KTH Royal Institute of Technology
DD2424-VT19-1 Deep Learning in Data Science
Assignment 1

Marina Herrera Sarrias

April 8, 2019

1 Exercise 1

Images corresponding to the following parameter settings:

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

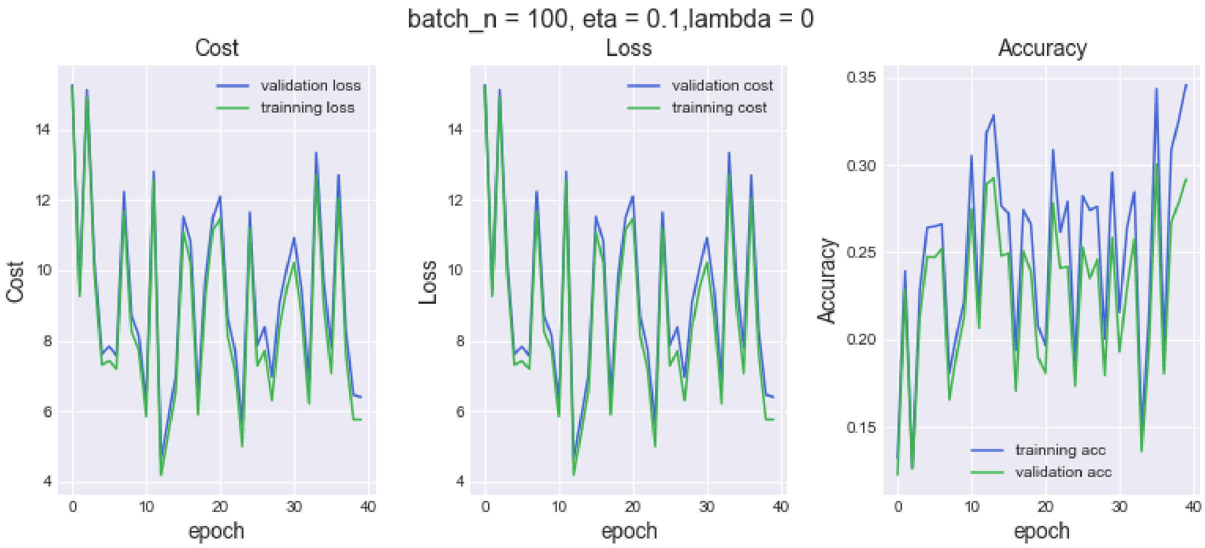

Figure 1: The graph of the training and validation loss computed after every epoch. The network was trained with the following parameter settings: `n_batch`= 100 `eta`= 0.1, `n_epochs`= 40 and `lambda`= 0.

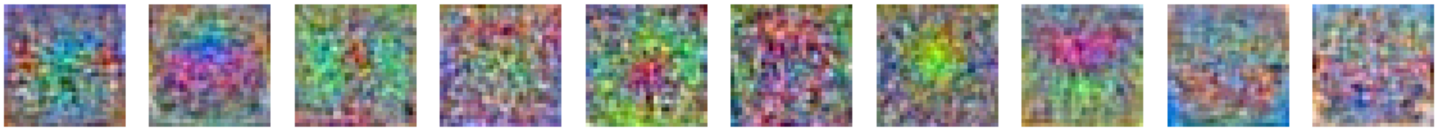

Figure 2: The graph of the training and validation loss computed after every epoch. The network was trained with the following parameter settings: `n_batch`= 100 `eta`= 0.1, `n_epochs`= 40 and `lambda`= 0.

