DL Assignment 2 Bonus

Meinan Gou meinang@kth.se

1 Ex 5.1

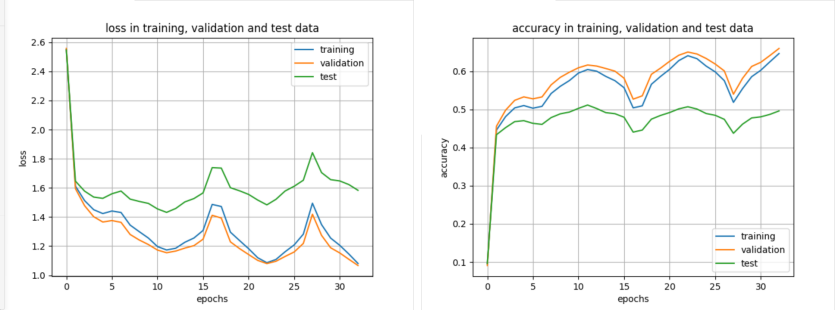
1.1 A more exhaustive random search

From the mandatory part of Assignment 2, I found that λ=3.38e-4 results the highest test accuracy. Thus, I did a more exhaustive random search based on the value. I set n_s=2700 and n_cycles=3.

λ	Validation Accuracy
0.0003909904909652946	0.6528
0.0003167896735290314	**0.6589**
0.00034953116452997166	0.6539
0.0003517326052602803	0.6586
0.00040364784826760577	0.6567
0.0004071317783831084	0.6532
0.000298611207224525	0.6581
0.0003160877972902649	0.6553
0.0003279849893484053	0.6563
0.0003177744098468442	0.6557

Table 1: More exhaustive random search of λ

Table 1 shows the found λ and their corresponding accuracy. The bold values are the λ which has the highest accuracy. Compared to the accuracy in mandatory part (0.5129), then validation accuracy here is 0.6589, which improves about 15% due to better λ and larger n_s.

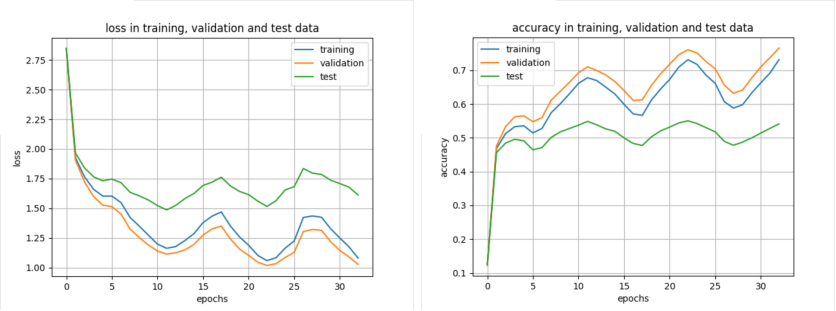


(a) loss (b) Accuracy

Figure 1: Best $\lambda \approx 0.0003168$, best test accuracy=0.5116

1.2 More hidden nodes

I increased the number of hidden nodes from 50 to 200 and set $\lambda = 0.002$. The best test accuracy improves a lot.

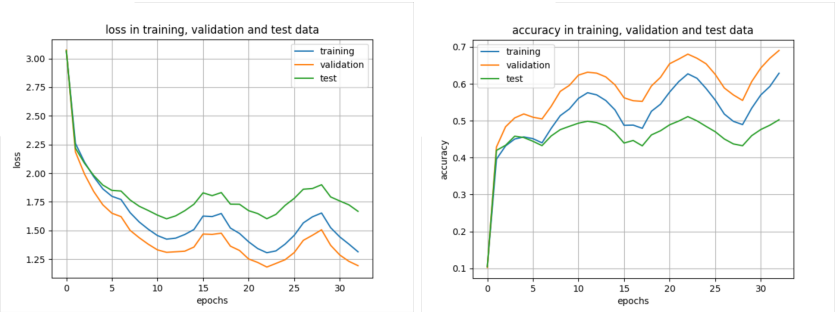


(a) loss (b) Accuracy

Figure 2: # hidden=200, $\lambda = 0.002$, best test accuracy=0.5503

1.3 Dropout

I set the number of hidden nodes to 300, λ to 0.002 and p (the probability of keeping an activation active) to 0.9. These were implemented in the function drop_out_data(). Moreover, I set n_s=2700 and n_cycles=3.



