|  |  |
| --- | --- |
| Assignment 2  ITRI 626 | ENRICO DREYER  31210783 |

Table of Contents

[Modifying the code 2](#_Toc80776176)

[Is there a better way? 2](#_Toc80776177)

[References 5](#_Toc80776178)

# Table of Figures

[Figure 1: My code 2](#_Toc80776186)

[Figure 2: Plot Graph 3](#_Toc80776187)

[Figure 3: My second code snippet 4](#_Toc80776188)

[Figure 4: Example code (Gau, 2021) 5](#_Toc80776189)

# Modifying the code

After reading the assignment, I added a minimum\_error variable that is equal to 0.000001. I used a while instead of a for loop to stop when the error is smaller than the minimum error. As seen in the screenshot below.

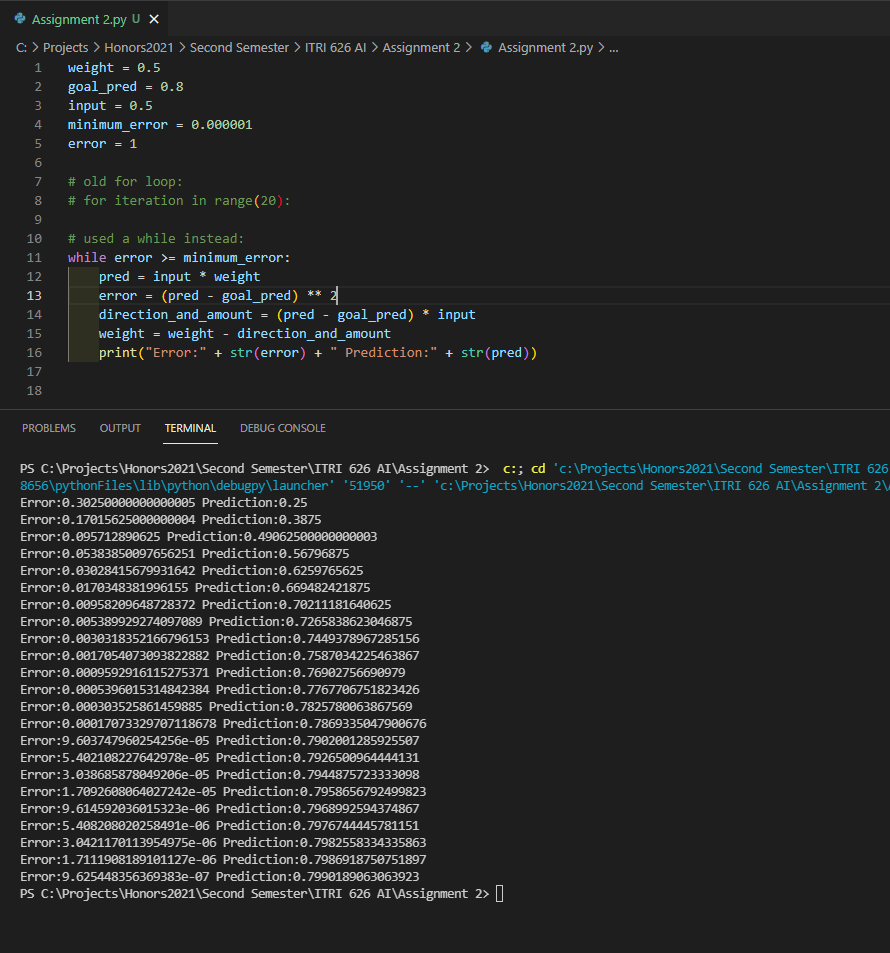


Figure 1: My code

# Is there a better way?

At first, I thought that the assignment can not be that easy. But as I read the assignment in detail, I understood that the main objective is to explain that in practice this is not a good idea.

I started by reading the Grokking Deep Learning Textbook from page 65 (Trask, 2019).

And the first topic I read was on understanding the relationship between the weight and the error, and that one variable changes the other. This was hot and cold learning that we used in assignment 1.

Then they explained the “Box with rods”, whereas you change one rod, the other rod will move with the first rod, this explained in simple terms what derivatives are. “When you move x, how much does y move”. In this assignment, the *goal\_pred* and *input* are fixed we can look at the derivative as a slope, this explains how much does one variable change, when you change the other.

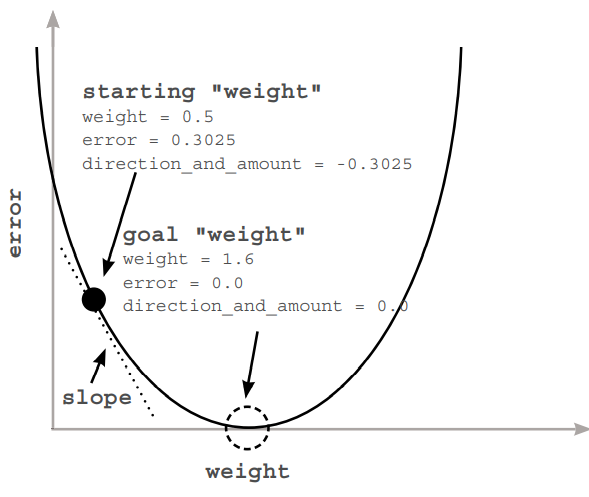


Figure 2: Plot Graph

When you use derivatives, you can pick two variables in a formula, and know how they interact with each other. And when I though about it, a neural network is a couple of weights that you can use to compute an error function.

