ANIRUDH SETH
aniset@kth.se

DD2424:Assignment 1
Deep Learning in Data Science

2020-03-30

Exercise 1

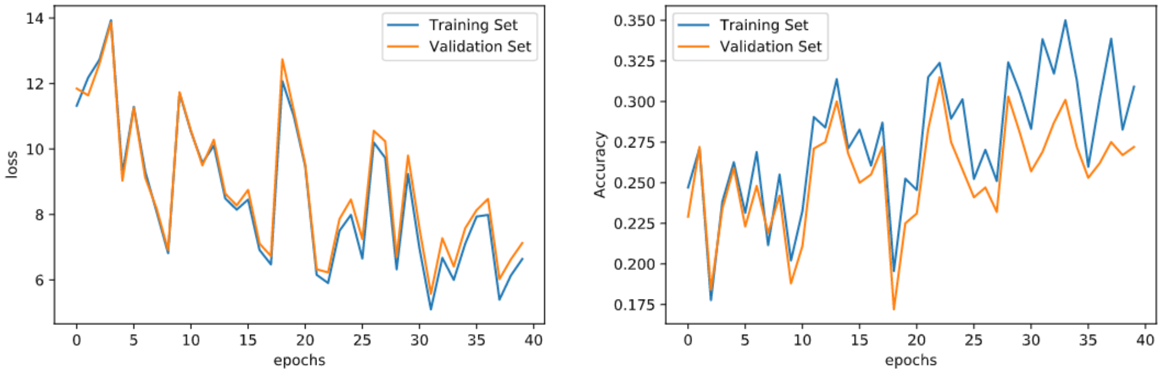
For this exercise , the neural network was trained with cross entropy loss. The analytical gradients of the weights and the bias of the network were compared to the numerical approximations and achieved a mean difference in the range of $\sim e^{-8}$ for the weights and bias.The below results were computed for the first epoch for one batch to validate the gradient computation.
*Mean Difference between Gradients of Weights: -1.2529*e-08, and*
*Mean Difference between Gradients of Bias: -4.4364*e-08.*
The table below summarizes the accuracy scores achieved for all of the mentioned parameter settings.

Training Size : 10,000 Validation Size: 1,000 Test Size: 10,000						
Parameters				**Accuracy**		
No. of Epochs	**ETA**	λ	**Batch Size**	**Training Set**	**Validation Set**	**Test Set**
40	0.1	0	100	0.3091	0.2720	0.2727
40	0.001	0	100	0.3812	0.36	0.3606
40	0.001	0.1	100	0.36922	0.358	0.3518
40	0.001	1	100	0.305	0.306	0.3012

The following graphs show the cost , accuracy along with the learnt weight matrix as an image for all of the above parameters settings.

Effect Of Learning Rate

The learning rate (ETA) is a hyper parameter to our neural network which controls how much the weights and bias of the model are updated in response to the estimated cost/error. A **large learning** can result in learning sub optimal weights too fast and can result in unstable learning because of large update steps. This can be seen in Figure:1 where ETA=0.1. The model achieves a training accuracy of 30.91% and a test accuracy of 27.27%. A **small learning rate** on the other hand can result in a long training process because of small steps and can also get stuck at sub optimal solutions. Figure:2 shows the results where the model ,trained with ETA:0.001 achieves a test accuracy of 36.06%. In optional exercise, we train a model that achieves better accuracy as compared to this setup with a slightly greater ETA.



