

## Assignment 2 COMP 5630/6630

### [Question 1]

[10 points]

Derive the update rule and show how to train a 2-layer (1 hidden layer and 1 output layer) neural network with backpropagation for regression using the Mean Square Error loss. Assume that you are using the Sigmoid activation function for the hidden layer. Explain briefly how this is different from the update rule for the network trained for binary classification using log loss.

### [Question 2]

[50 points]

For the given data on Canvas, construct a neural network for the regression task. Your network must have 1 hidden layer and 1 output layer. Use sigmoid to be your activation function for the hidden layer(s). You can choose the number of neurons in each layer using your intuition.

The data is already split to have your input data for training (`X_train.csv`) and testing (`X_test.csv`) and their corresponding target values `Y_train.csv` and `Y_test.csv`, respectively. You can load the data using NumPy as follows:

```
X_train = np.loadtxt("X_train.csv")
```

Implement the backpropagation algorithm and train your network until convergence.

#### Answer the following questions:

1. What is the activation function that you will choose for the output layer? Justify your answer briefly.
2. How many neurons should there be in the output layer? Why?
3. Report the average MSE loss and the accuracy.
4. Plot the loss and accuracy as a function of the number of iterations.
5. What is the effect of the learning rate on the training process? Vary the learning rate to be between 0.001 and 1.0 and plot the resulting accuracy as a function of learning rate.
6. What is the effect of the number of neurons in the hidden layer? To answer this question, you will need to consider and answer the following:
  - a. You will need to vary the number of neurons from 1 to 10. Does the update rule need to be changed/derived again? Why or why not?
  - b. Report your observations by reporting the final loss and plotting the true labels and your predicted labels, along with a brief (2-3 lines) description.
7. What is the effect of the activation functions in the network? Explore two different activation functions other than sigmoid such as tanh, linear, or ReLU.
  - a. Will you need to change the update rule?
  - b. What is the change that you need to make to achieve this experiment?
  - c. Report your observations by reporting the final loss and plotting the true labels and your predicted labels, along with a brief (2-3 lines) description.
8. Split the training data into training and validation set and apply early stopping criteria.
  - a. How the training and validation loss changes as you change the "patience" in early stopping?
  - b. Plot the training vs. validation loss curves. Justify whether your model overfits or underfits as the patience changes.
9. Implement another regularization technique for neural network as discussed in the class. Compare and contrast with early stopping and your chosen regularization technique.

Which one would you prefer for this dataset setting? Justify your answer.

**Submission Requirements:**

You will need to submit the following as a single IPYNB file:

1. Use the “text cells” on Colab to write your report.
2. Include a cell with README instructions at the top of your notebook to note on any dependencies that are required to run your code.

**Note:**

1. If your code does not run on Colab, you will not get any credit for the code segment. We will only grade what is in your report.
  - a. This includes any syntax errors due to indentation, unnamed/unknown libraries that were not listed in the README file, etc.
2. Please submit code only in Python and in the IPython notebook format. You can write your answers as part of the notebook if you do not want a separate report file, but it must be comprehensive.
  - a. Any code not in Python will not be graded at all.