The textbook also explained a better way, and that is to use derivatives. The slope in the graph represents the derivative. The slope points to the opposite direction to the lowest point of the curve. Thus, if you have a negative slope, you can increase the weight to get the minimum errors.

A derivative gives the relationship between the weight and the error, so you can move the weight in the opposite direction of the derivative to find the lowest weight. This method is called Gradient Descent, and this is one of the better ways to do the assignment, as the neural network learns as it goes.

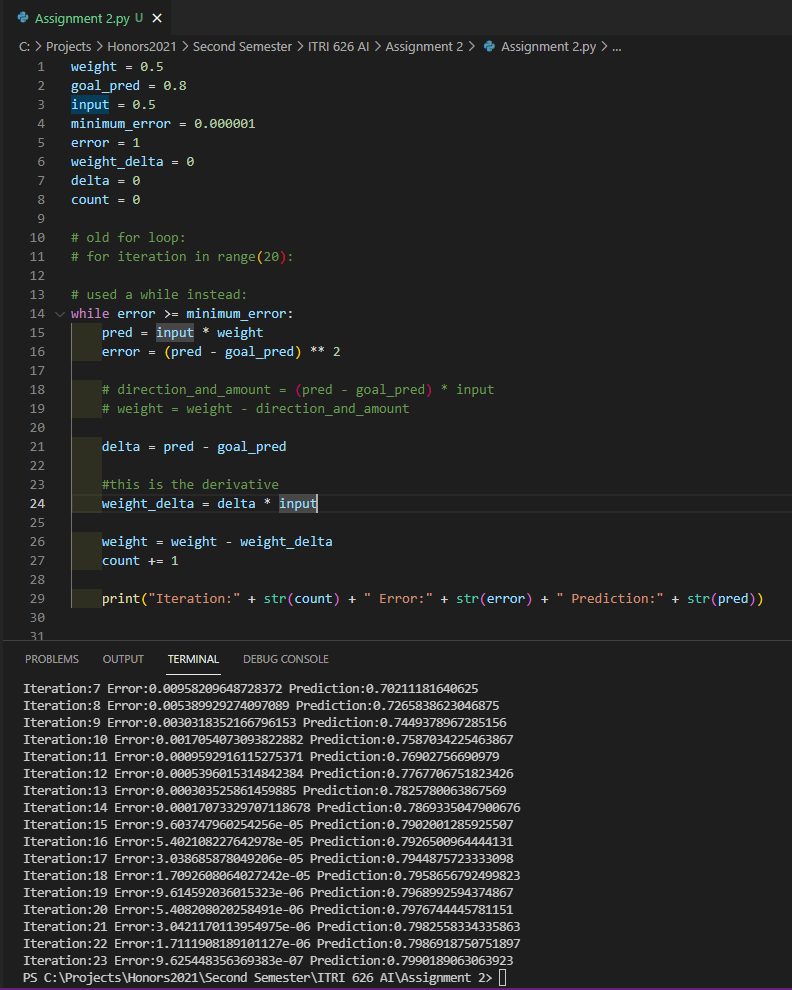


Figure 3: My second code snippet

I started searching ways to create a simple neural network, to give me insight on different networks and what they do to achieve what is asked in the assignment (Gau, 2021). They also used a derivative to solve the problem, as shown in the example below. This gave me insight on how the neural network decides how to manipulate the weight in the right direction, as shown in the snippet below.

The difference between an optimal neural network and an imperfect neural network is the total iterations that it takes for the network to find its goal.

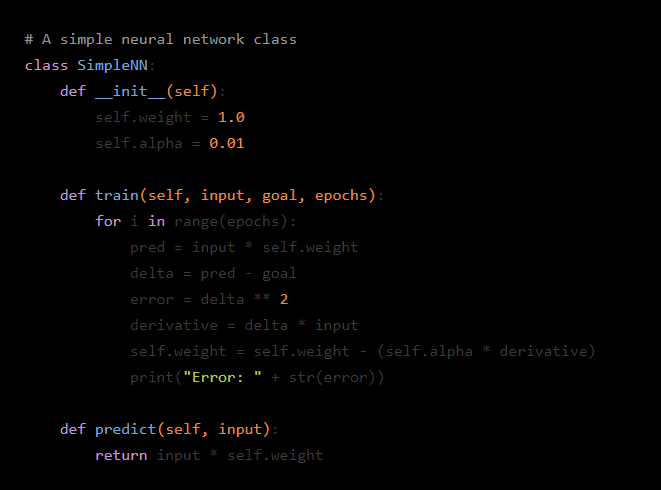


Figure 4: Example code (Gau, 2021)

# References

Gau, L. (2021). How To Create A Simple Neural Network Using Python. <https://hackernoon.com/how-to-create-a-simple-neural-network-using-python-6o2d33yo>

Trask, A. W. (2019). *Grokking Deep Learning*