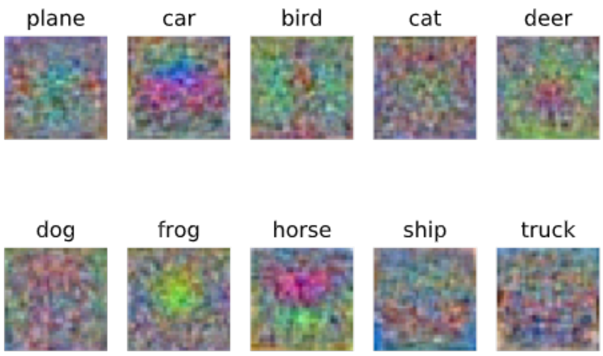


Figure 1: ETA : 0.1 $\lambda = 0$

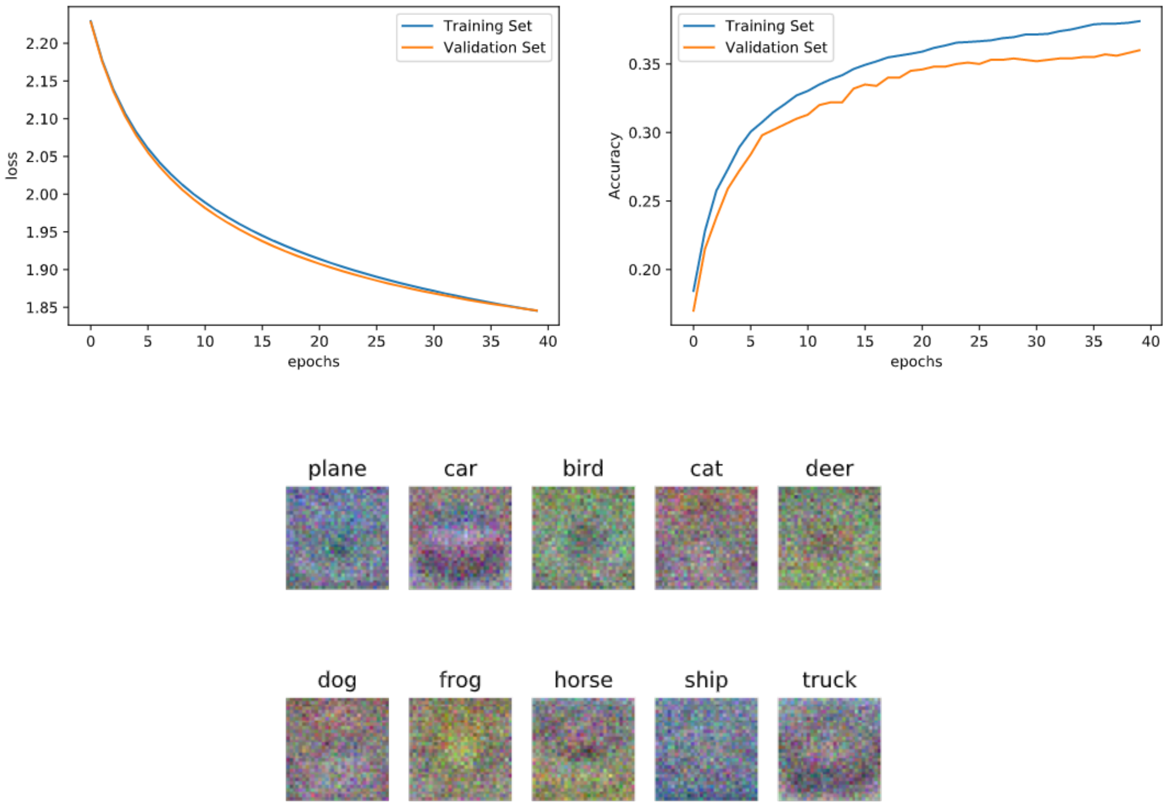


Figure 2: ETA : 0.001 $\lambda = 0$

Effect Of Regularization

In SGD(without regularization of the weights), we search for solutions towards the gradient of the loss function. This can often result in over fitting where the model suffers from poor performance on unseen data and poor generalization. For the model with $\lambda = 0$ and ETA=0.001 the difference between accuracy of training and test set is ~5.403%. With regularization (L2 in our model) we also ensure simplicity of the solution by penalizing high weights(avoid over fitting). In model with $\lambda = 0.1$ and ETA=0.001 we can see that difference between accuracy of training and test set is further reduced to $\sim 4.7\%$ for the same seed. The effect of this regularization can also be seen as smoothing of the learnt weight matrix. With a further high $\lambda = 1$, the weights can be seen to smoothen further (see Figure 4)

