Assignment-1 Report

Name: Vasigaran Senthilkumar

We use CIFAR-10 image classification data set for this task. There are various batches and each batch contains 10000 images with 3072 collar dimensions. There are 10 class labels . We use neural network with only one layer using mini-batch gradient descent applied to a cost function that computes the cross-entropy loss of the classifier applied to the labelled training data and an L_2 regularization term on the weight matrix.

d- No of features or no of image pixels=3072
n-no of input samples
k-no of class labels
X- Input data of the form (d x n)
Y- one-hot representation of the label for each image. of form (k x n)

W-weight of form (k x d)
B - bias (k x 1)
Initially during training we initialise the entry with Gaussian random values with zero mean and standard deviation .01 in W and B

S= WX+b

S is of form (k x n)

P=softmax(S) of form (k*n)
P is vector of probabilities for each class label

Then we have the loss function to minimise cross entropy loss and a regularisation term W

$$J(\mathcal{D}, \lambda, W, b) = \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x}, y) \in \mathcal{D}} l_{\text{cross}}(\mathbf{x}, y, W, \mathbf{b}) + \lambda \sum_{i,j} W_{ij}^2$$

In order to check if the gradients in the function is the correct gradients, we compare the gradients obtained by the analytical method with the gradient calculated with the numerical method. I tried to find the relative error for the first image using only 20 dimensions from the training data.
For weights W there are 99.5% of relative errors below 1e-6
For bias b there are 100.0% of relative errors below 1e-6
And the same was tested with a higher number of images and dimensions to check if the relative error was minimal.

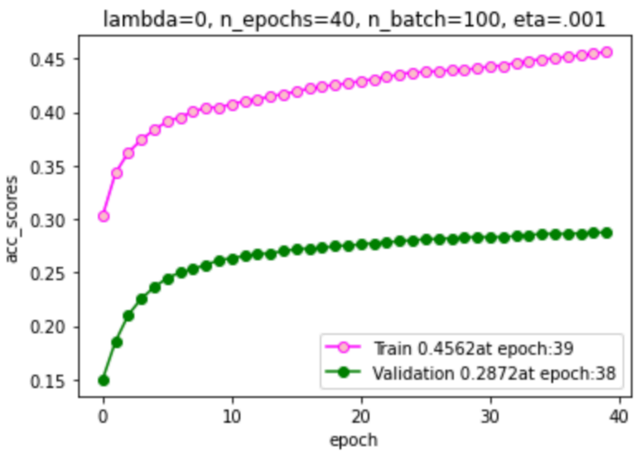
Since this is the mini batch gradient descent, we have initial random parameters W and b. Then the estimate W and b is updated with the following equations after each batch is processed.

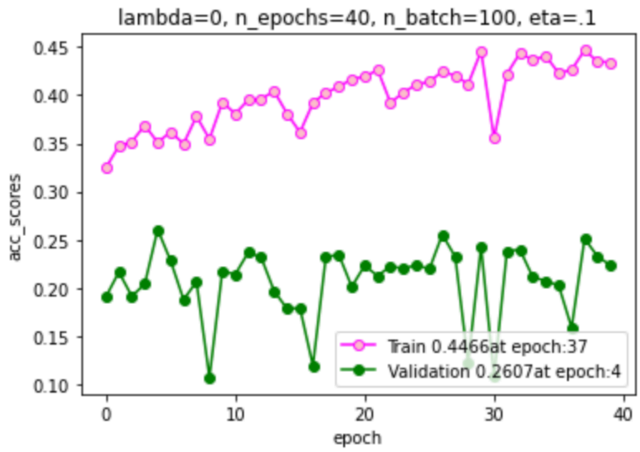
$$W^{(t+1)} = W^{(t)} - \eta \left. \frac{\partial J(\mathcal{B}^{(t+1)}, \lambda, W, \mathbf{b})}{\partial W} \right|_{W=W^{(t)}, \mathbf{b}=\mathbf{b}^{(t)}}$$

