DD2424 - Assignment 2 Bonus

Timo Nicolai

April 16, 2019

1 Optimization

I skipped improvement (e) because it seemingly didn't work out at all in the last assignment. I tried out all other improvements in the order that seemed most sensible to me:

First I tested if increasing the number of nodes in the hidden layer improves validation set performance and if so, how many nodes are necessary until this effect saturates or even reverses. I performed random search **afterwards** because the optimal hyperparameters most likely vary strongly between networks with hidden layers of different sizes. I then checked whether applying dropout on top of L2 regularization could further improve generalization performance. Finally I constructed an ensemble of networks using the described method and the best training setup determined so far. I'll describe each of these steps in more detail below. Overall, increasing the size of the hidden layer was the only measure that significantly improved validation set performance (and the likely cause for the jump in test set performance).

Initially I reused the hyperparameters used to train the final network in the non-bonus part of this assignment and used 45.000 training and 5000 validation samples. As a baseline, Figure 1 shows the validation set confusion matrix for the final network from the non-bonus section of this assignment.

1.1 More Hidden Nodes

Figure 2 shows training and validation accuracy learning curves for networks with hidden layers of different sizes. Increasing the number of hidden nodes from 50 to 100 and from 100 to 200 causes a large improvement in peak validation accuracy while a further increase to 400 hidden nodes already only results in a very small performance increase. I chose to go with 400 hidden nodes. It is possible that adding even more nodes could improve performance even further if strong regularization is added but I feel like the model is already complex enough and I don't want to expend too much computing power finding the perfect combination of hidden nodes and regularization. The 400 hidden node network achieves a validation set accuracy of 57.1%.

1.2 Random Search

Keeping the size of the hidden layer fixed, I performed random search over λ (from $1e-5$ to $1e-2$) and the cycle length (from 200 to 2000). I think its unlikely that the number of cycles is going to have a large influence on the networks performance (and if it does we can always train longer and use early stopping or use the best intermediate parameters). Changing the batch size might have been insightful but I did not want to search over more than two parameters (which would have required more samples to get a clear picture of each parameters influence on the network performance).

Figure 3 shows a rough interpolation of the validation accuracy "landscape" defined by 30 random samples. There's no clear pattern here, nevertheless I tried to "home in" on the most promising area with a further fine-grained search over $\lambda \in [5e-4, 2e-3]$ and $\eta_{ss} \in [1000, 1250]$. Figure 4 shows the results which are again disappointing, the best found parameter combination achieves only 0.2% better validation set performance than the "pre-search" network. This is unlikely to have a meaningful effect on the test set performance so I have decided against modifying the hyperparameters based on these search results.

1.3 Dropout

Adding dropout decreases validation set performance. I implemented inverted dropout and trained with dropout probability 0.4, 0.5 and 0.6, achieving validation set accuracies of about 52.06%, 49.76% and 49.02%. On this basis I decided against using dropout.

1.4 Ensemble

Finally, Figure 5 shows the points at which I sampled the parameters for a ten model ensemble classifier (this time training on 49.000 training samples) Figurer 6 shows the final test set performance achieved by the resulting network ensemble, an increase of in classification accuracy of round 4.5% could be achieved.

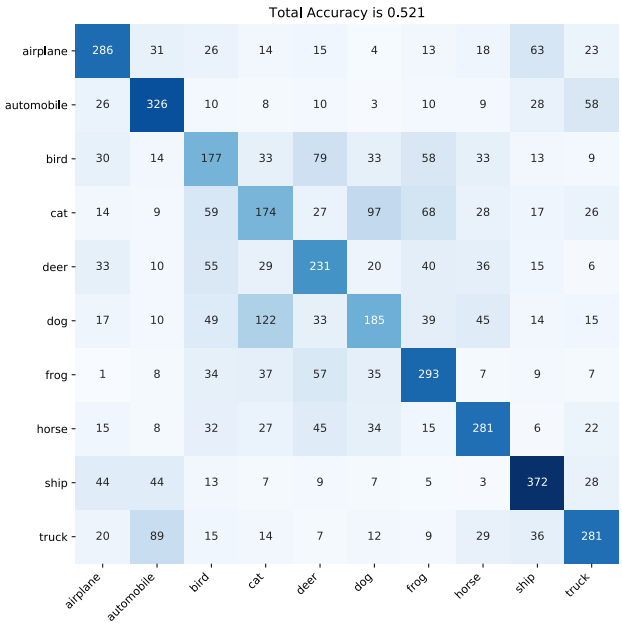

Figure 1: Reference validation set confusion matrix.