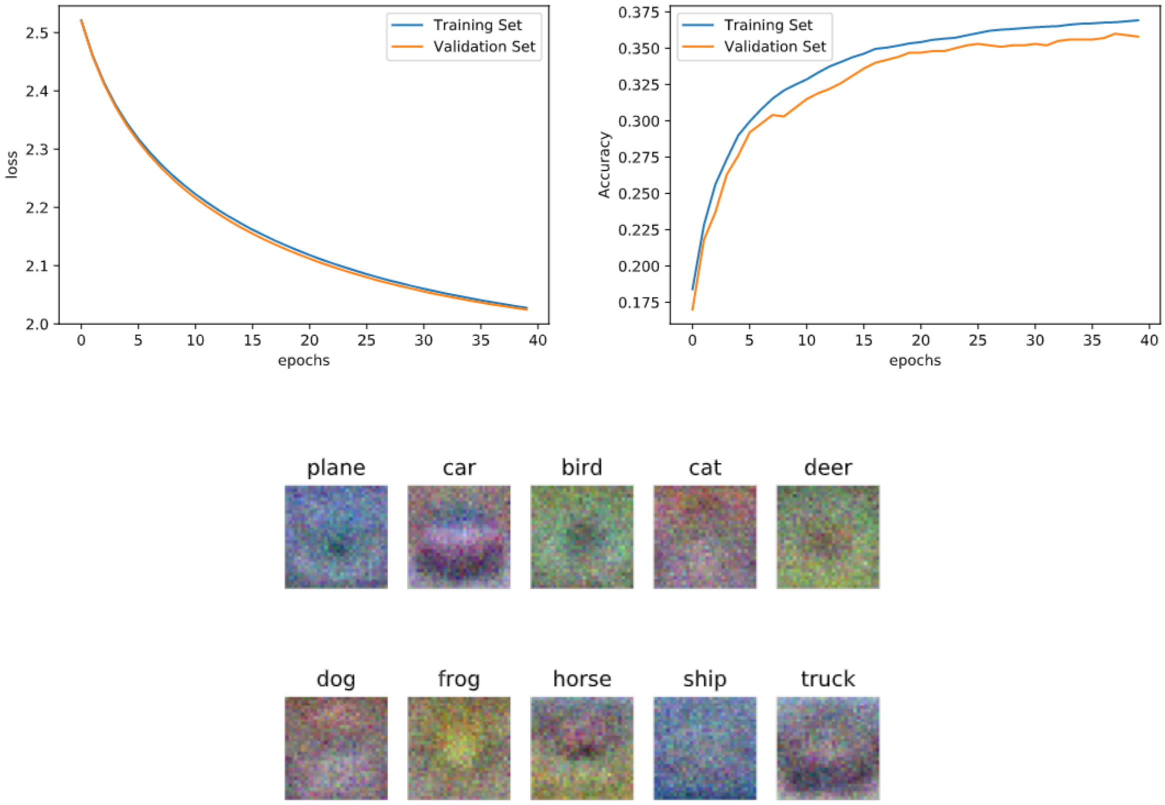


Figure 3: ETA : 0.001 $\lambda = 0.1$

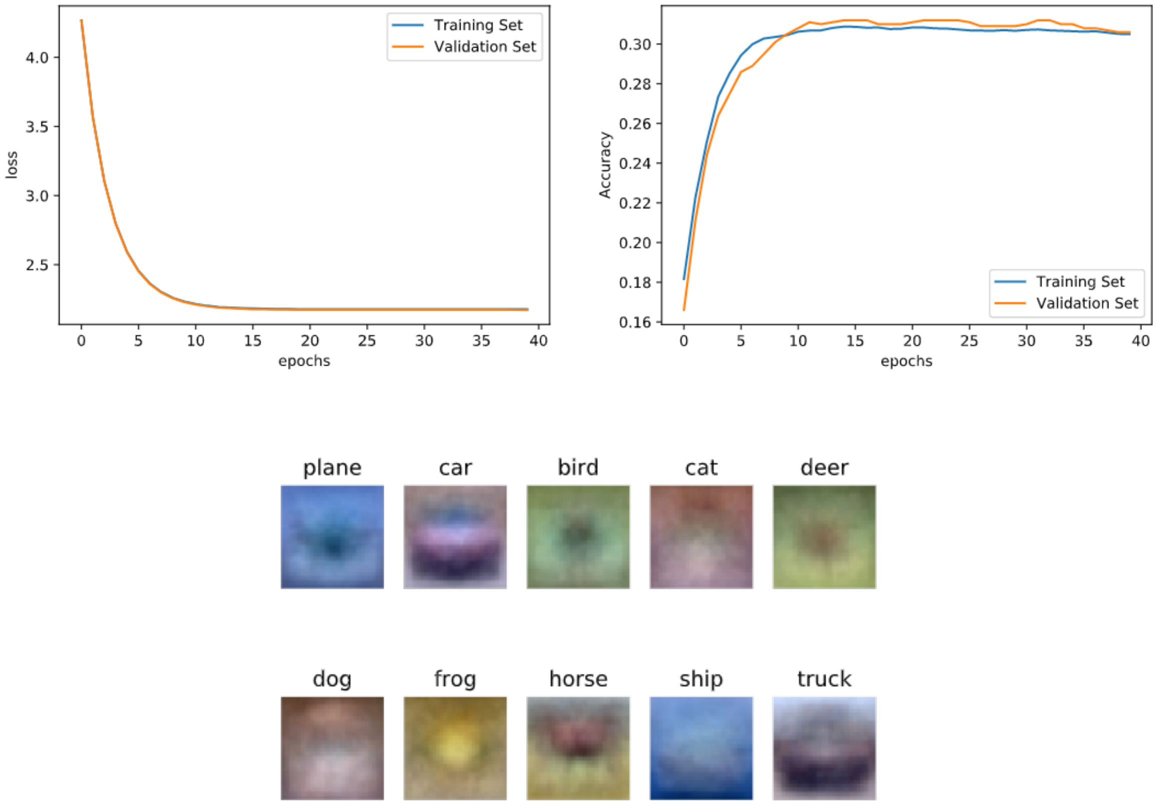


Figure 4: ETA : 0.001 $\lambda = 1$

Exercize 2 (Bonus Points)