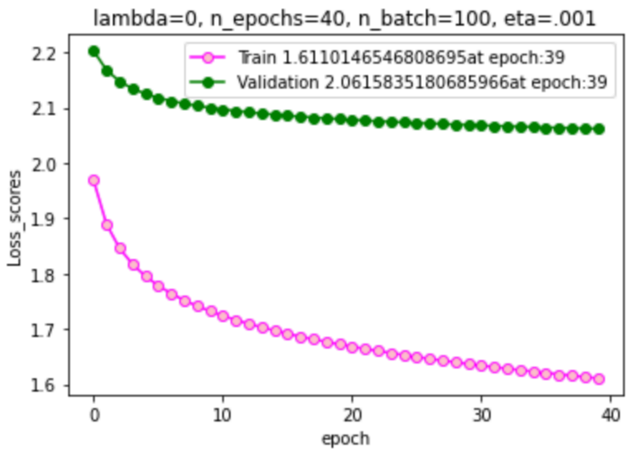
$$\mathbf{b}^{(t+1)} = \mathbf{b}^{(t)} - \eta \left. \frac{\partial J(\mathcal{B}^{(t+1)}, \lambda, W, \mathbf{b})}{\partial \mathbf{b}} \right|_{W=W^{(t)}, \mathbf{b}=\mathbf{b}^{(t)}}$$

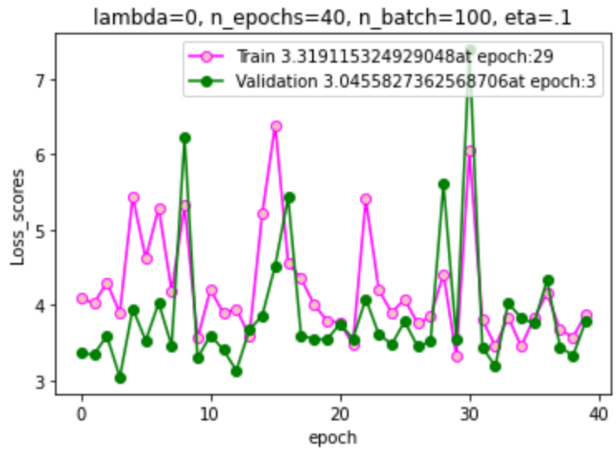
where eta is the learning rate .
In the mini batch gradient descent, we find the loss and accuracy after each epoch for training as well as a validation data set.

Comparing the cost function graph:
The accuracy scores and loss scores for different values of learning rate (eta), batch size, number of epoch, regularisation lambda are plotted for training as well as validation.

