# Deep Learning in Data Science Assignment 3

Harald Stiff

April 29, 2018

2i) The gradients were checked by comparing the values with the numerical ones from *ComputeGradsNum.m*. Relative errors

$$\mathcal{E} = \frac{|g_a - g_n|}{\max(1e-9, |g_a| + |g_n|)}$$

were computed for random elements of all of the weight matrices and biases. $\mathcal{E}$ was always less than e-5 which is the proof for that the gradients were computed correctly.

2ii) The introduction of batch normalization improved the training a lot. In figure 1 the validation and training losses are plotted for a three layered network using with and without batch normalization respectively. The network was not learning anything without normalization which can be spotted from the relatively constant cost in figure 1a. The networks were trained using $\eta = 0.3$, $\lambda = 10^{-5}$ and decay rate $= 0.8$.

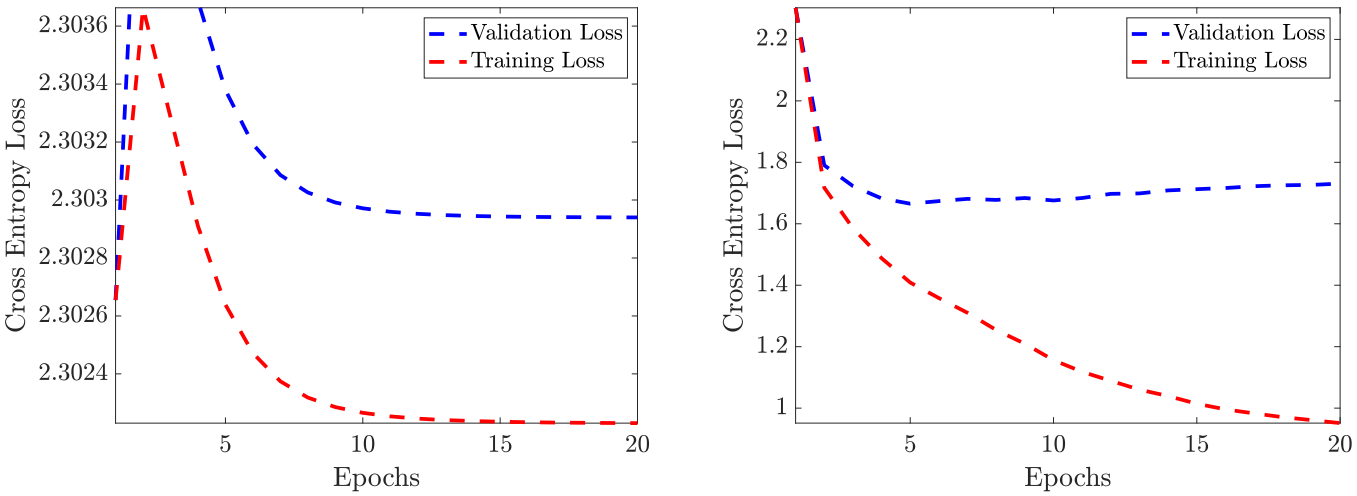


Figure 1: Training and validation loss with (right) and without (left) batch normalization layers.

2iii) The fine search was performed with values of in the range of

$$0.08 \leq \eta \leq 0.2, \qquad 10^{-6} \leq \lambda \leq 5 \times 10^{-4}.$$

The test was done by training 25 networks with 10 epochs and ranking the networks according to what parameters got the best accuracy on the test set. In table 1 the three best performing networks are shown. The accuracy column shows the accuracy of the test set after 10 epochs of training.

| $\eta$ | $\lambda$ | Accuracy |
|---|---|---|
| 0.184541 | 0.000003 | 0.454500 |
| 0.171172 | 0.000016 | 0.453300 |
| 0.096206 | 0.000005 | 0.452000 |

Table 1: Best performing networks.

2iv) In figures 2-4, there are loss plots with learning rates $\eta = 0.01$, 0.1 and 1 respectively. The figures to the left show networks without batch normalization and the networks to the right show networks with batch normalization.

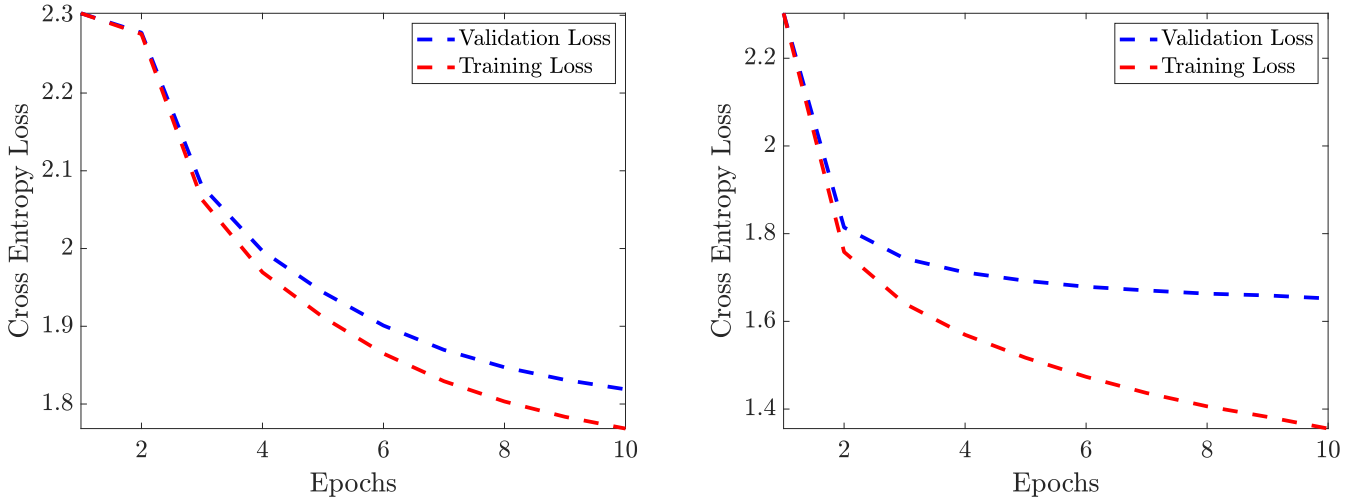


Figure 2: Training and validation loss with $\eta = 0.01$.