The network was additionally trained by minimizing the SVM multi class loss along with the Cross Entropy loss. In order to optimize the performance of the network , several improvements were implemented. The result of each of them is discussed below.In the last section, we combine all improvements to get a better model. For a single data point , SVM loss is :

$$L_i = \sum_{j \neq y_i} \left[\max\left(0, w_j^T x_i - w_{y_i}^T x_i + \Delta\right)\right] \tag{1}$$

, where Δ is the required margin.The derivative wrt to weight w_{y_i} (the correct class) [1]is

$$\nabla_{w_{y_i}} L_i = -\left(\sum_{j \neq y_i} 1\left(w_j^T x_i - w_{y_i}^T x_i + \Delta > 0\right)\right) x_i \tag{2}$$

and the derivative wrt to weight $w_{j \neq y_i}$ [1]is

$$\nabla_{w_j} L_i = 1\left(w_j^T x_i - w_{y_i}^T x_i + \Delta > 0\right) x_i \tag{3}$$

The analytical gradient for SVM loss were also checked with the numerical approximations to verfiy the correctness. For the parameter setting (ETA:0.1 $\lambda = 0$), the below results were computed for the first epoch for one batch.
Mean Difference between Gradients of Weights: 1.0902e-09
Mean Difference between Gradients of Bias: 1.9539e-09

Epoch: 40 ETA: 0.001 Lambda: 0.1 Batch Size: 100						
Enhancement	**Accuracy**					
	Cross Entropy			**SVM Loss**		
	Training Set	Validation Set	Test Set	Training Set	Validation Set	Test Set
Increased Training Size (All Batches)	0.3498	0.355	0.3453	0.3792	0.383	0.3665
Random Shuffling	0.3510	0.356	0.3454	0.3787	0.381	0.3713
ETA Decay	0.3100	0.317	0.3135	0.3665	0.364	0.362
Data Augmentation	0.3492	0.354	0.3462	0.3773	0.383	0.3665

Table 1: Accuracy achieved with cross entropy and SVM loss for some recommended improvements

Increasing Training Data

All 5 batches of the data were used , resulting in training size of 49,000 images , validation size of 1000 images and test size of 10,000 images. Increasing the training size should increase the predictive power of the neural network. It was observed that SVM Loss achieved a 6.13% higher accuracy on the test set.

