EE 569: Homework #5

Issued: 03/28/2024   Due: 11:59PM, 04/14/2024

General Instructions:

1. Read *Homework Guidelines* for information about homework programming, write-up, and submission. If you make any assumptions about a problem, please clearly state them in your report.
2. You are required to use PYTHON in this assignment. You are encouraged to use an interface tool called PYTORCH. You can use KERAS, built upon TensorFlow, as an alternative choice if you feel more comfortable. We only provide a sample tutorial on PYTORCH.
3. DO NOT copy codes from online sources, e.g., Github.
4. You must understand the USC policy on academic integrity and penalties for cheating and plagiarism. These rules will be strictly enforced.

Problem 1: CNN Training on LeNet-5 (85%)

In this problem, you will learn to train a simple convolutional neural network (CNN) called the LeNet-5, introduced by LeCun et al. [1], and apply it to three datasets **MNIST** [2], **Fashion-MNIST** [3] and **CIFAR-10** [4].

LeNet-5 is designed for handwritten and machine-printed character recognition. Its architecture is shown in Fig. 1. This network has two *conv* layers and three *fc* layers. Each conv layer is followed by a *max pooling* layer. Both *conv* layers accept an input receptive field of spatial size *5x5*. The filter numbers of the first and the second *conv* layers are 6 and 16, respectively. The stride parameter is one, and no padding is used. The two *max pooling* layers take an input window size of *2x2* and reduce the window size to *1x1* by choosing the maximum value of the four responses. The first two *fc* layers have 120 and 84 filters, respectively. The last *fc* layer, the output layer, has a size of 10 to match the number of object classes in the dataset. Use the popular ReLU activation function [5] for all *conv* and all *fc* layers except for the output layer, which uses softmax [6] to compute the probabilities.

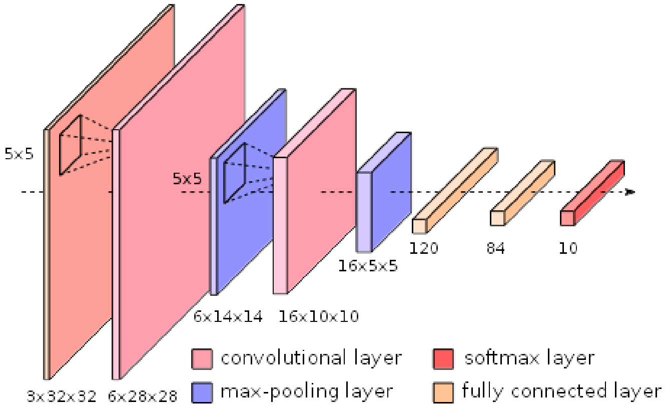

Figure 1: A CNN architecture derived from LeNet-5

The following table shows statistics for different datasets:

	Image type	Image size	# Class	# training images	# testing images
MNIST	Gray	28*28	10	60,000	10,000
Fashion-MNIST	Gray	28*28	10	60,000	10,000
CIFAR-10	Color	32*32	10	50,000	10,000

(a) CNN Architecture (Basic: 10%)

Explain convolutional neural networks' architecture and the operational mechanism by performing the following tasks. You answer to this question should be no more than 2 pages.

1. Describe CNN components in your own words: 1) the fully connected layer, 2) the convolutional layer, 3) the max pooling layer, 4) the activation function, and 5) the softmax function. What are the functions of these components?
2. What is the over-fitting issue in model learning? Explain any technique that has been used in CNN training to avoid over-fitting.
3. Explain the difference between three activation functions: ReLU, Leaky ReLU, and ELU.
4. Read official documents of three loss functions: the L1 loss, the MSE loss, and the BCE loss. List applications where those losses are used and state why they are used in those specific cases.

Show your understanding as much as possible in your own words in your report.

(b) Compare classification performance on different datasets (30%)

Train the CNN given in Fig. 1 using the training images of MNIST, then test the trained network on the testing images of MNIST. Compute and draw the accuracy performance curves (epoch-accuracy plot) on training and test datasets on the same figure. You can adopt proper preprocessing techniques and random network initialization to simplify your training.

1. Plot the performance curves under three different yet representative hyper-parameter settings (optimizers, initialization of filter weights, learning rate, decay, etc.). Discuss your observations and the effect of different settings.
2. Find the best parameter setting to achieve the highest accuracy on the test set. Then, plot the performance curves for the test and training sets under this setting. Your testing accuracy should be no less than 99%.
3. Repeat 2 for Fashion-MNIST. Your best testing accuracy should be at least 90%.
4. Repeat 2 for CIFAR-10. Your best testing accuracy should be at least 65%.

5. Compare your best performances on three datasets. How do they differ, and why do you think such a difference exists?

Note: for each setting, you need five runs. Report the [best test accuracy among five runs, mean test accuracy of 5 runs, standard deviation of test accuracy among five runs] to evaluate the performance.

(c) Evaluation and Ablation Study (30%)

Note: you may use the best setting you found in Problem 1(b) on each dataset.

1. (Confusion Matrix) Generate the normalized confusion matrix for the 10 classes on the testing set. What are the top three confused pairs of classes? Show one example for each of these three pairs. Do this for all the datasets in 1(b). Describe your observations and explain.
2. (One-vs-Rest multiclass ROC) The final softmax function will yield a confidence score for each class; they can be used to compute the ROC curve in a one-vs-rest manner. For CIFAR-10, plot the ROC curve for each class and discuss its difference against accuracy. (See reference [7])
3. The area under the ROC Curve (AUC) is another important metric to evaluate the classifier. It computes the area under the ROC curve, which gives numerical values to compare the performance. Compute the AUC for CIFAR-10, which is defined as the weighted sum of AUC values for all One-vs.-Rest multiclass ROC's AUC.

