of this page is to understand how to implement a neural network from scratch without any external libraries. I'll assume that you already have some basic knowledge of Artificial Neural Networks. The main reason for doing this is to demystify the "black box" of Deep Learning.

For this project, the main resource (though not the only one) is this awesome video from Andrej Karpathy. I will cover how to implement AutoGrad for backpropagation. By the end of this post, you will be capable of creating MLPs from scratch!

0.2 Basic knowledge of derivatives

To make sense of how an NN trains and learns, you first need a solid grasp of what a derivative is. A derivative is essentially an operation that gives us the slope of a function at any given point. For our purposes, we'll keep it simple. Take the function $f(x) = 3x^2 - 4x + 5$, and see the graph below.

"'bjnpllihnn #| code-fold: true #| code-summary: "Show Python code" import numpy as np import matplotlib.pyplot as plt

1 Define the function

```
def f(x): return 3*x**2 - 4*x + 5
```

2 Generate x values

x = np.linspace(-5, 5, 400) y = f(x)

3 Plot

plt.figure(figsize=(6, 4)) plt.plot(x, y, label=r”$f(x) = 3x^2 - 4x + 5$“) plt.axhline(0, color=”black”, linewidth=0.5) plt.axvline(0, color=“black”, linewidth=0.5) plt.xlabel(“x”) plt.ylabel(“f(x)”) plt.title(“Graph of the function”) plt.legend() plt.grid(True) plt.show()