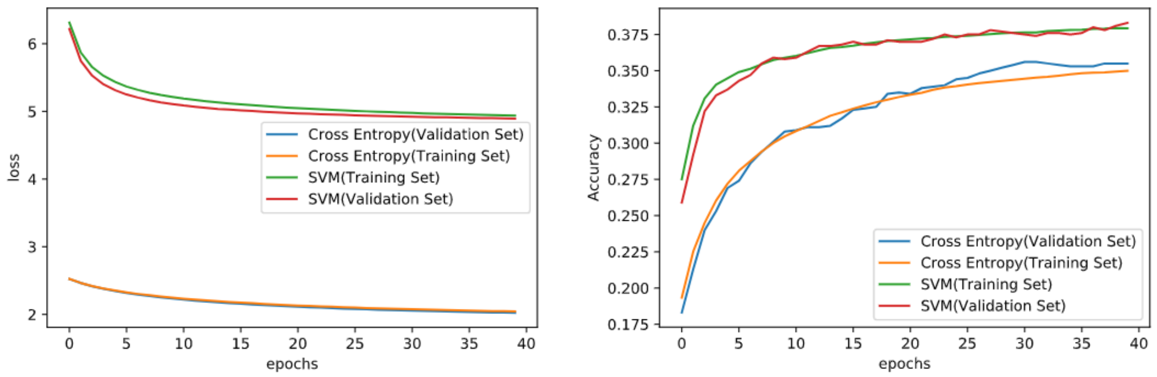


Figure 5: Effect of increasing training data on accuracy and cost

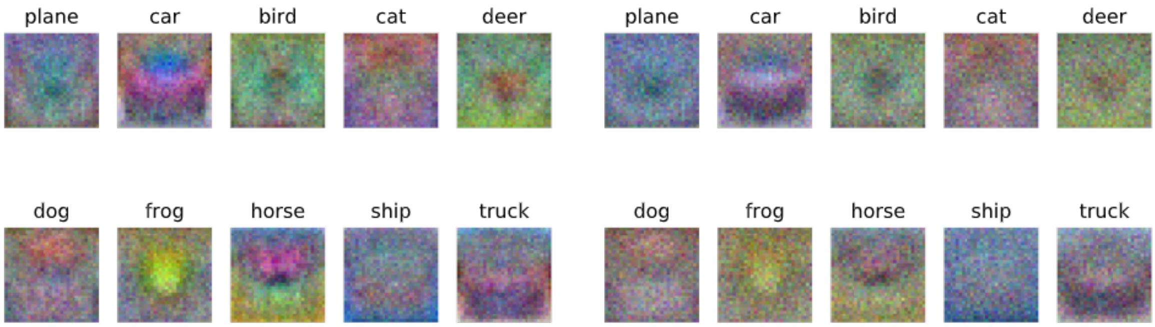


Figure 6: Learnt weight matrix: SVM Multi Class Loss (left) , Cross Entropy Loss(right)

Random Shuffling

Before each epoch the training examples were randomly shuffled.This is done because permuting the training data gives an unbiased estimate of the true gradient, that could improve the accuracy of the model. The accuracy score for this approach were similar to the above case. Not much improvement was observed(a relative difference $\sim .002\%$ for both cross entropy as well as SVM). The accuracy can be seen to fluctuate (Figure:8).

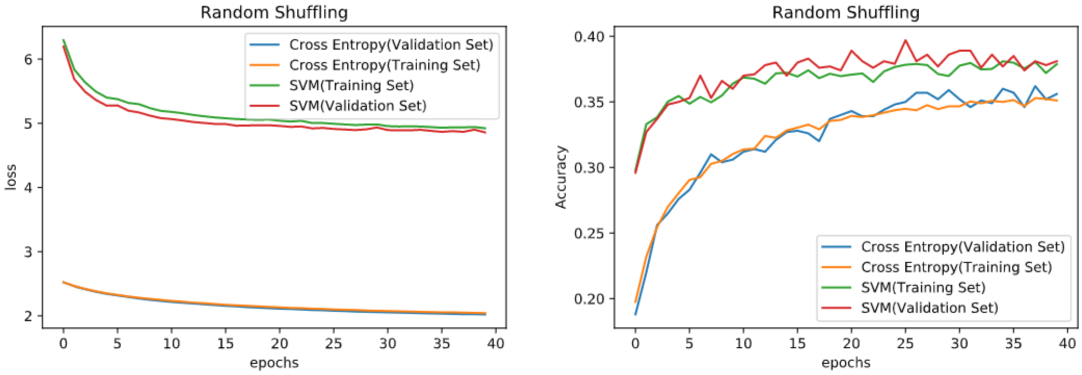


Figure 7: Effect of random shuffling before each epoch on accuracy and cost

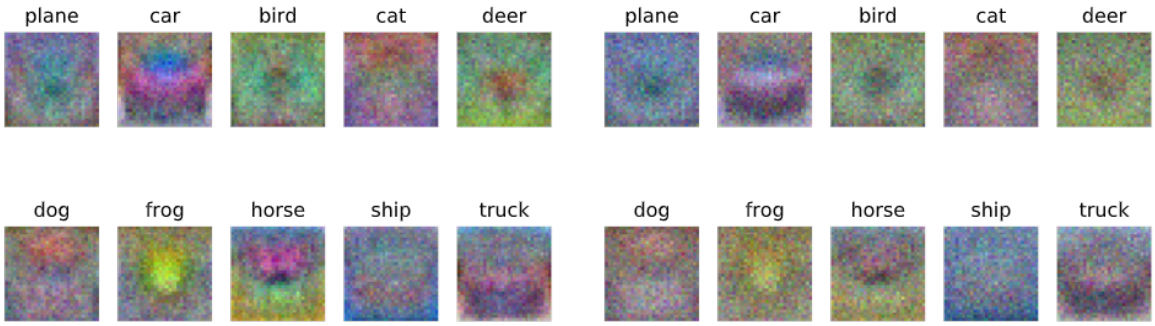


Figure 8: Learnt weight matrix: SVM Multi Class Loss (left) , Cross Entropy Loss(right)

Decaying the learning rate

An initial large learning rate accelerates training or helps the network escape spurious local minima and decaying the learning rate can help the network converge to a local minimum and avoid oscillations. Step Delay was implemented by decaying the ETA by a factor of 0.90 after each epoch . The training cost and accuracy seems to have smoothed (Figure 9) as compared to the above model.This comes at the cost of accuracy which seems to have reduced to 31.35% for cross entropy and 36.2% for SVM loss.

