**Cairo University Faculty of Computers and Artificial Intelligence**

**Third year (2020-2021) Supervised learning (Spring 2021)**

**Assignment 2**

**Programming Language:** Python

**Input:** You get centroid feature vectors from mnist data and use it as input for the neural network (For this assignment you will not be asked how you got the centroid as long as it is correct, and you know what function gets it). Use a subsample of the mnist dataset (use a minimum of 5k samples for training, use a small subset then increase size when you are confident your code is ok). You are allowed to a helper function to reshape your data **only at the beginning of the code and only on the input data** before passing it to your implementation of a neural network. The labels for your data need to be a one hot vector (meaning the value is 1 in the correct class and 0 in the rest, there will be 10 classes so a vector of 10).

**Output**: The output of the neural network should be a vector of 10 values giving the scores of each class. The maximum of these scores will be the class that the input belonged to (each class represents a number from 0 to 9). The final output of the neural network should be the class that the input belongs to, the vector of 10 values is an intermediate step. An optional step is to transform the vector of 10 values to probabilities (not advised during training, but possible), then get the class by getting the maximum value.

**Requirement:** Build your own neural network using only python’s available built-in functions, without any libraries/imports (except a few helper functions written in the restriction part). The neural network should contain **at least 2 layers**: 1 hidden, 1 output.

An example: input 🡪**hidden layer** **1**🡪 hidden layer 1 output 🡪 **output layer** 🡪 output.

The number of neurons in a layer should be of a minimum 4. Do note that the number of neurons is **NOT** fixed, do not write your code based on a fixed number, it should be **dynamic and changeable**.

**Prediction:** The neural network will do the predictions for you if you train it correctly. The extra steps will be to get the correct class.

**Summarized Steps:** You will use the equations that you have taken in the lecture to pass the input through your neural network layers. You will initialize the weights of the layers with random values. You have the equations to calculate the output for all the forward passes which is simple multiplication of the values and the input values/values from previous layer. Do note to save the output values in each layer as you will need them for the back propagation.

Your activation function for the final output layer will be the sigmoid function, and your error function will be MSE between the one hot vector and the prediction vector of the NN. For the back propagation you can also refer to the lecture to know how to implement it. You will propagate back the error and change the weights accordingly. Your learning rate should be set to between 0.01 to 0.05, run the learning for 100 epochs. 1 epoch means a forward and backward propagation of all your samples.  
For the final layer (the output layer), the activation must be sigmoid. This is due to other variations requiring more instructions here (which will lead to confusion and a larger document). For any intermediate hidden layer, the activation is sigmoid also, tanh can be used for hidden layers, but this may require more logic handling in your code and is not advised.

**Restrictions:** No imports should be used except for **math**, **time** and **random** library are the exception and those are included for the use of ceil, floor, round, exp, log, sin, cos, time and random. You should build the neural network and calculate everything that happens in it yourself using lists, arrays, for loops, etc… no shortcuts allowed. From numpy (if you choose to transform your data into numpy arrays) you can use matrix multiplication, addition and subtraction, and you can use transpose, defining an array of zeros or ones. The reshape can be used only on the input and only before passing it to the neural network as mentioned at the start of the document, otherwise it is not allowed. Anything outside what is mentioned is likely not allowed.  
If there is doubt on a certain function email us, if there is something confusing in this document email us.