PRNN - 24, Assignment 1

Prof Prathosh A P

March 4, 2024

General Instructions:

- For Each of the problems you can find the data in 'Data' folder shared with you, traverse to the relevant folder, and choose the data corresponding to your group as per the group ID in the Excel sheet.(Each data is different and if you choose the wrong data you will be penalized)
- You are supposed to submit a single Jupiter notebook with all the solutions made into separate blocks.
- No ML library other than **numpy and matplotlib** should be used, failing which will attract zero marks.
- Only for the question on SVM, LIBSvM may be used.
- The final evaluation does not depend on the accuracy metrics but is based on the quality of your experiments and observations thereof.
- We will run a plagiarism check on both your report and the codes. Any suspicion of copying would lead to a harsh penalty from negative marks in the assignment to a failing grade in the course, depending upon the severity. Therefore, kindly refrain from copying others' codes and/or reports.

1 General Tasks

- Implement error backpropagation algorithm as Python classes for (a) Fully connected multi-layer feed-forward neural network and (b) Multilayer Convolutional Neural Networks. The following hyperparameters should be the class attributes: loss function, dimensionality of the input/output variables, number of hidden layers, nodes in each layer, number of kernels and their sizes, and padding/stride size.
- For all the tasks below, Set the hyper-parameters to overfit the data
- Regularize the models with at least 3 regularization techniques and plot the bias-vairance curves.

2 Regression Tasks

- Consider the same data as given in A1 and Solve all the regression tasks with the same metrics as in A1 by using multi-layer perceptrons (MLPs) and report the metrics.

3 Classification Tasks

- For all the classification tasks in A1, use MLP and repeat the experiments with at least two loss functions.
- Implement SVMs both with and without slack formulations. Experiment with at least 3 Kernels and grid search on hyper-parameters on different kernels and report your observations (you can use standard Python library, LibSVM and need not implement SMO). For multi-class classification, implement a one-vs-rest approach.

- Consider the Kuzushiji-MNIST dataset consisting of 28x28 grayscale, 70,000 images from the Japanese Hiragana script. The dataset can be downloaded from - This link. Solve the 10-class classification problem on this dataset using Logistic Regression, SVM with the Gaussian Kernel, MLP, and CNN. Compare the performance of each of the models with respect to model size vs performance.