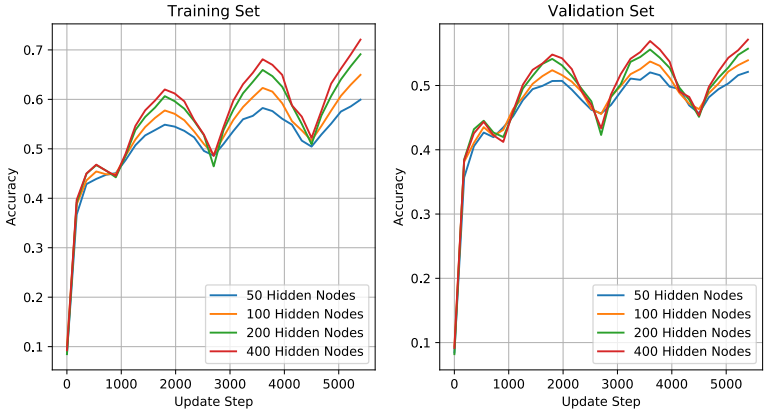

Figure 2: Learning curves for two layer networks with variable hidden layer size.

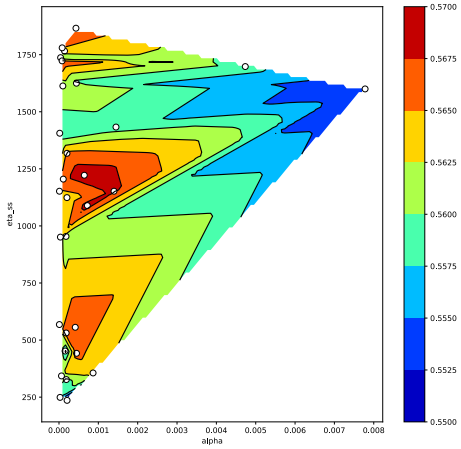

Figure 3: Coarse hyperparameter search results over λ (here called *alpha*) and η_{ss}.

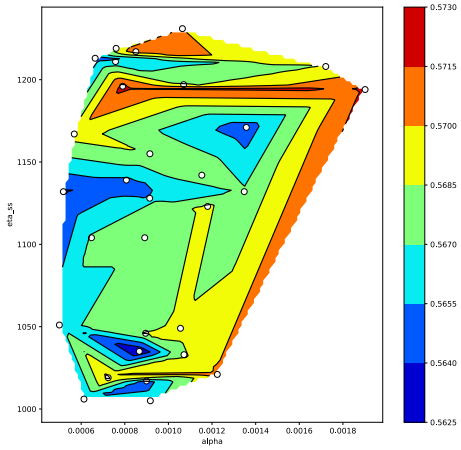

Figure 4: Fine hyperparameter search results over λ (here called *alpha*) and η_{ss}.

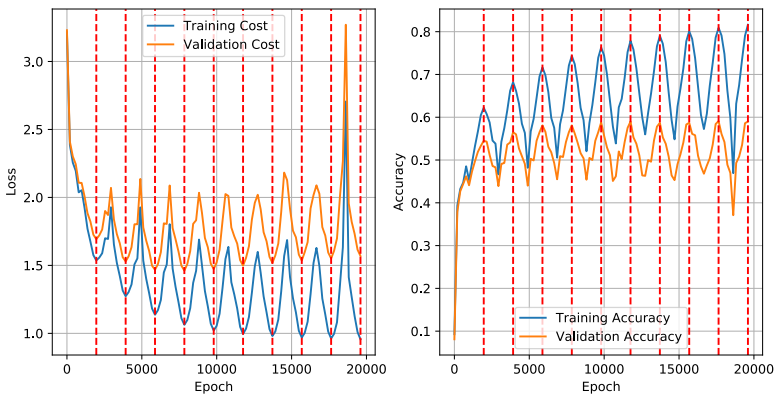

Figure 5: Ensemble learning training curve.