1.2 lambda=0, n_epochs=40, n_batch=100, eta=.01

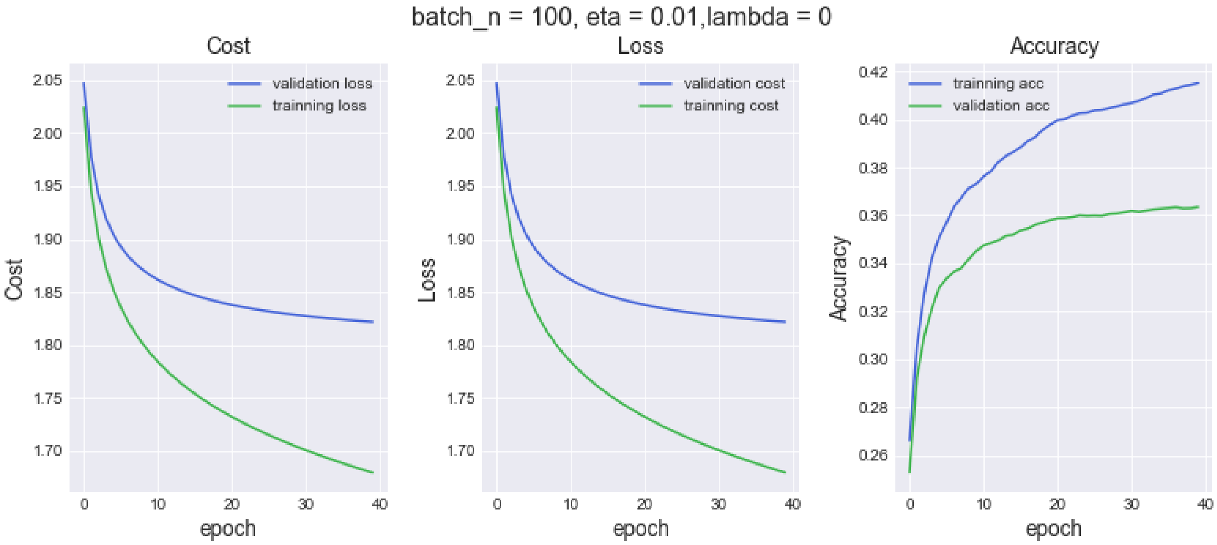

Figure 3: The graph of the training and validation loss computed after every epoch. The network was trained with the following parameter settings: `n_batch`= 100 `eta`= .01, `n_epochs`= 40 and `lambda`= 0.

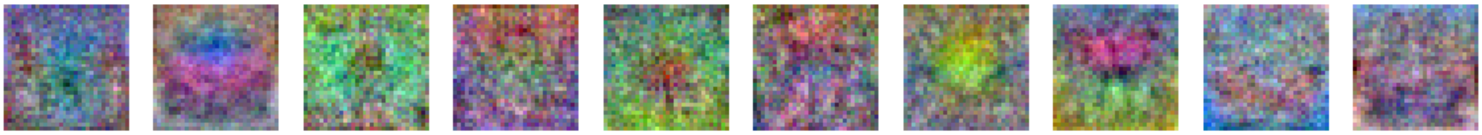

Figure 4: The graph of the training and validation loss computed after every epoch. The network was trained with the following parameter settings: `n_batch`= 100 `eta`= .01, `n_epochs`= 40 and `lambda`= 0.

1.3 `lambda=.1, n_epochs=40, n_batch=100, eta=.01`

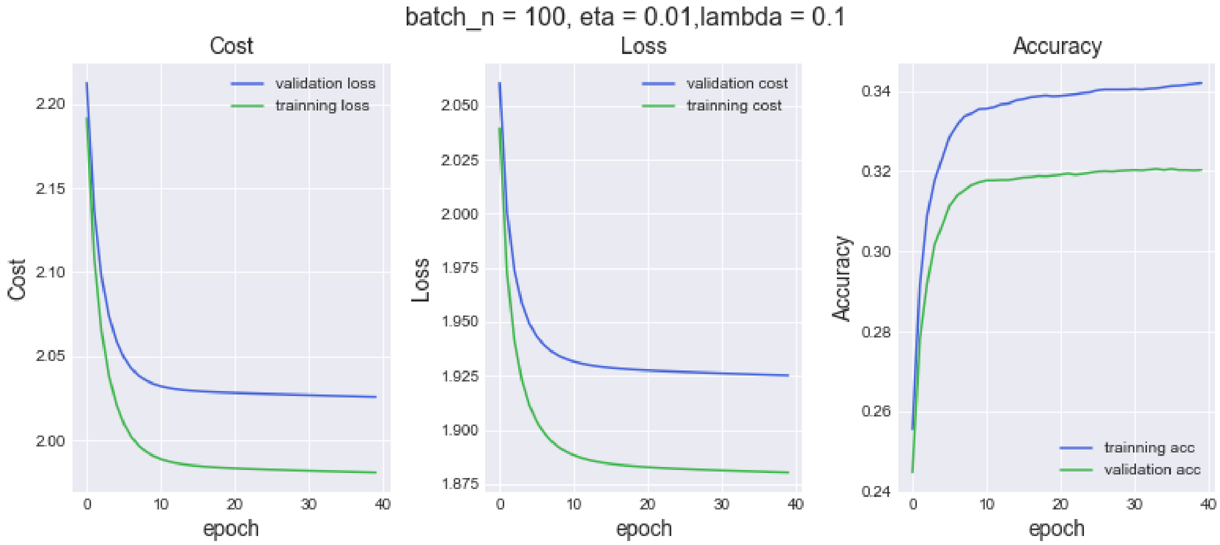

Figure 5: The graph of the training and validation loss computed after every epoch. The network was trained with the following parameter settings: `n_batch`= 100 `eta`= .01, `n_epochs`= 40 and `lambda`= 0.1.

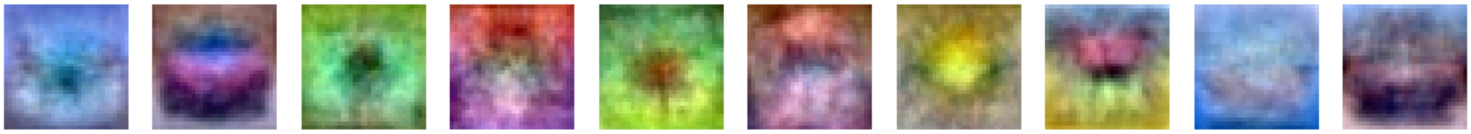

Figure 6: The learnt W matrix visualized as class template images. The network was trained with the following parameter settings: `n_batch`= 100, `eta`= .01, `n_epochs`= 40 and `lambda`= 0.1.

