February-June 2022 Semester
CS671: Deep Learning and Application
Programming Assignment 2

Date: March 30, 2022
Deadline for submission of code and report: Monday, April 18, 2022, 10:00 PM

The objective of this programming assignment is to deepen your understanding of the **optimizers for backpropagation algorithms** and **autoencoders**. One of the tasks is to train the fully connected neural network (FCNN) using different optimizers for the backpropagation algorithm and compare the number of epochs that it takes for convergence along with their classification performance. Another major task includes building an autoencoder to obtain the hidden representation and use it for classification.

You are given the subset of the MNIST digit dataset for the same. Each group is given 5 classes. Given dataset is train-validation-test separated. Every image is of the size 28 x 28. **Flatten each image to represent it as a vector of 784-dimension (28 x 28)**. The tasks for this assignment are as follows:

1. **Tasks based on different optimizers:**
   a. Develop an FCNN with 3 hidden layers. Use cross-entropy loss. Experiment with different a number of nodes in each of the layers. Train each of the architectures using (a) stochastic gradient descent (SGD) algorithm - (`batch_size`=1), (b) batch gradient descent algorithm (vanilla gradient descent) – (`batch_size`=total number of training examples), (c) SGD with momentum (NAG) – (`batch_size`=32), (d) RMSProp algorithm – (`batch_size`=32), and (e) Adam optimizer – (`batch_size`=32). Use the *difference between average error of successive epochs fall below a threshold 10^{-4}* as convergence criteria. Consider $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 10^{-8}$ for Adam optimizer. Consider momentum parameter as 0.9, learning rate as 0.001 and $\beta = 0.99$ for RMSProp. You are free to tune these parameters.
      i. Observe the number of epochs considered for convergence for each of the architectures. Tabulate and compare the number of epochs considered by each of the optimizers for each architecture.
      ii. Present the plots of average training error (y-axis) vs. epochs (x-axis).
      iii. Give the training accuracy and validation accuracy for each of the optimizers in each of the architectures.
      iv. Choose the best architecture based on validation accuracy. Give the test confusion matrix and test classification accuracy along with training accuracy and confusion matrix for the chosen best architecture.

2. **Tasks based on autoencoder**:
   a. Develop an autoencoder that learns a compressed representation of the input features for a classification predictive modeling problem. Train the autoencoder using Adam optimizer. Use sigmoid activation function for the nodes in all the hidden layers. Use the *difference between average error of successive epochs fall below a threshold* 10^{-4} as convergence criteria.
      i. Build autoencoders with one hidden layer and 3 hidden layer architectures. For each architecture, **experiment with a different number of neurons** in hidden layers including the compressed layer. Present the mean squared error (MSE) i.e., the average reconstruction error for training data as well as validation data for each of the architectures.
      ii. Choose the best architecture for (a) encoder with one hidden layer and (b) encoder with 3 hidden layer architectures based on validation error. Give the test reconstruction error for the chosen best architectures.
      iii. Present the plots of average training reconstruction error (y-axis) vs. epochs (x-axis) for the best architecture for (a) encoder with one hidden layer and (b) encoder with 3 hidden layer architectures.
      iv. Take one image from the training set and one image for the validation set, from each of the classes, and give their reconstructed images for each of the experiments.
      v. *Classification using the compressed representation from the* ***encoder with one hidden layer***:
         - Present each training data to the best encoder with one hidden layer and save the output of the hidden layer (compressed layer). This gives the compressed representation of training data. Similarly obtain the compressed representation of validation and test data.
         - Build the FCNN using Adam optimizer for classification. Experiment with the different number of hidden layers and the different number of neurons in each hidden layer. Select the best architecture based on validation accuracy. Report the train, validation & test accuracy along with the confusion matrix and compare the results with the best result from Question 1.
      vi. *Classification using the compressed representation from the* ***encoder with 3 hidden layers***: Repeat the experiments as described in previous question (Question 2.a.iii)
      vii. ***Weight visualization***: For the best compressed representation in one hidden layer autoencoder, plot the inputs as images that maximally activate each of the neurons of the hidden representations (plot of weights from the input layer to the compressed layer).

b. Develop a denoising autoencoder with 20% noise and 40% noise for the best one-hidden layer autoencoder architecture from Question 2.a.i. Follow the concepts discussed in the class to corrupt the inputs on the fly during training.
   i. Take the same images used in Question 2.a.ii. and give their reconstructed images
   ii. Give the classification accuracy for classification using the compressed representation. Compare their performance with that of the corresponding vanilla autoencoder from Question 2.a.iii.
   iii. ***Weight visualization***: Plot the inputs as images that maximally activate each of the neurons of the hidden representations obtained using both the denoising autoencoders (plot of weights from the input layer to the compressed layer). Compare it with that of the images obtained in Question 2.a.v.

Instructions:

- **Each group of students must use the dataset identified for that group only.**
- **You can use deep learning APIs (Tensorflow, PyTorch, Keras, etc.).**
- **Codes should be self-written. If copied, zero will be awarded.**
- **Report by a team should include observations about the results of studies.**
- **Report should be in PDF form.**
- **Your codes and report should be uploaded as a single zip file on Moodle.**
    - **Strictly name the folder as Group<*number*>-Assignment2**
      Example: Group*01*-Assignment2
    - **Strictly name the zip file as Group<*number*>-Assignment2.zip**
      Example: Group*01*-Assignment2.zip

If you don’t follow the instructions given, the submission will not be accepted.