(a) loss (b) Accuracy

Figure 3: # hidden=300, $\lambda = 0.002$, p=0.9, best test accuracy=0.5113

1.4 Summary

From my experiments, all of the three methods can improve the test accuracy, among which increasing the number of hidden nodes has the most effect. Choosing the appropriate m can improve the accuracy by 5% and more.

2 Ex 5.2

According to the instructions on the paper by Smith, I defined a function find_eta() to return the eta array. I set base_lr=1e-3, max_lr=1e-1, and stepsize=5e-3, and the code

$$eta\_array = np.arange(base\_lr, max\_lr, stepsize)$$

to generate the sequence.

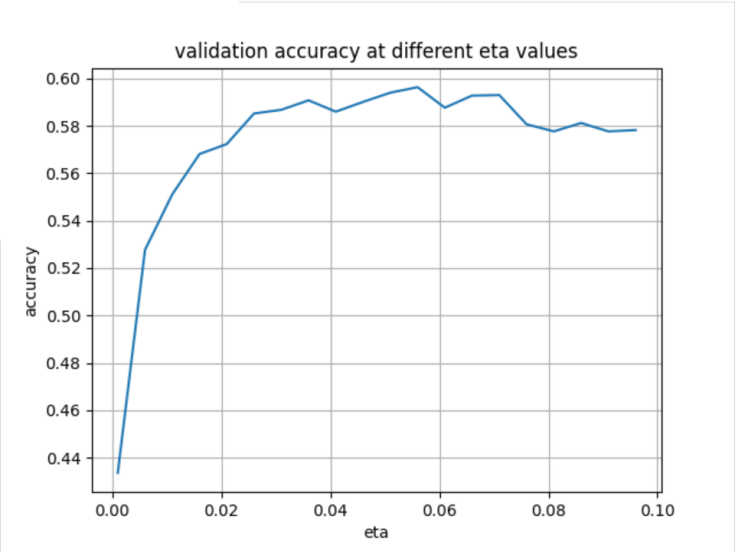
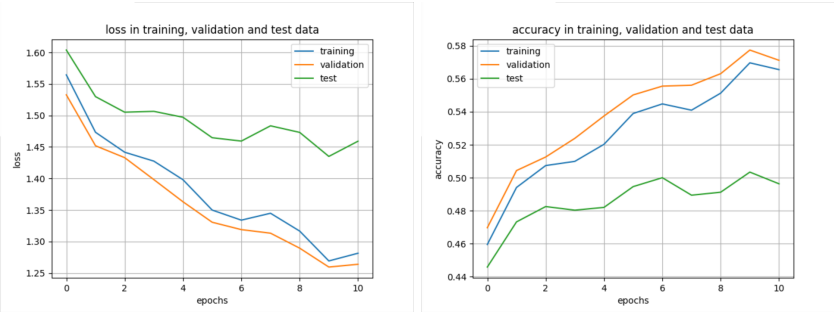


Figure 4: # hidden=50, n_epochs=10, $\lambda = 0.000338$, n_s=900, n_cycles=1, best validation accuracy=0.5963

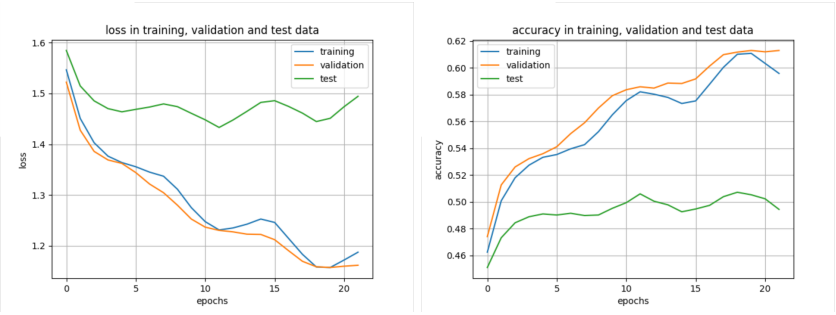
Since the accuracy starts to fall at eta=0.037, I set eta_max = 0.037, and eta_min = base_lr = 1e-3.

I used these parameters to training the final network with $\lambda = 0.000338$, n_cycles=3, n_s=900 and 1800 respectively.



(a) loss (b) Accuracy

Figure 5: # hidden=50, $\lambda = 0.000338$, best test accuracy=0.5034



(a) loss (b) Accuracy

Figure 6: # hidden=50, $\lambda = 0.000338$, best test accuracy=0.5071

From my experiments, compared to the last part of mandatory, using eta found by me does not really improve the final test accuracy. The plots of accuracy and costs are similar.