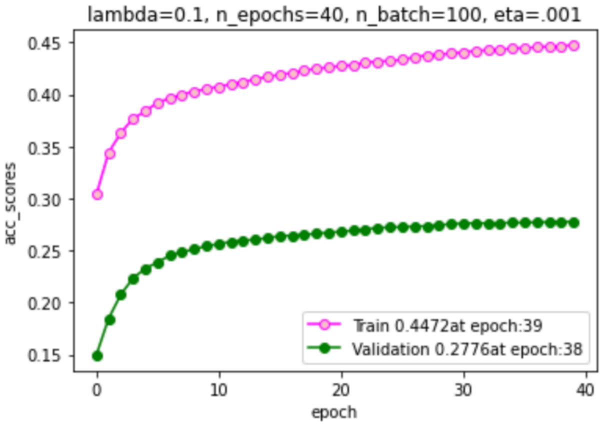


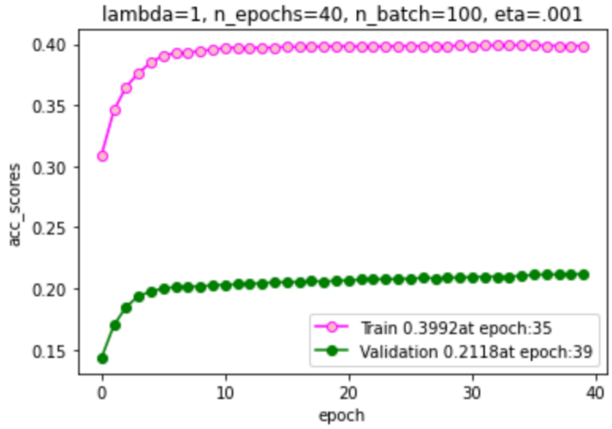


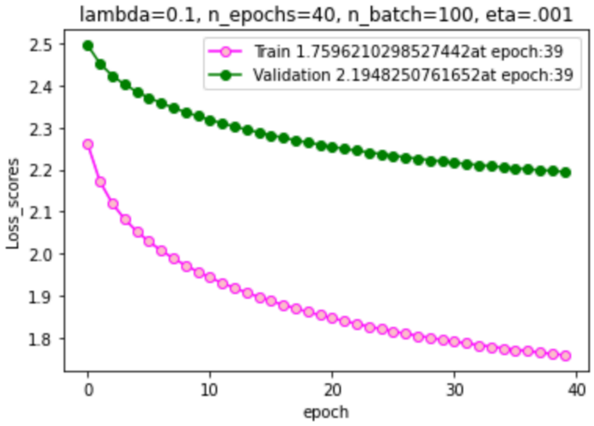


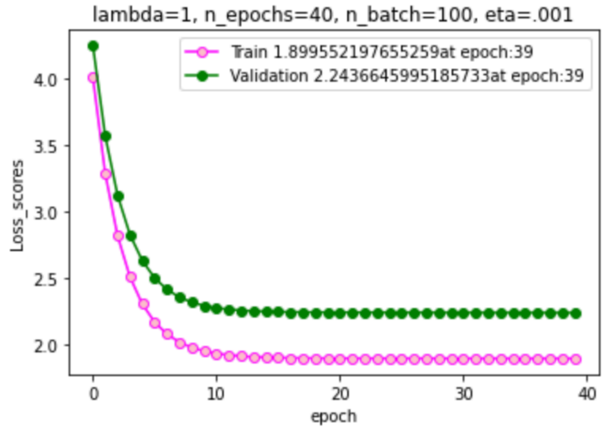


When the learning rate was 0.1 the graph of cost function was unstable. As the regularisation lambda increases , the cost function graph of training and validation converges towards each other more and gap is reduced.

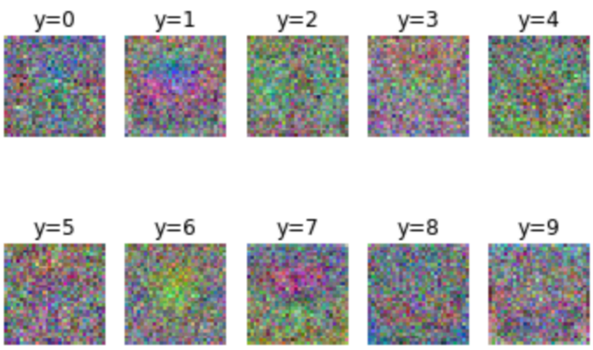




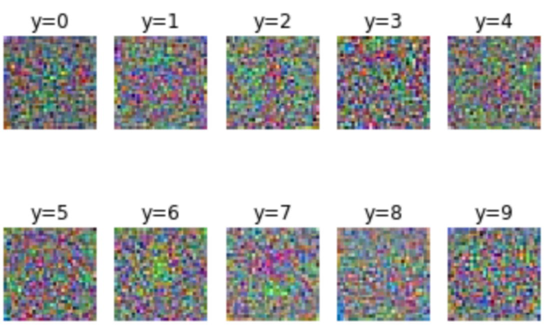




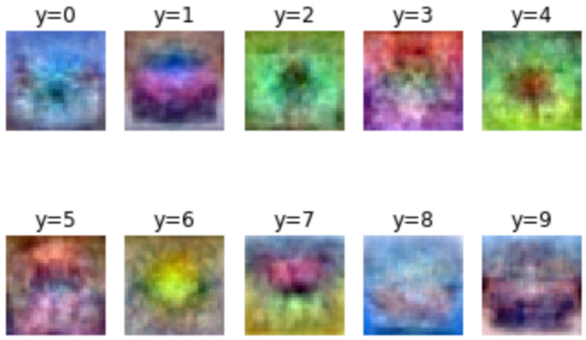
Comparing the weight Matrix



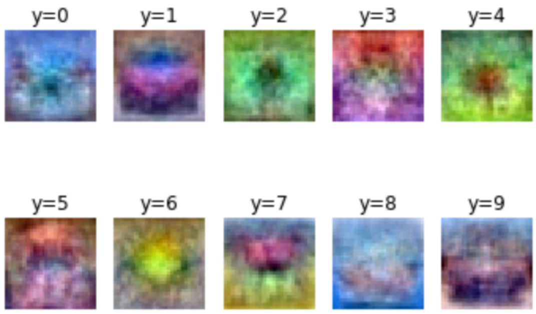
lambda=0, n_epochs=40, n_batch=100, eta=.001



lambda=0, n_epochs=40, n_batch=100, eta=.1



lambda=0.1, n_epochs=40, n_batch=100, eta=.001



lambda=1, n_epochs=40, n_batch=100, eta=.001

From the weight matrix , it is observed that as regularisation lambda increases the weights . And also for some of the images we can visualise the image from a some what clear shape and structure like the automobile.

Test Accuracy:
The following is my final test accuracy results for each parameter setting.

test accuracy at lambda=0, n_epochs=40, n_batch=100, eta=.001 :0.2941
test accuracy at lambda=0, n_epochs=40, n_batch=100, eta=.001 :0.219
test accuracy at lambda=0, n_epochs=40, n_batch=100, eta=.001 :0.2223
test accuracy at lambda=0, n_epochs=40, n_batch=100, eta=.001 :0.2144