EEE3314-01

INTRODUCTION TO AI

2019 Fall Semester

PROJECT

Submission Due Date: 12 December to TA office before 5 pm

INSTRUCTIONS:

1. This paper consists of **7 pages** with **3 Questions** only.

2. You may use deep learning library/toolbox to solve the problem. However, you must describe the related information such as name, version etc in the introduction part.

3. **FOUR or NOT LESS than THREE** students form a group, and distribute a similar load to each student.

4. Submit a **report** for each group containing the following items:

   Example:
   Introduction: student-A(100%) A brief introduction of the problem.
   Problems and Solution: student-A(50%), student-B(50%)
   Discussion: student-A(50%), student-B(50%)
   Conclusion: student-B(100%)
   Contributions: provide a table stating the contributions of each student
   Remark: Introduction and conclusion only write once in the report.

5. Use English in the report.

Unless you make prior arrangements with me before the due date, late submission will not be accepted

Question 1: MINIST Features Visualization with CNN

In this question, you will design and implement a simple Convolutional Neural Network (CNN) to classify and visualize the distribution of MNIST ten handwritten digits features in a 2D scatter plot.

The MNIST database of handwritten digits, has a training set of 60,000 examples, and a test set of 10,000 examples. The digits have been size-normalized and centered in a fixed-size image. The original source of MNIST is http://yann.lecun.com/exdb/mnist/, but you may take it wherever available.

1. Your network should follow the specifications as depicted in Table below.

	Stage 1		Stage 2		Stage 3	Stage 4
Layer	Conv	Pool	Conv	Pool	FC	Output
	$(5,20)_{/1,0}$, RELU	$2_{/2,0}$	$(5,50)_{/1,0,}$ RELU	$2_{/2,0}$	500, ReLU	10, softmax

where $(m, n)_{/a,b} \times c$ denotes c cascaded convolution layers with n filters of size $m \times m$, where the stride and padding are a and b respectively. $p_{/r,s}$ denotes the max-pooling layers with window of $p \times p$, where the stride and padding are r and s, respectively.

Train your network. The network would be trained via Cross Entropy with mini-batch GD. Describe the information that associated to the network training:

- Data Preprocessing, data augmentation (If any), initialization
- Optimal hyperparameters such as learning rate schedule, momentum coefficient, L2 coefficient, dropout rate, batch number etc.

Present them systematically in the *table form*.

(a) Plot the following:

- Training and test loss (not classification accuracy) vs epoch.
- Classification accuracy on the training and test set vs epoch.

(b) Fill the table with the final accuracy that you obtained.

Training Accuracy (%)	**Testing Accuracy (%)**

Remark: You should do your best to get the highest possible testing accuracy. The generalization techniques discussed in the lecture would be helpful. The score for this part is proportional to the accuracy that you can obtain. Note the accuracy values should be consistent to the graph shown in (a). If no graph is shown, answer in (b) is discounted.

2. Modify the CNN as follows

	Stage 1		Stage 2		Stage 3		Stage 4	Stage 5
Layer	Conv	Pool	Conv	Pool	Conv	Pool	FC	Output
	$(5,32)_{/1,2}$ x2 Leaky RELU	$2_{/2,0}$	$(5,64)_{/1,2}$ x2 Leaky RELU	$2_{/2,0}$	$(5,128)_{/1,2}$ x2 Leaky RELU	$2_{/2,0}$	2, linear	10, softmax

Train your network. Describe the information that associated to the network training:

- Data Preprocessing and data augmentation (If any)
- Hyperparameters such as learning rate schedule, momentum coefficient, L2 coefficient, dropout rate etc.

(a) After trained, input both training images (60,000) and test images (10,000) to the network, **calculate the training/testing accuracy** and **collect the activation values** (neuron output values) at **stage 4**, plot the activation distribution scatter plot as shown in Figure 1. Note this is a scatter plot for training data only, you must plot for both training and testing data.

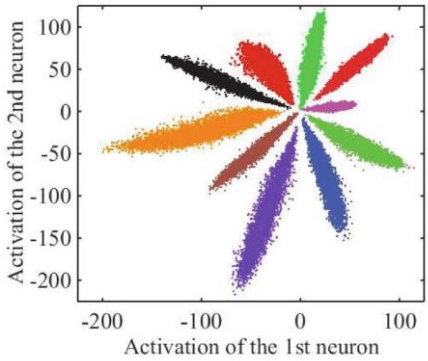

Figure 1: The distribution of deeply learned features in training set, under the supervision of CE loss. The points with different colors denote features from different classes.

(b) Discuss your observation on the plot and the relation with **classification performance**.

