ELL 409/784 2020 Assignment 2

Prathosh A. P.

December 19, 2020

1 SVM and Kernel Machines

1. Consider the binary classification data in problem 1 of Assignment 1. Try an SVM on the same problem with the following kernels: (a) Linear, (b) Polynomial (with degree 2, 3 and 4), (c) Gaussian (do a grid search on C and Gamma parameters). You may use the LibSVM library, no need to explicitly implement SMO (see question 1.4)
2. Plot the decision boundaries in the best case with the support vectors marked.
3. Repeat the previous step by removing the support vectors from the dataset. Record your observations.
4. (Bonus, Not mandatory) - Implement the SMO algorithm and plot the time taken by the two as a function of the dataset size.

2 Neural Networks

Consider the tiny-imagenet dataset (https://www.kaggle.com/c/tiny-imagenet/overview). Implement the following on using the imagnet dataset.

1. Starting from scratch, hard-code the error-back propagation algorithm for a feed-forward neural network.
2. Over-fit the tiny-imagenet using a fully-connected Feed-forward Neural network (you may need to do it in Google Colab).
3. Repeat the first part by hard-coding the following regularizers: (a) L1 and L2, (ii) Dropout, (iii) Batch-normalization, (iv) Early stopping.
4. Implement the following standard Convolution Neural Network architectures using Pytorch: LeNET, ALexNet, VGG-16.
5. Take any one of the above architectures and compare following optimizers: SGD, ADAM and SGD with Momentum.

3 Learning++

In this problem, we shall focus on some interesting issues that arise in practical scenarios. For this problem you need to consider the following two datasets: MNIST and SVHN with overlapping class-labels. Implement the following:

1. Take any 3 classes from the MNIST dataset, with the ratio of 70:25:5 (class-prior). With 80:20 random train-test split, build any two classifiers using cross-entropy and MSE as loss. Try to modify the loss-function to improve the accuracy.

2. Implement K-means clustering on SVHN with different Ks (3, 5,10, 15 and 20). Re-label the dataset considering each cluster as a class and try supervised learning on the relabelled data. Record your observations.

3. Train a classifier on the MNIST dataset and test it on the SVHN and vice-versa. Try to improve the accuracy of a classifier trained on MNIST and tested on SVHN assuming that the samples of SVHN dataset is accessible during training but without the labels (samples of SVHN can be used during training MNSIT but not their labels).

4. Consider the MNIST dataset. Suppose you know that there are 10 classes in it but only have access to 5 of them during training. E.g, you have samples of digits 1-5 but haven't seen the samples of digits 6-10. Build a 10-class Neural network (FCN or CNN) in such conditions and test it on the novel classes. Try novel methods to improve the accuracy on the novel classes.

5. Try 2D PCA and t-SNE on SVHN and compare them.