Deep Learning minor Assignment 1 Report
Robin Tollenaar - 672023 - 18/02/2023

The software

The software consists of 3 files; the main.py, the mainp2.py and the mynetwork.py. These files are used to run a simple neural network. To use this software, you have to run the main.py of the mainp2.py. This will start the process of initializing and training a simple neural network and returning the loss as a measure of quality.

Main files

The main files consist of 2 parts. First, a training data set is set. The set is a list containing a number of training scenarios. For the main.py, this is:
[[[0,1,2,3],0],[[1,1,2,3],2],[[2,1,2,3],4],[[3,1,2,3],6],[[4,1,2,3],8]]

Every scenario consists of 4 input numbers and one expected outcome. The first 4 are fed into the network and then compared to the expected outcome to calculate the loss and teach the network accordingly. For the mainp2.py, this is:
[[0,0],[1,2],[2,4],[3,6],[4,8]]

Here, every scenario consists of 1 input number and one expected outcome. After the training dataset is created, an instance of MyNetwork is created with the necessary parameters. For part one, this means a singular weight and bias and the according node dimensions. Then the train function is called with the training dataset as input.

MyNetwork

The MyNetwork uses only one imported library, NumPy. This allows easier calculations and the use of NumPy arrays. When initialized, all parameters are set using the input and the staring velocity is 0. When the train method is called, the network will start going through all data a number of times and continuously update the weights in order to perfect the output.

Forwards propagation

First, a prediction of the target output is done. This is done using the sigmoid activation function. The weight(s) and bias(es) are given together with the input matrix from the input layer or the previous layer. The output is the new representation of the input layer. This is seen as the prediction made by the network.

Backpropagation

First, the loss is calculated by the mean square method. Then the weights are updated by lowering them by the gradient. The updated weights represent the network's intelligence and accuracy to predicting the actual target output. These new weight(s) are then used in the next forwards propagation.