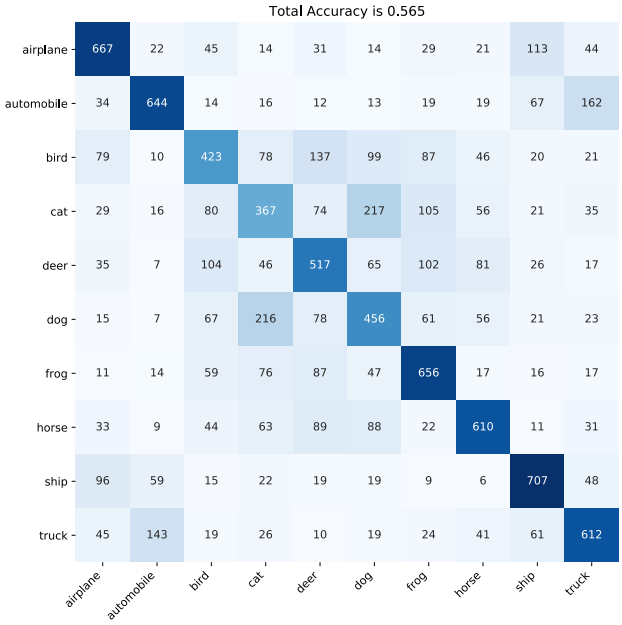

Figure 6: Ensemble classifier test set confusion matrix.

2 Finding η_{min} and η_{max}

Note that I performed this optimization independently of the other ones and did not incorporate it into the best network found in the last section because it did not seem yield any improvements.

To find good values for η_{min} and η_{max} I followed the guidelines in the paper. I executed a dummy training run in which I increased the learning rate exponentially from $1e-5$ to $1e-1$ after every parameters update. To compute the gradients I used batches of size 100, I also recorded the training set accuracy over the whole training set after each parameter update and finally plotted this accuracy as a function of the learning rate at each step. Figure 7 shows this plot. I also repeated this process but this time linearly increased the learning rate from $1e-5$ to 0.2, the results are shown in Figure 8.

Based on these figures I chose the parameters $\eta_{min} = 1e-5$ and $\eta_{max} = 0.25$. Training the final network from the non-bonus section of this assignment on 49.000 training samples with these parameters yields the test set performance shown in Figure 9 which is slightly lower than with the initial range for η.

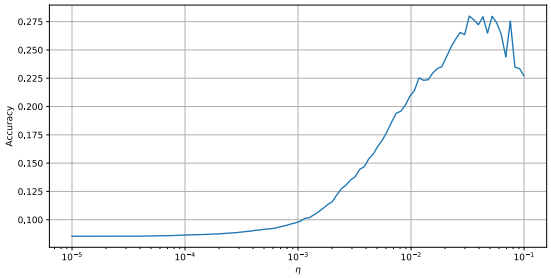

Figure 7: LR range test result for $\eta \in [1e-5, 1e-1]$ (exponentially increasing).

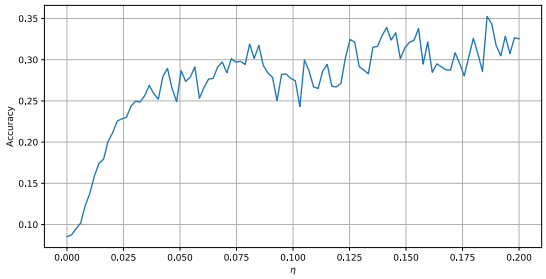

Figure 8: LR range test result for $\eta \in [1e-5, 0.2]$.

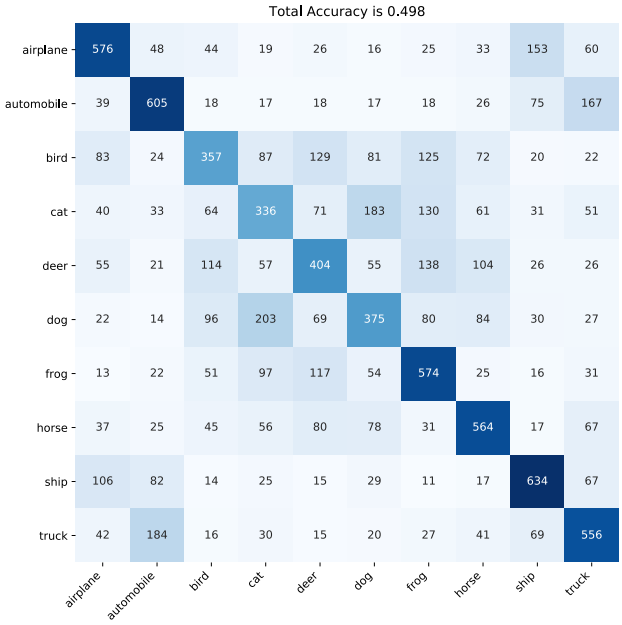

Figure 9: Test set confusion matrix for network training with $\eta_{min} = 1e-5$ and $\eta_{max} = 0.25$.