1.4 `lambda=1, n_epochs=40, n_batch=100, eta=.01`

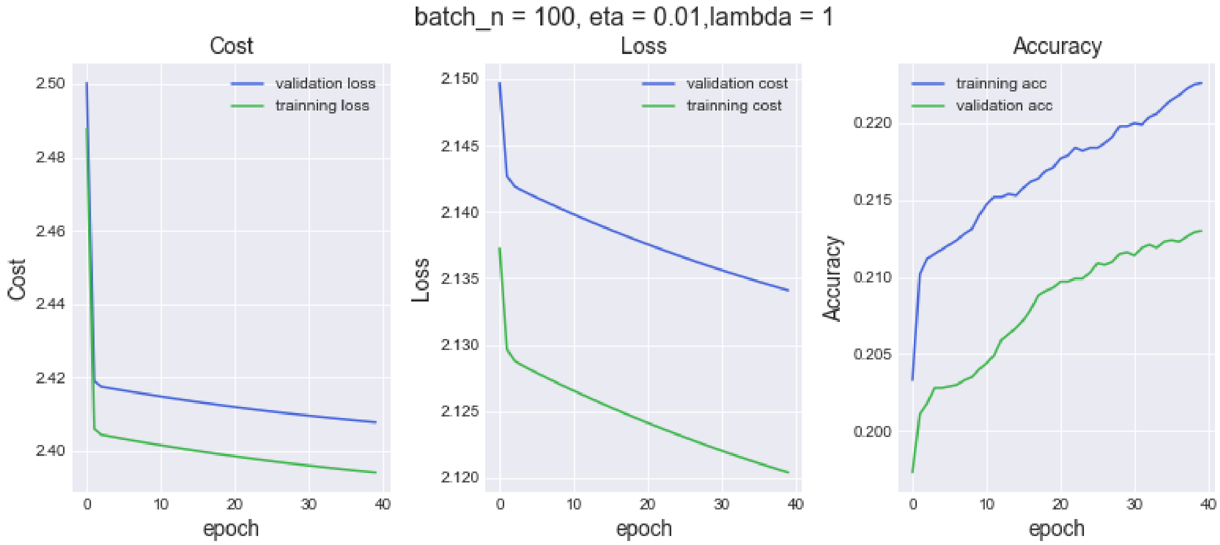

Figure 7: The graph of the training and validation loss computed after every epoch. The network was trained with the following parameter settings: `n_batch`= 100 `eta`= .01, `n_epochs`= 40 and `lambda`= 1.

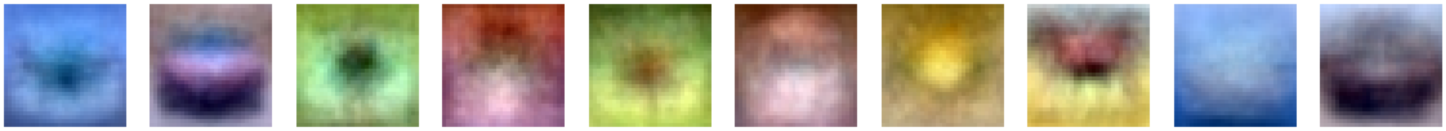

Figure 8: The learnt W matrix visualized as class template images. The network was trained with the following parameter settings: `n_batch`= 100, `eta`= .01, `n_epochs`= 40 and `lambda`= 1.

After training the networks for the different parameter settings using mini-batch gradient descent, we can conclude that the training with minimum validation cost and maximum accuracy is given by the parameter settings in section 1.2, achieving a maximum accuracy of 36.51%.

In none of the cases above, adding regularization improved the accuracy of the network. As the value of `lambda` increased from 0.1 (Figure.5) to 1 (Figure.7), the network's accuracy decreased. Furthermore, we can also say that adding the term does not improve the minimization of the cost function, in both cases the loss decreased more than the actual cost, implying that adding regularization does not contribute to the cost minimization of the model.

Finally, the best performance was achieved on the setting with lower learning rate (`eta`= .01). When the value of `eta` is set too high, such as in Figure (1) the learning rate causes instability and fails to converge. While on the other hand, when the value of `eta` is set too low it will take longer to converge, if it ever does. Furthermore, in both of the cases mentioned before, a ”bad” value of `eta` leads to an increase on the training error.

The learning rate controls how quickly or slowly our network learns, therefore, it is one of the most important hyper-parameters to tun in our model. There is not a possible way to calculate the best learning rate for a given model and data-set. Therefore we will need to find it. A grid search can help us to find a fixed learning rate while a method like applying a learning rate decay can make our model to learn faster. I will discuss both techniques on the bonus assignment.