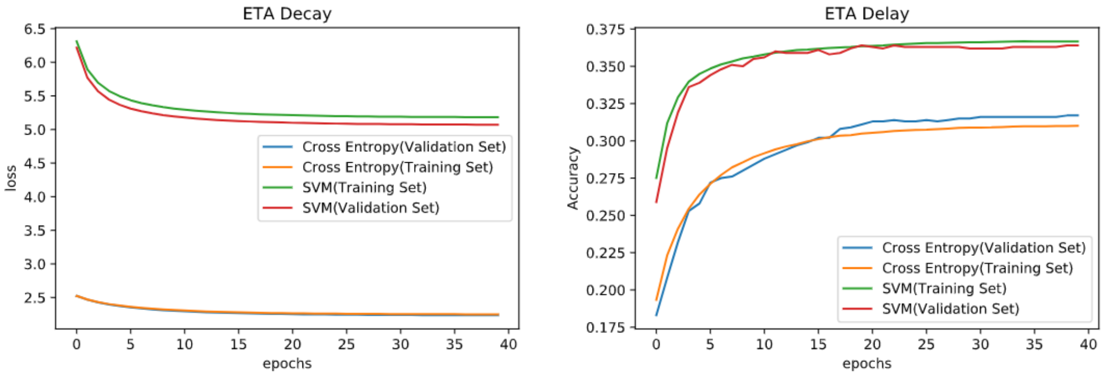


Figure 9: Effect of ETA decay on accuracy and cost

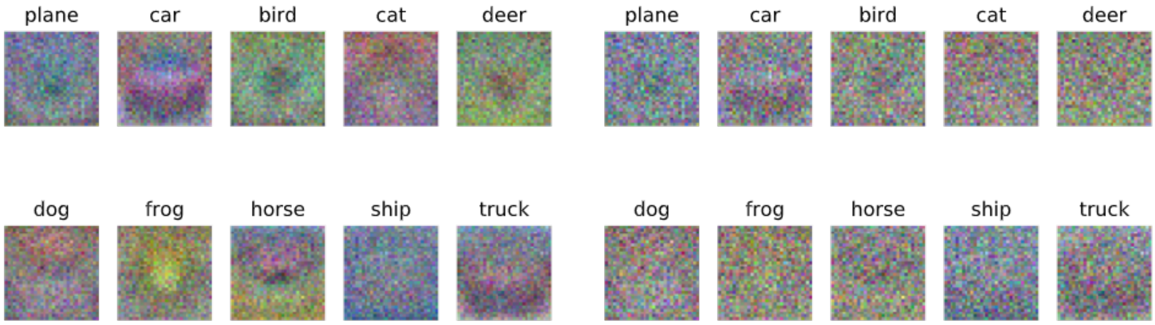


Figure 10: Learnt weight matrix: SVM Multi Class Loss (left) , Cross Entropy Loss(right)

Training Data Augmentation

Each batch dataset was augmented by randomly selecting a subset and addition of white noise (Gaussian).For my model, the batch data was augmented by a factor of 10%. A Gaussian noise $\mathcal{N}(0, 0.0001)$ was used to simulate noise/blurriness in the image. This approach worked the best and improved the accuracy to 34.62% for cross entropy loss and 36.65% for SVM Loss.