(d) Classification with noisy data (15%)

Data in real-world applications could be noisy with wrong labels. Symmetric Label Noise (SLN) is the type of labeling noise where $\alpha\%$ of the data with the rue label of class i is labeled as other classes $j \neq i$ with uniform probability. For example, in a 3-class classification problem, the normalized confusion matrix between the true label and the noisy label is close to the following format, where epsilon is the noise level (say, 40%):

$$\begin{bmatrix} 1-\epsilon & \frac{\epsilon}{2} & \frac{\epsilon}{2} \\ \frac{\epsilon}{2} & 1-\epsilon & \frac{\epsilon}{2} \\ \frac{\epsilon}{2} & \frac{\epsilon}{2} & 1-\epsilon \end{bmatrix}$$

Now you'd like to synthesize the Symmetric Label Noise on the training set of Fashion-MNIST and investigate the performance of neural networks under different noise levels.

1. Implement the Symmetric Label Noise. Describe your method and show the normalized confusion matrix for $\epsilon = 50\%$.
2. Train LeNet-5 with the noisy training set and measure the testing accuracy. Try $\epsilon = 0\%, 20\%, 40\%, 60\%, \text{and } 80\%$. Draw the curve of [testing accuracy vs. ϵ]. Note that for each ϵ, five runs are needed to calculate the mean and standard deviation of the testing accuracy, which are then used to draw your plot.

3. Discuss your observations on the result of 2 and analyze.

Problem 2: Vision Transformer (ViT) (15%)

In this problem, you will learn to train a simplified version of Vision Transformer introduced in [8], and apply it to the **MNIST** [2] with the reference code [9].

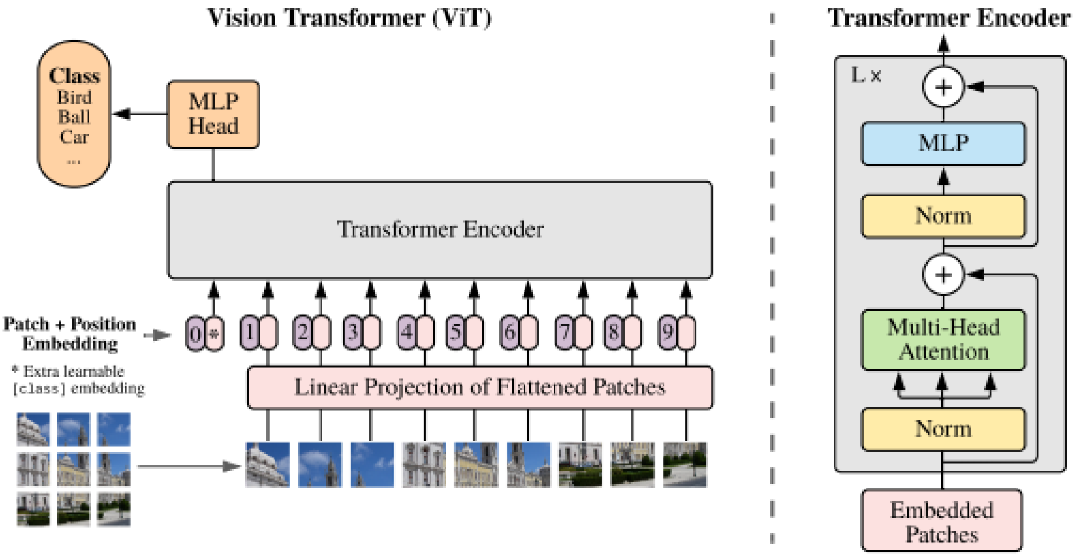

Figure 2: A Vision Transformer architecture introduced in [8]

(a) ViT Architecture (8%)

1. Describe ViT components in your own words: 1) patch embedding, 2) positional encoding, 3) multi head attention. What are the functions of these components? You answer to this question should be less than 1 page. Figures and diagrams are welcomed.

(b) Compare classification performance over MNIST dataset (7%)

In this part, you may use the code from [9], which is a simplified version of ViT. Please follow the settings in the readme file and run it over the MNIST dataset. Set the training epoch to be 20 (This takes about 1 hour to train over Google Colab.) Report and discuss your experimental results.

1. Report the test accuracy and draw the training and testing acc curve of the 20 epochs.
2. Estimate the number of parameters of this simplified ViT model. You may use the model settings to calculate the model size or directly check the file size after saving the model.
3. Compare the performance of ViT against the CNN results in Problem 1. Is ViT doing better than CNN? Discuss the possible reasons for your observations.

References

[1] LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324
[2] http://yann.lecun.com/exdb/mnist/
[3] https://github.com/zalandoresearch/fashion-mnist
[4] https://www.cs.toronto.edu/~kriz/cifar.html
[5] ReLU https://en.wikipedia.org/wiki/Rectifier_(neural_networks).
[6] Softmax https://en.wikipedia.org/wiki/Softmax_function
[7] https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html
[8] [2010.11929] An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale (arxiv.org)
[9] GitHub - s-chh/PyTorch-Vision-Transformer-ViT-MNIST-CIFAR10: Simplified Pytorch implementation of Vision Transformer (ViT) for MNIST dataset.