Question 2: AlexNet with CIFAR-10

In this question, we will adopt AlexNet discussed in the lecture. We will use CIFAR-10 dataset (https://www.cs.toronto.edu/~kriz/cifar.html). You can find the tutorial for how do you train a CNN with CIFAR-10 according to the specific deep learning library [1]-[4]. Most of the major deep learning libraries come with tutorial with CIFAR10 example.

Unlike question 1 that serves as a warm-up problem, this problem let you play with one of the state of the art networks in deep learning together with a more complicated dataset. You should have good knowledge on AlexNet architecture so that you can modify the hyperparameters in the network according to the CIFAR 10 characteristics.

Train a standard AlexNet with CIFAR 10 via CE loss. Note the following:

- Modify output layer (Softmax layer) to 10 nodes, corresponds to 10 classes in CIFAR-10.
- Use original CIFAR 10 images (32 x 32 x 3) to input AlexNet. With this input size, you will find feature map vanishes at the last conv layer, so modify the stride number, padding etc.
- If you face insufficent memory problem, modify the filter size, channel numbers, number of neurons in the FC layers.
- What you cannot change is the architecture (5 conv layers, 3 max pooling layers and 2 FC layers).

(a) Describe training setup eg. data pre-processing/augmentation, initialization, hyperparameters such as learning rate schedule, momentum, dropout rate, batch number etc. Present them systematically in the **table form**.

(b) Fill in the table below to show the weights and neurons of each layer in your modified AlexNet network.

Layer	# of filters/filter size (Window)/Stride/Zero Padding	Feature Map Size	# of weights	# of neurons (Memory)	# of conv operations	Receptive Field
Input	-	32x32x3				
Conv1						
Pool1						
Conv2						
Pool2						
Conv3						
Conv4						
Conv5						
Pool3						
FC1	-					-
FC2	-					-
Output	-	10				-
Total	-	-				-

(c) Plot the following:

- Training and test loss (not classification accuracy) vs epoch.
- Classification accuracy on the training and test set vs epoch.

(d) Fill the table with the final accuracy that you obtained.

Training Accuracy (%)	**Testing Accuracy (%)**

Remark: The testing accuracy should be at least 70% and above. With proper tuning the network hyperparameters, you can achieve more than 80%. The generalization techniques discussed in the lecture would be helpful. The score for this part is proportional to the accuracy that you can obtain. Note the accuracy values should be consistent to the graph shown in (b). If no graph is shown, answer in (c) is discounted.

(e) Discuss the issues that you experienced and the effort that you've taken to overcome them. This could be non-convergence training or/and poor testing accuracy etc.

Remark: If your machine is slow in solving this question due to CPU or poor GPU, consider Google Colab if you are using Python. See https://towardsdatascience.com/getting-started-with-google-colab-f2fff97f594c

Question 3: Transfer Learning

Now we want to see whether the features of AlexNet developed in Question 2 can generalize to other dataset. Here we use **imdb_wiki face dataset** **(subset)** that consists of 100 subjects with 30 samples per subject as target dataset. Note this is the case where source and target data are dissimilar and very small in size, relative to CIFAR 10.

(a) First, based on the model that has been trained with CIFAR 10, **re-train the last (softmax) layer** in the trained network with face training samples (20 samples per subject). This is equivalent to you freeze the rest of layers in the network and replace with a new classifier that tune to face dataset. Note number of output nodes should change to 100 instead of 10. This serves as a baseline model. Evaluate the baseline model with face test set, which composed of 10 samples per subject.

(b) Then, try to fine tune a few chosen convolution and FC layers and train the softmax classifier on top with face training images. Describe clearly what you choose to fine tune. Evaluate the model with face test set.

For each model,

(i) Describe clearly the learning rate and other hyperparameters that you set.
(ii) both training and test loss vs. epoch.
(iii) both classification accuracy on the training and test set vs. epoch.

Fill in the table

	Training Accuracy (%)	**Testing Accuracy (%)**
Baseline Model		
Fine tune Model		

(c) Discuss your observation in terms of generalization performance of transfer learning.

Remark: Almost every deep learning library comes with tutorial of CNN tuning such as Tensorflow [5], MATLAB [6], Keras [7] etc.

References:

[1]. https://www.mathworks.com/help/nnet/examples/train-residual-network-on-cifar-10.html.

[2]. https://www.tensorflow.org/tutorials/deep_cnn

[3] https://blog.plon.io/tutorials/cifar-10-classification-using-keras-tutorial/

[4] http://caffe.berkeleyvision.org/gathered/examples/cifar10.html

[5] https://kratzert.github.io/2017/02/24/finetuning-alexnet-with-tensorflow.html

[6] https://www.mathworks.com/help/nnet/examples/transfer-learning-using-alexnet.html.

[7] https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html