Deep Learning in Data Science, DD2424

Lab Assignment 1: Single-Layer Classifier on CIFAR-10

Amrita Panesar

March 28, 2020

1 One Layer Classifier

I implemented functions to compute the gradient for cross-entropy loss via both the numerical and analytical methods. I then ran tests to compare the gradient on batches of sample size 1, using a subsest of the dataset (20 samples). I compared the gradients using the relative error, and found that the error on one sample was $8.27 \cdot 10^{-11}$.

I examined the accuracies on the validation and test datasets. Here, as suggested in the Lab instructions, I set the training dataset to data_batch_1, the validation set to data_batch_2, and the test set to test_batch. The results are as follows (eta represents learning rate η, lamda is the regularisation constant λ:

batch_size	eta	lamda	n_epochs	test_accuracy	val_accuracy
100	0.100	0.0	40	0.2877	0.2766
100	0.001	0.0	40	0.3882	0.3864
100	0.001	0.1	40	0.3894	0.3865
100	0.001	1.0	40	0.3759	0.3631

Figure 1: Test accuracy is highest (38.94%) for batch size=100, learning rate η=0.001, regularisation λ=0.1.

We see that no regularisation results in a much lower performance on the test set (28.77%), whilst other parameter settings were fairly close in test and validation performance.

For each of the four parameter settings, the cost and accuracy curves on the validation and training set are shown in Figures 2-5. The weight matrices after training are also shown.

The weights in Figure 5 most resembles the 10 classes, despite having slightly smaller accuracy ($\leq$ 0.01 difference) from the parameters in Figure 4. The regularisation term here helped in smoothing out the weights to make them more interpretable (visually more similar to the original class) by removing extra noise.

