DD2424
Deep Learning in Data Science
Assignment 1 Bonus Points

Franco Ruggeri, fruggeri@kth.se

May 6, 2020

1 Improvements

I tried the following improvements:

(a) Coarse-to-fine random search also for step size (n_s) and number of cycles (n_{cycles}), besides for the amount of regularization penalty (λ).

(b) More hidden nodes.

(c) Model ensemble "on the cheap", that is built with snapshots taken at the end of each cycle.

(d) Dropout. I implemented it as inverted dropout (i.e. compensation during training) and, as suggested in [1], with different parameters for the probabilities of keeping active inputs and hidden nodes, respectively called p_x and p_h.

To analyze the benefits systematically, I enabled one improvement at a time and I compared the result with the one got without improvement. Then I trained the final network, without using explicitly the improvement (a) for reasons of training time, but setting n_s and n_{cycles} based on the result got using it alone. The results are shown in table 1. Note that λ was always adjusted with a coarse-to-fine random search, a part from test cases 3 and 10 which were used to analyze in isolation ($\lambda = 0$) the effect of dropout isolated.

We can summarize the analysis as follows:

- The most useful enhancement is the use of more hidden nodes (compare test cases 1 and 5-8).
- Building a model ensemble taking snapshots during the training can give a small boost without much computational cost (compare test cases 2 and 9).
- Dropout helps preventing over-fitting, i.e. is a regularization technique (compare test cases 3 and 11 and see figure 1). In case of high number of hidden nodes, dropout can be added to the regularization penalty and improve the generalization (compare test cases 11-12).
- There is not really a correlation with number of hidden nodes and over-fitting. This can be seen by looking at λ and m: the best value of λ (found with cross-validation) does not necessary increase when increasing m. This confirms what has been discussed during the lecture 5: over-fitting and generalization are not so well understood in neural networks.

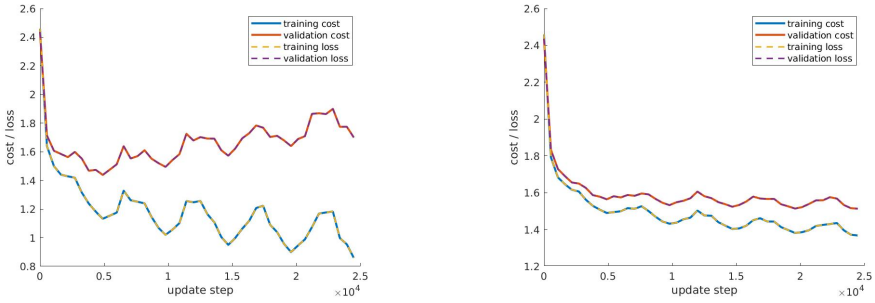
The optimization worked very fine. The accuracy was increased from 52.30% to 58.04% (final network, test case 12).

TC	λ	n_s	n_{cycles}	p_x	p_h	m	Improvements	Accuracy (%)
1	0.002584	980	3	1	1	50	-	52.30
2	0.002584	980	5	1	1	50	-	52.25
3	0	980	5	1	1	50	-	49.34
4	0.003848	2450	5	1	1	50	(a)	52.56
5	0.000012	980	3	1	1	100	(b)	53.32
6	0.002290					200	(b)	55.27
7	0.002608					500	(b)	56.09
8	0.000510					1000	(b)	56.80
9	0.002584	980	5	1	1	50	(c)	52.58
10	0	980	5	0.8	0.8	50	(d)	51.75
11	0.000510	2450	5	0.8	0.8	1000	(b)+(c)	57.14
12	0.000510	2450	5	0.8	0.8	1000	(b)+(c)+(d)	58.04

Table 1: Accuracy for several parameter settings and improvements.

2 LR range test

As explained in [2], it is possible to estimate reasonable boundary values for the learning rate (η_{min} and η_{max}) with a single training run of several epochs. Briefly, the strategy is to increase the learning rate linearly for the whole training and plot the accuracy versus the learning rate. Then, η_{min} should be set to the



(a) (b)

Figure 1: Learning curves for test case 3 (a) which clearly has over-fitting, and 11 (b) in which dropout is applied and prevents over-fitting.

