**ENPM703- Assignment-1**

**Part1: SVM**

## CIFAR-10 Dataset

Contains collection of 60000 RGB images with dimensions 32x32. Thus, loading the training dataset forms an array of size (50000x32x32x3). This data is pre-processed to a row matrix of dimensions (50000x3072) by straightening up the pixel values. We use a sample of 49000 images for training, 1000 images for validation and 1000 images for testing are used.

## SVM

SVM is a classification technique which transforms the input data (not linearly separable) to a high dimensional space to make them a linearly separable data. Then an optimal separating plane is found by optimizing the weights of different parameters.

## SVM Loss

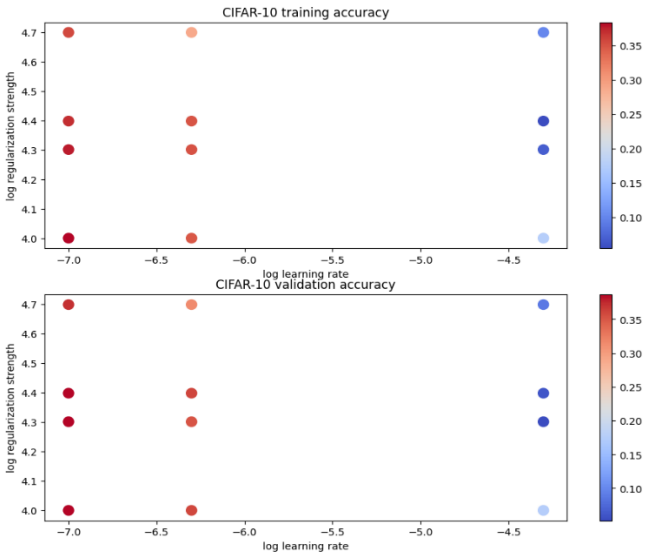
The optimal separable plane is found by a parametric approach, where in we project the flattened to the weights of the plane (ie. taking dot product) and get the scores for all individual class. Then the scores are compared with the ground truth through a loss function and this loss is minimized using gradient descent algorithm.

In this assignment we have computed SVM loss by iterating through each training data and each class, but this approach is time complex. Thus, we use a vectorized approach for finding the loss and its corresponding gradient with respect to W.

## Stochastic Gradient Descent

This gradient which we found is used to find the global minima of the loss function, thus minimizing the loss. Here we sample mini batches from the training set and for each iterations unique batches are used to find the loss and gradient. At the end of each iterations the weights are adjusted based on the leaning rate, which is one of the hyperparameter.

## Cross Validation



The hyperparameter that needs to be optimized are learning rate and regularization strength. Here we initialize few values for the hyperparameters and we iterate through different combinations to find the optimal value. We also store the accuracy values of each combination in a dictionary and visualize to get a clear view of the combinations.

The combination lr = 1.000000e-07 reg = 2.000000e+04 has the maximum accuracy.

**Visualizing Hyperplanes**

When the weights are reshaped and transformed in RGB images, it resembles the visual characters of each class. So, when a test image is projected on these hyperplanes, the plane with maximum resemblance is chosen as the predicted class (ie. image having the maximum dot product with the weights)