DD2424 – Assignment 1

Aloysius Chow

05/04/2023

Analytical gradient computation and tests

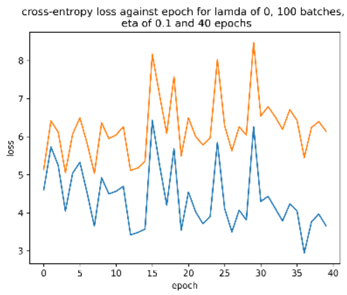
Testing between the analytical and numerical methods of computing gradients was performed to ensure accuracy, with varying number of entries and values of lambda. The results can be found in the table below. The mean relative error was calculated as per the equation described in the assignment.

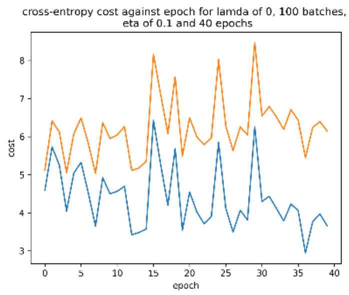
Number of Entries	Lambda	Mean Relative Error for grad wrt W	Mean Relative Error for grad wrt b
1	0	1.7038028267690432e-08	3.323746230789913e-10
10000	0	4.059251924250201e-09	8.690777234490633e-08
10000	0.01	4.625983664684216e-09	8.690777234490633e-08
10000	0.1	4.456274010995312e-09	8.690777234490633e-08

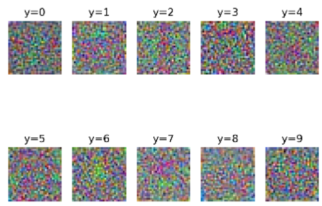
Training the network

(For all graphs below, orange represents training loss/cost and blue represents validation loss/cost)

Parameter set 1: lambda=0, n_epochs=40, n_batch=100, eta=.1







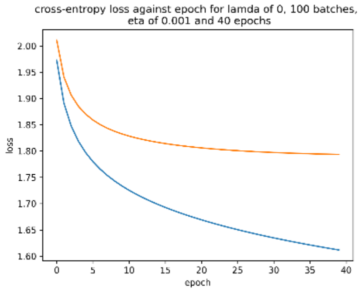
Final accuracy on test set: 0.2881

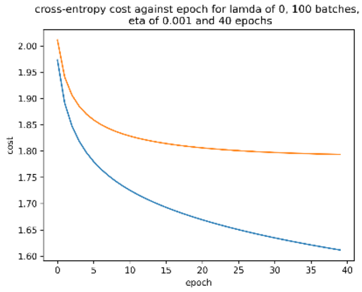
We notice that with a learning rate of 0.1, the model never manages to decrease its loss consistently. Perhaps the learning rate is so large that when we attempt to perform mini batch gradient descent, the local minima are consistently skipped over, leading to the volatile loss that we can see.

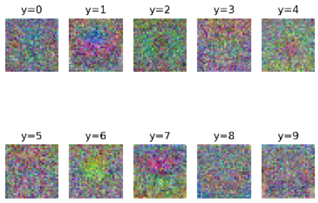
With a lambda (L2 regularisation term) of 0, there is no difference that can be seen between the graphs for loss and cost.

The matrix appears relatively noisy and grainy with no distinct shapes or outliens, suggesting that the templates learnt may be largely contributed to by random noise (such as from intialisation or the unsuccessful gradient descent process).

Parameter set 2: lambda=0, n_epochs=40, n_batch=100, eta=.001







Final accuracy on test set: 0.3893

This performs much better than the previous set of parameters, and is the first set of parameters suggested to us in the assignment file. Here, we see that both training and validation loss decrease consistently through the epochs.

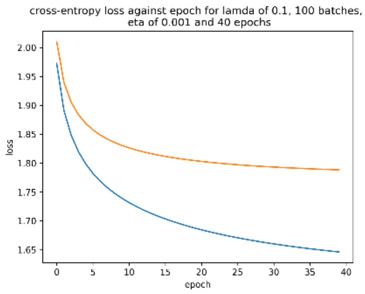
With a lambda (L2 regularisation term) of 0, there is no difference that can be seen between the graphs for loss and cost.