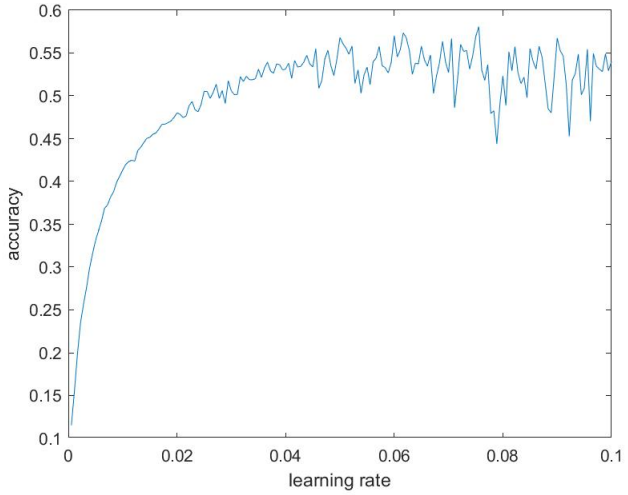


Figure 2: Accuracy as a function of the learning rate for the LR range test.

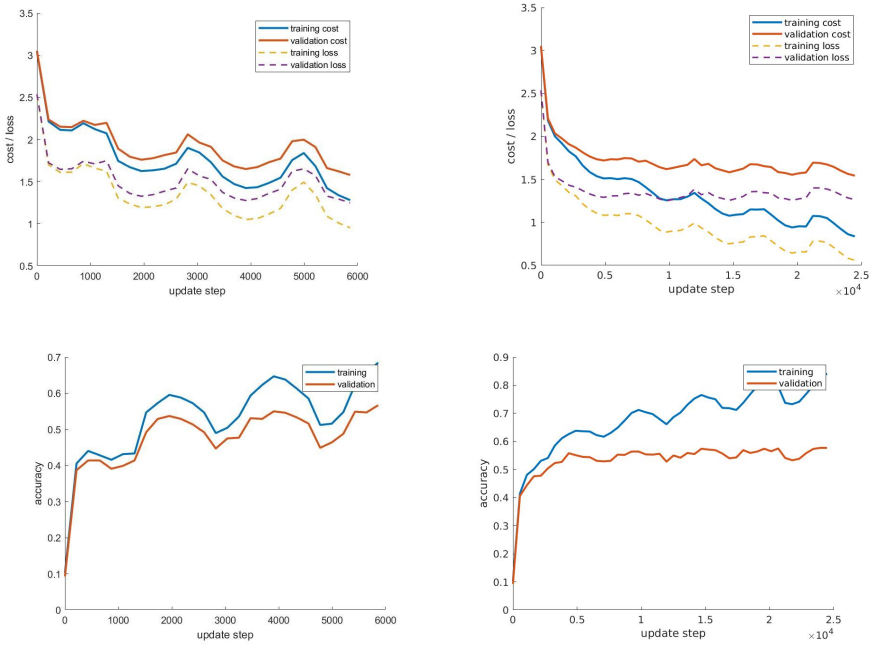


Figure 3: Learning curves for the final network, using the provided $\eta_{min} = 1e{-}5$ and $\eta_{max} = 1e - 1$ (on the left) and the found $\eta_{min} = 1e - 5$ and $\eta_{max} = 0.045$ (on the right).

value when the accuracy starts to increase and η_{max} to the value when the accuracy becomes ragged or starts to fall.

I implemented the LR range test by running a 8-epoch half cycle ($n_{cycles} = 0.5$ and $n_s = 8\ floor(n/n_{batch})$) with $\eta_{min} = 1e - 6$ and $\eta_{max} = 1e - 1$. As shown in figure 2, the accuracy starts immediately to increase, so a good value for η_{min} is 1e-5, while peaks and drops become significant after $\eta = 0.045$, which is then a good value for η_{max}. The comparison of the learning curves is shown in figure 3. The test accuracy obtained with the new boundary values is 57.6%, very similar to the previous one, confirming that the proposed LR range test is valuable.

Finally, this test is the reason of a big advantage of cyclical learning rates: they remove much of the trial-and-error associated with finding a good learning rate and some of the costly hyper-parameter optimization for adaptive learning rates, as stated in the instructions.

References

[1] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. `http://www.deeplearningbook.org`. Cambridge, MA, USA: MIT Press, 2016.

[2] Leslie N. Smith. *Cyclical Learning Rates for Training Neural Networks*. 2015. arXiv: `1506.01186 [cs.CV]`.