CSE641 Deep Learning
Assignment 01

Date: 14 Feb 2022
Deadline: 28 Feb 2022 **Max Marks: 50**

General Instructions:

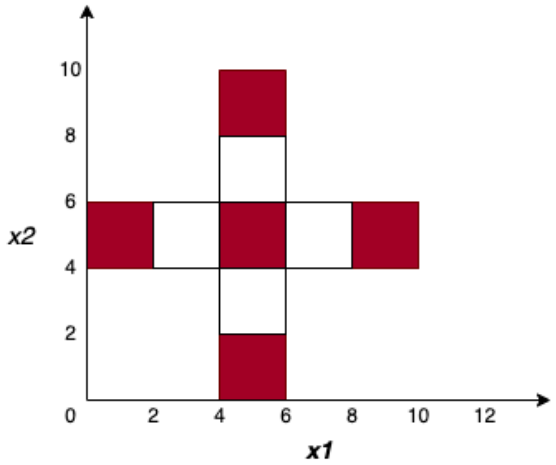
1. Use of any Deep Learning libraries is not allowed.
2. Each group member must do at least one of the following tasks. But both should know the working of all the tasks. (Recommended: Divide the sections among yourselves.)
3. The assignment can be submitted in a group of a maximum of two members.
4. For Plagiarism, institute policies will be followed strictly.
5. Make sure to use Pickle or any other library to save all your trained models. There will not be enough time during the demo to retrain your model. This is a strict requirement. You must upload your best models on Classroom to reproduce your results during the demo. If you are not able to reproduce your results during the demo then no marks will be given.
6. You need to submit README.pdf, Code files (it should include both .py files and .ipynb files), Output.pdf, and models dumped after training.
7. Mention methodology, helper functions, preprocessing steps, any assumptions you may have, and the contribution of each member in README.pdf
8. Mention your sample outputs in the output.pdf.
9. You are advised to prepare a well-documented code file.
10. Submit code, models, readme, and output files in ZIP format with the following name: ***A1_Member1_Member2.zip***
11. Use classroom discussion for any doubt.
12. Most probably you won't be able to run these programmes on your desktops because they are computationally expensive. Use Google colab or other workstations(servers) if possible.

PART I: PERCEPTRON TRAINING ALGORITHM (PTA) (25 marks)

1. **Implement a Perceptron Training Algorithm (PTA)** [10 marks]
   a. Follow the basic algorithm outlined in class, updating the weights when a mistake is made. Calculate the number of steps (i.e. weight updates) necessary for convergence for the following operations using: [6 points]
      i. Two variables: AND, OR
      ii. One variable: NOT
   b. Draw the decision boundary at each step of learning. [2 points]

c. As shown in 1a, PTA can only converge on linearly separable data. Demonstrate that PTA cannot compute XOR operation. How many steps do you need to take to prove it? [2 points]

2. **Implement Madeleine Learning Algorithm** [15 marks]
   a. Using the Madeleine learning algorithm compute the following functions, $f(x_1, x_2)$. Colored (red) regions are 1, rest are 0. Report the number of neurons required. [10 points]
   b. Can you compute $f(x_1, x_2)$ in <= 2 neurons? Justify your answer. [5 points]



PART II: MULTILAYER PERCEPTRON (MLP) (25 marks)

1. **Implement a Multilayer Perceptron** [15 marks]
Implement a Deep Learning Toolkit for training a multi-layer perceptron on the given subset of Fashion-MNIST dataset. Create the neural network using the descriptions given below. Implement forward propagation and backward propagation from scratch on the constructed neural network. You are required to create the *toolkit.py* file and use the following conditions for this question.

- **Network Structure:** Input 784, output 10. You are free to use any number of hidden layers with any number of neurons. Analyze the model and report results for various combinations of your choice. [5 points]
- **Optimizer:** Gradient-descent optimizer (must make *batch_size* parameter = length of entire dataset in this case). [2 points]
- **Activation:** Show results using these activation functions for hidden layers: Sigmoid, Tanh, ReLU. Output layer will always have Softmax activation. [6 points]
- **Initialization:** Random [2 points]

2. **Implement Gradient Descent Optimizers** [10 marks]
   Using the best configuration achieved above, show results for different optimizers. Implement the following optimizers from scratch and do a thorough analysis of the output of each one of them. Use mini-batch size = 64.

   - Gradient Descent with Momentum [2 points]
   - Nesterov's Accelerated Gradient [2 points]
   - AdaGrad [2 points]
   - RMSProp [2 points]
   - Adam [2 points]