Assignment 3
COMP 5600/6600
Due: 10/25/2023 11:59 pm

[Question 1]   **[15 points]**

Design a convolutional neural network in Keras of at least 10 convolutional layers. Use the MNIST dataset for evaluation. You must try three designs as detailed below and provide your observations on the performance of each:

1. A regular CNN where the number of filters in each layer increases as the depth of the network grows i.e., the Lth layer will have more filters than the $(L-1)^{th}$ layer.
2. An inverted CNN where the number of filters in each layer *decreases* as the depth of the network grows i.e., the Lth layer will have less filters than the $(L-1)^{th}$ layer.
3. An hour-glass shaped CNN where the number of filters will increase till the Lth layer and reduce afterwards.

Your goal is to design these networks and optimize them to their best performance by choosing the right hyperparameters *for each network*, such as the learning rate, batch size and the choice of optimizer ('SGD', 'adam', 'RMSProp'). You must provide a detailed report of what values you tried for each hyperparameters, your observations on why the network performed well (or not) and the final accuracy for each network on the MNIST dataset.

You can refer to the Keras documentation for more details.

[Question 2]   **[25 points]**

Implement the LeNet Convolutional Neural Network using Keras. It is a seven-layer network with three convolutional layers, two max-pooling layers and 2 dense layers. The structure is shown below:

Layer 1: convolution layer with 6 convolution kernels of 5x5 with stride 1
Layer 2: max-pooling layer with 2x2 kernels with stride 2
Layer 3: convolution layer with 16 convolution kernels of 5x5 with stride 1
Layer 4: max-pooling layer with 2x2 kernels with stride 2
Layer 5: convolution layer with 120 convolution kernels of 5x5
Layer 6: dense layer with 84 neurons
Layer 7: output layer

Use the 'Adam' optimizer to train your network on the CIFAR-10 dataset for a fixed set of 25 epochs. You can use the built-in functions to load the data. Each image is 32x32x3 matrix and you will have 60,000 images for training and 10,000 for test. There are 10 classes in the dataset each representing an object in the image.

Perform the following analysis and answer each question briefly (3-5 sentences). Use plots and figures as necessary.

1. What is the effect of learning rate on the training process? Which performed best?
2. What is the effect of batch size on the training process? Which performed best?
3. Try different hyperparameters to obtain the best accuracy on the test set. What is your best performance and what were the hyperparameters?
4. Implement an equivalent feed forward network for the same task with each hidden layer containing the same number of neurons as the number of filters in each convolution layer. Use the 'Adam' optimizer to train your network on the CIFAR-10 dataset for a fixed set of 25 epochs. Compare its performance with your LeNet implementation based on the following questions:
   a. What is its performance?
   b. How many parameters are there in this network compared to the LeNet implementation? Are they worth it?

[Question 3]   **[10 points]**

Consider the below matrices as input (X) and convolutional kernel f. Consider that the depth of the input is 1.

$$X = \begin{bmatrix} 7 & 5 & 0 & 0 & 3 & 2 \\ 6 & 4 & 5 & 1 & 4 & 8 \\ 9 & 0 & 2 & 2 & 5 & 4 \\ 6 & 3 & 4 & 7 & 9 & 8 \\ 5 & 7 & 5 & 6 & 9 & 0 \\ 7 & 9 & 0 & 8 & 2 & 3 \end{bmatrix} \quad f = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

Compute the following:
1. What are the dimensions of the input and the kernel (or filter)? How many parameters are there in the kernel f? [2 points]
2. What is the output activation map when you apply the convolutional operation using the filter f on the input X without padding? [4 points]
3. What is the output when you apply a max-pooling operation on the output from the previous question? [4 points]

Note: For parts 2 and 3 in question 3, please provide the actual output. You can work this one out by hand or write your own code to do it. Provide documentation of how you got the outputs.