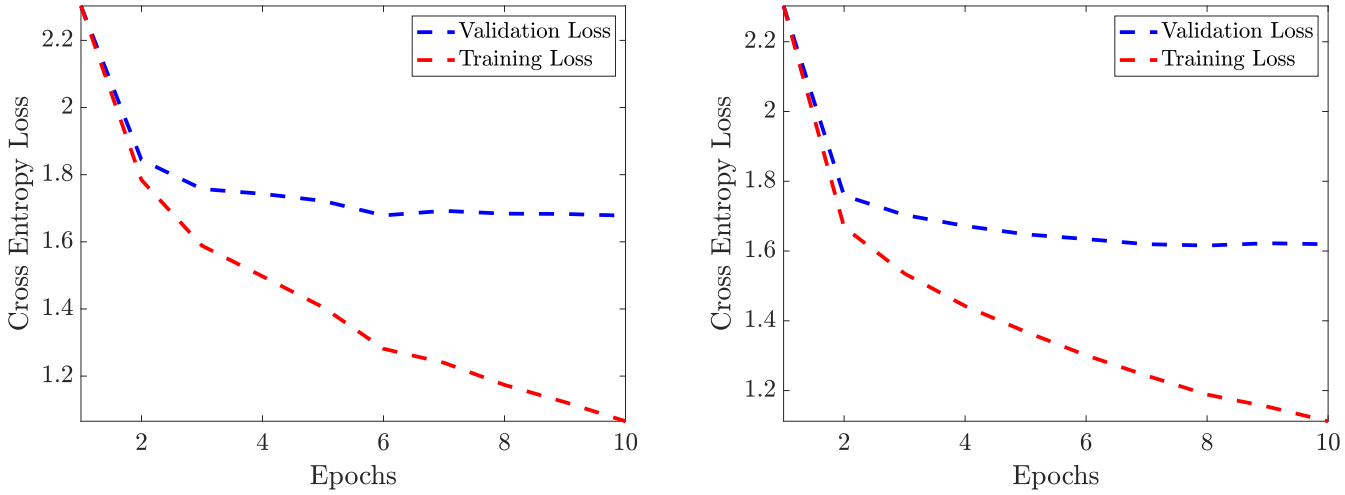


Figure 3: Training and validation loss with $\eta = 0.1$.

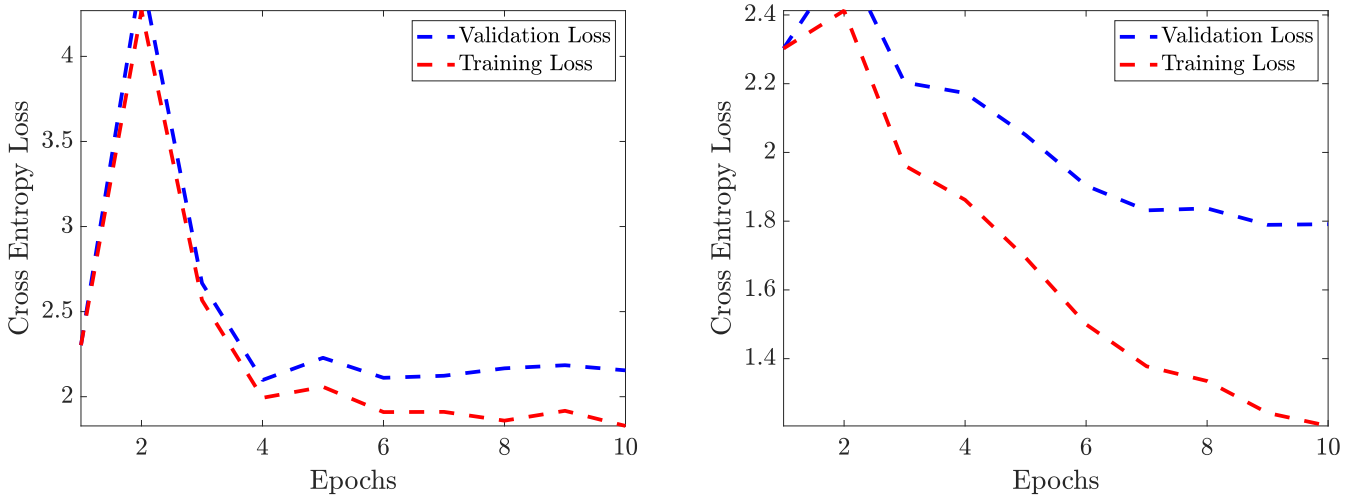


Figure 4: Training and validation loss with $\eta = 1$.

The conclusion that can be drawn is that batch normalization both stabilizes training and can handle very large learning rates without becoming unstable as it would without the normalization layers. Also, the networks converges much faster which reduces the amount of time needed for training.