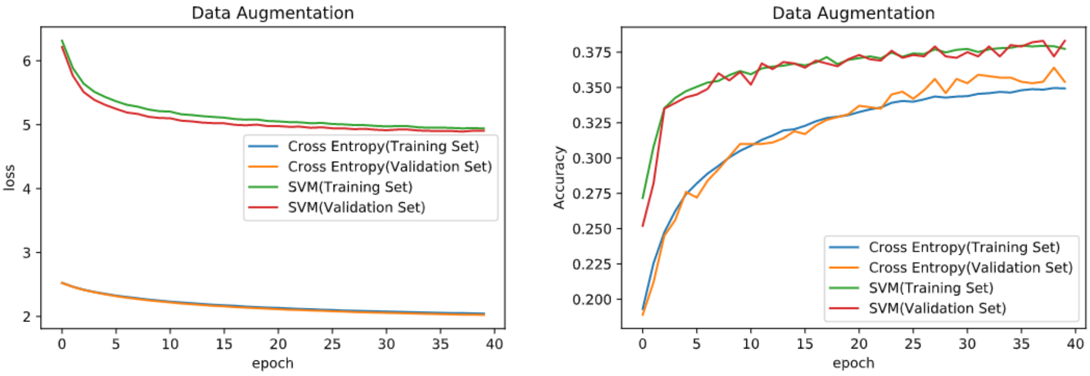


Figure 11: Effect of training data augmentation on accuracy and cost

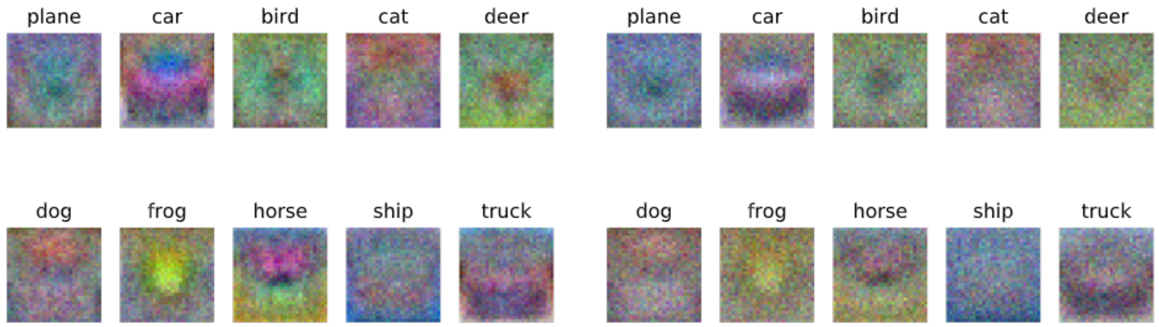


Figure 12: Learnt weight matrix: SVM Multi Class Loss (left) , Cross Entropy Loss(right)

Combining Improvements

Along with the above mentioned improvements, i trained the neural network with a range of hyper parameters to get the optimal settings. The training iterations were also increased along with Xavier Initialization for weights of hidden layer. Below two models achieve a higher training and test accuracy as compared to the models observed so far. Each model was trained with cross entropy loss as well as SVM multi class loss. The results are presented below.

Training Size : 49,000 Validation Size : 1,000 Test Size : 10,000		
HyperParameter	**Model 1**	**Model 2**
Epoch	40	60
ETA	0.007	0.007
Lambda	0.05	0.001
Batch	200	200
ETA Decay	0.9	0.9
Random Shuffling	True	True
Xavier Initilization	False	False
Data Augmentation	0.1	0.1

Table 2: Hyperparameters

Model 1

Accuracy					
Cross Entropy			**SVM Loss**		
Training Set	Validation Set	Test Set	Training Set	Validation Set	Test Set
0.3733	0.367	0.3697	0.3817	0.382	0.3671