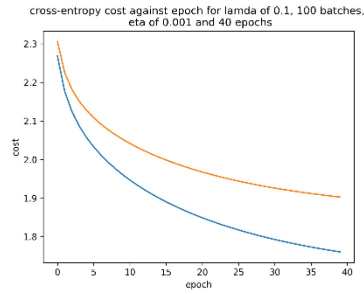
However, the training loss is continuing to decrease as we reach the last epochs, while the validation loss is starting to plateau. This suggests that some degree of overfitting is occuring, and this could be more easily observed if we continued to train the model beyond 40 epochs.

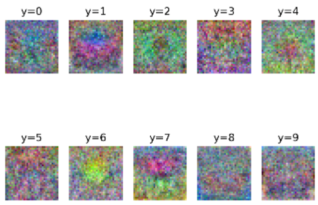
The matrixes display much clearer forms than before, with distinct shapes and regions visible in each template.

The accuracy achieved on the test set is also significantly better than before.

Parameter set 3: lambda=.1, n_epochs=40, n_batch=100, eta=.001







Final accuracy on test set: 0.3909

As with before, both training and validation losses decrease through training.

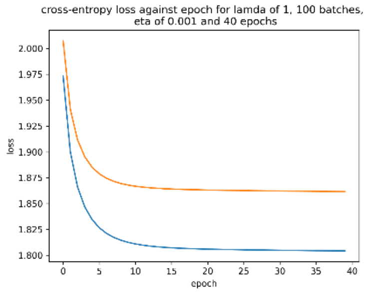
With a lambda (L2 regularisation term) of 0.1, we can see that both training and validation losses are higher than the costs due to the inclusion of the regularisation term.

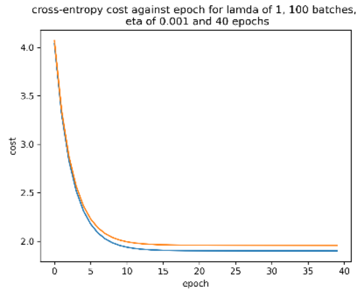
Compared to before, the validation loss is plateauing less at 40 epochs, and there may be less overfitting taking place.

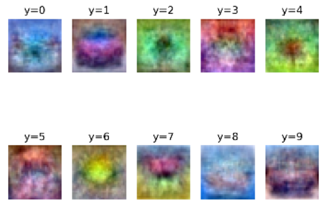
The matrices show greater colour contrast now compared to without regularisation, although I am not sure how this should be interpreted.

The test accuracy improves further to 0.3909, this may have resulted from reduced overfitting.

Parameter set 4: lambda=1, n_epochs=40, n_batch=100, eta=.001







Final accuracy on test set: 0.375

As with before, both training and validation losses decrease through training.

However, all losses and costs plateau rapidly, suggesting that the regularisation term has resulted in a local minima that was quickly found and could not be optimised further.

With a lambda (L2 regularisation term) of 1, we can see that both training and validation losses are much, much higher than the costs due to the inclusion of the regularisation term.

The matrices show the brightest and most distinct colours and shapes of all the parameter sets. We can even start to see shapes resembling the classes we are attempting to classify these images into. Perhaps with a large regularisation term, we are forcing the model to look at the broader, more general features like larger shapes and forms, which allows the model to better generalise.

Conclusion

The results show the importance of having an appropriate learning rate, as one that is too large could lead to failure to find the local minima and result in an inability to reduce loss.

At the same time, it also shows the role of regularisation in ensuring that the model we train is able to pick up on the features we consider important while avoiding overfitting.

Lastly, this assignment also served as an introduction for me on using Numpy to perform matrix calculations for me, and it forced me to better visualise what each matrix means when performing evaluation, forward passes or backward passes.