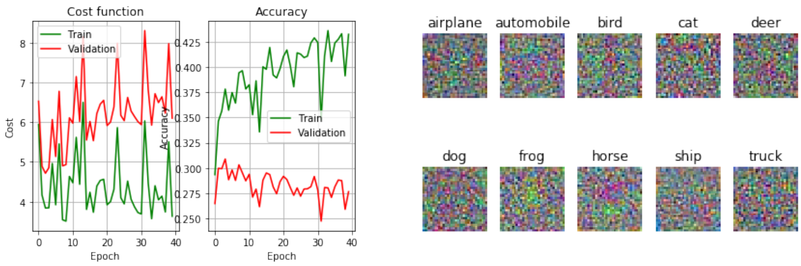

Figure 2: λ=0, n epochs=40, n batch=100, η=0.1

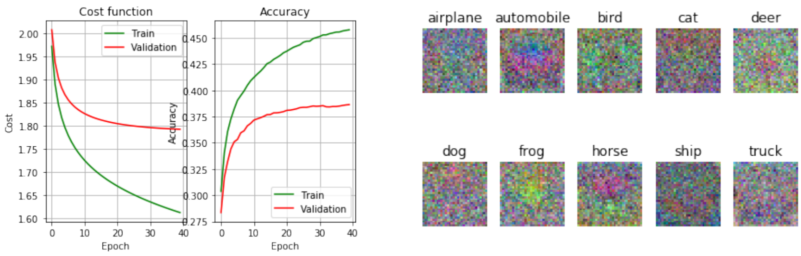

Figure 3: λ=0, n epochs=40, n batch=100, η=.001

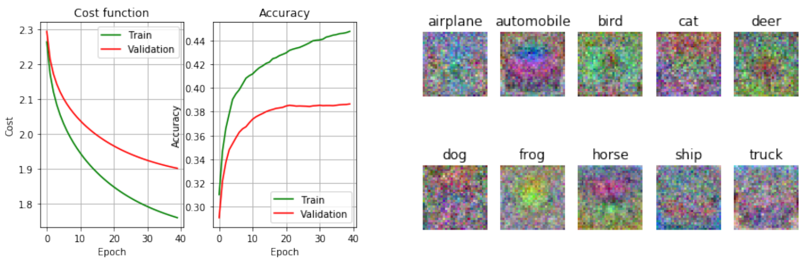

Figure 4: λ=0.1, n epochs=40, n batch=100, η=.001

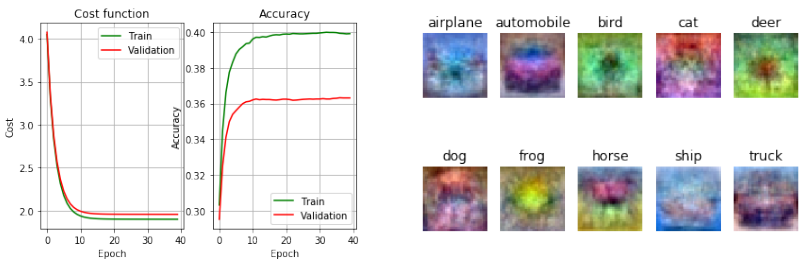

Figure 5: λ=1.0, n epochs=40, n batch=100, η=.001

2 Bonus Points

2.1 Improvements to Classifier

To try to boost performance, I implemented the following improvements:

1. Randomly shuffle the training data at the beginning of each epoch.
2. Train for longer.
3. Randomly shuffle and train for longer.
4. Make use of all of the data to increase size of the training data set.

These experiments were carried out for the best parameters from the previous section: η=0.01, batch size=40. Improvement 1 and 4 were trained for 40 epochs.

Improvement	test_accuracy
Randomly shuffle	0.3953
Train for longer (100 epochs)	0.3933
Randomly shuffle, train for 100 epochs	0.3947
Use more training data	0.4077

Figure 6: Test accuracies for improvements. Biggest boost in performance came from using more training data.

I monitored the validation loss curve when training for longer to ensure that the network was not overfitting, and also tested the validation set accuracy every 10 epochs and monitored this throughout training. The loss (cost) curve is shown below:

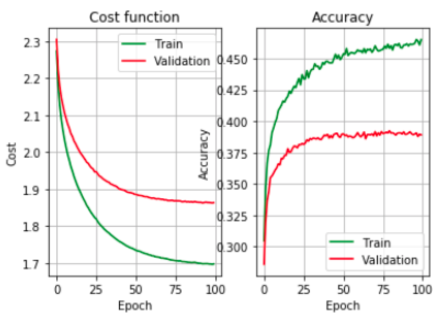

Figure 7: Validation cost does not start to increase during 100 epochs.

2.2 SVM Loss

I then implemented the SVM loss according to the following cost function, where δ is 1:

$$J = \frac{1}{N} \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq y^i}}^{C} max(0, s_{ji} - s_{y^i i} + \delta) + \lambda \sum_{c,d}^{C,D} W_{ij}^2$$

Experiments showed that the best parameters were as follows:

We see that the test accuracy was ¡¿ than the better than the best test accuracy achieved by using cross entropy loss (using data_batch_1 as the training set, data_batch_2 as the validation set, and test set to test_batch.

The costs and weights are shown below:

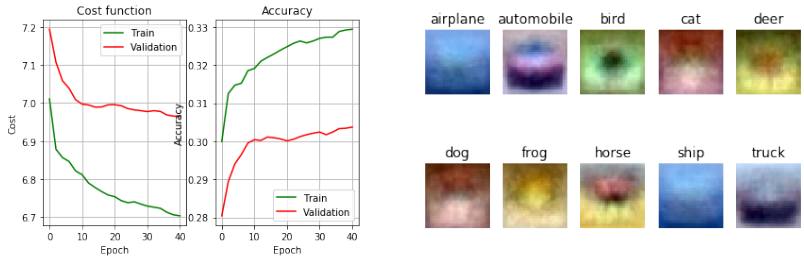

Figure 8: Costs, accuracies and weights for SVM loss, η=0.01, run for 40 epochs

Interestingly, the best accuracy was achieved with no regularisation (row 1 in the table below):

batch_size	delta	eta	lamda	n_epochs	accuracy_svm	accuracy_cross_entropy
100	1	0.100	0.0	40	0.3154	0.2877
100	1	0.001	0.0	40	0.3043	0.3882
100	1	0.001	0.1	40	0.3009	0.3894
100	1	0.001	1.0	40	0.2916	0.3759

Figure 9: Accuracies significantly lower when SVM loss was used compared to when Cross Entropy loss was used.

Overall, we see that using cross entropy with regularisation, running for longer and including more training data achieved the highest performance in the single layer classifier.