Total Marks: 250 **Advanced Machine Learning** **Assignment-2**

Question 1. Dictionary Learning

This question requires you to download and use the CIFAR-10 dataset (https://www.cs.toronto.edu/~kriz/cifar.html). With the pre-defined training and testing splits for the 10-class problem, perform the following:

1. Train the Dictionary Learning algorithm on the entire training set. Train the model for 100 iterations, and plot the training error per each iteration. You *cannot* call an in-built Dictionary Learning function or a toolbox. You are required to implement the algorithm. Keep the weight parameter for sparsity (lambda) as 1, and the number of atoms as 1000. For *n* training samples, the loss function of Dictionary Learning is given as:

$$\min_{\mathbf{D},\,\mathbf{A}} \frac{1}{n} \sum_{i=1}^{n} \left(\frac{1}{2} \left\| \mathbf{x}^i - \mathbf{D}\boldsymbol{\alpha}^i \right\|_F^2 + \lambda \left\| \boldsymbol{\alpha}^i \right\|_1 \right)$$

   where, x^i refers to the i^{th} training sample, and $(alpha)^i$ refers to its representation.
2. Visualize 100 atoms of the learned dictionary, and reconstructions of any 100 randomly chosen samples. Is the dictionary able to learn any meaningful atoms? Comment on the reconstructions.
3. Repeat steps 1,2 by changing the weight parameter (lambda) to 0, 0.5, 10, and 100. Comment on any difference you observe between the convergence, final error, and visualizations of the reconstructions of the above four configurations?
4. Select the best model from the above (the one with the least final error), and perform classification of the test set. Use the representations for performing classification with Cosine distance. Report the classification performance.
5. Implement Label-Consistent Dictionary Learning (Slide 9 of Lecture-1 on DL). Report the classification performance on the test set. Is it substantially different from the ones reported earlier? Comment upon the benefit of incorporating labels during Dictionary Learning, versus learning a single Dictionary for all samples, and learning different dictionaries for each class. Conceptually, which one should have performed the best, and practically, which one did?

Note: Try and explore optimization techniques beyond Gradient Descent. K-SVD and solving via least squares + OMP are some ideas we discussed.

[20+10+20+20+30 marks]

Question 2: Autoencoder

- For this question, use the STL10 dataset. Link to the dataset: https://cs.stanford.edu/~acoates/stl10/
- Use only the training and testing images for part (1), part (2) and part (3), and use unlabelled, training, and testing set for part (4). To reduce computation, you may convert images to grayscale.

Create an autoencoder model of two hidden layers, having dimension [n/2, n/4] where n is the dimension of the input vector. Apply Relu activation function on all encoder and decoder layers. Train two separate models using only training set in the following manner:

A. Train the entire network together.
B. Perform training of the model in a stacked manner.

With the above two trained models, perform the following:

1. Visualize the reconstructions from both the models and comment on which model appears better trained for the task of reconstruction.
2. Use the hidden most layer for performing feature extraction. Train a two-layer neural network [k/2, k/4] on the extracted features, where k is the dimension of the feature. Which of the two models appear better trained for the task of classification? Can you think of any reason for the above behavior?
3. In part (2), we would now add regularization! Regularization is a widely used technique in deep learning to prevent the model from overfitting. In each of the encoding and decoding layers of both autoencoder, add:
    a. L1 regularization
    b. L2 regularization
    c. Elastic net (λ 1 x L1 + λ 2 x L2)

    Does this help in improving classification accuracy? Now, instead of autoencoder layers, add each of the regularizations in layers of the neural network. What do you observe now? (**Note**: Implement regularization from scratch).
4. In this part, the aim is to introduce supervision in autoencoder. For the proposed autoencoder architecture, derive the output X_E from the ReLu activated encoder [last layer of encoder] and apply Softmax loss. Let this loss be L1. The autoencoder already has a reconstruction loss $L2 = min\ ||X - X_D||_2$ obtained using input X and decoder output X_D. Thus, from scratch, train the Multi-loss L= (L1+L2) autoencoder for 100 epochs to obtain a supervised autoencoder, where loss function L should be minimized. Show a plot of all 3 losses over 100 epochs [loss 1(L1), loss 2(L2), and combined loss(L1+L2)] and analyze it. Compare the performance with the performance obtained in part (2).

[20+20+30+30 marks]

Viva+Report: [50 marks]

Submission Policy and Requirements

1. Any kind of plagiarism is not accepted. We will strictly follow institute policies for plagiarism.
2. Recommended programming languages: MATLAB, python.
3. You may use any external libraries or GitHub codes. However, the evaluation will test your knowledge of the algorithm and the choice of hyperparameters. Do cite the libraries/codes.
4. **Submission should include:** Working code for each of the part separately and a report to show the analysis of results in each of the parts.

Assessment criterion

The assessment will be done on basis of the following components:

1. Working codes
2. Analysis and clarity of results (drawing comparisons across different parts) & clarity of the report
3. Understanding the theoretical concepts and the choice of hyperparameters.