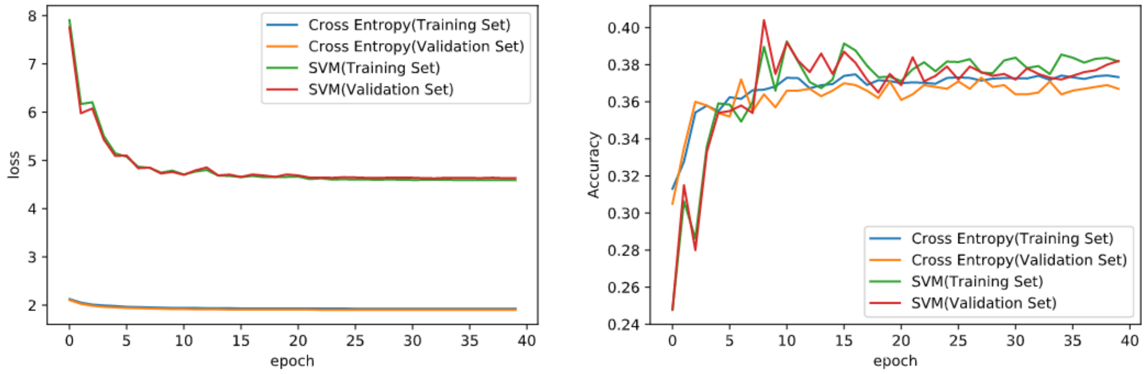


Figure 13: Effect of training data augmentation on accuracy and cost

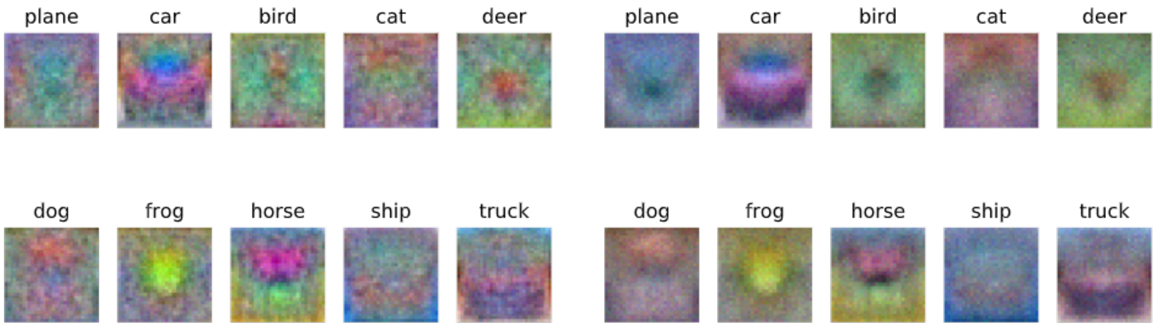


Figure 14: Learnt weight matrix: SVM Multi Class Loss (left) , Cross Entropy Loss(right)

Model 2

Accuracy					
Cross Entropy			SVM Loss		
Training Set	Validation Set	Test Set	Training Set	Validation Set	Test Set
0.3857	0.375	0.3801	0.3903	0.386	0.373

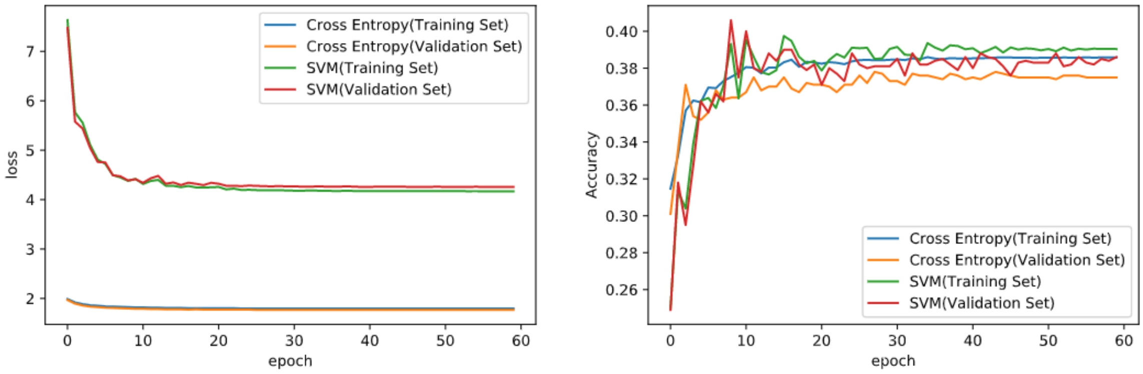


Figure 15: Effect of training data augmentation on accuracy and cost

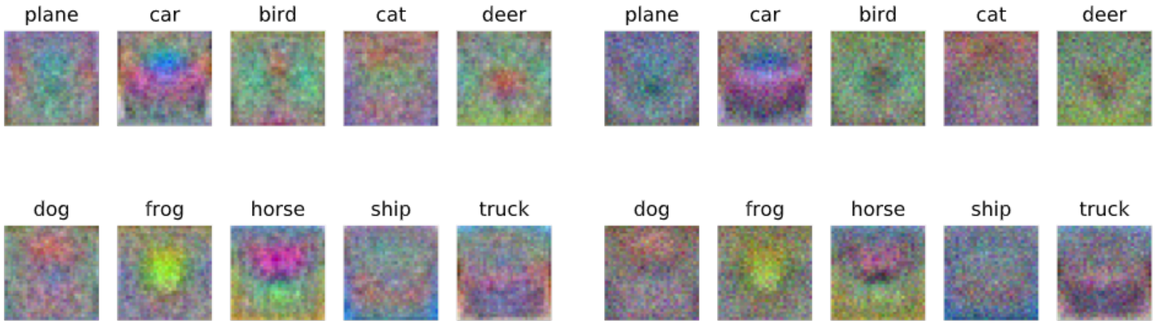


Figure 16: Learnt weight matrix: SVM Multi Class Loss (left) , Cross Entropy Loss(right)

Results

For each of the models above ,an increase in the accuracy on the test set can be observed relative to the earlier models. We also observe the trend that the cost for cross entropy loss is more conservative as compared to the SVM loss. The accuracy on the test set seems to be quite similar for both losses(Figure 13 and Figure:15). Though given the simplicity of the neural network this conclusion would require additional validations. The improvement resulting in largest gain would be the increase in training data and data augmentation. The improvement from random shuffling was not well observed and could be attributed to the simplicity of the model.

References

[1] http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture3.pdf