MSc Machine Learning

DD2424 Deep Learning in Data Science

Assignment 2: Image classification with a 2-layer network

Author:
Alexandros Ferles
ferles@kth.se

Professor:
Josephine Sullivan

Contents

1 Read in the data & initialize the parameters of the network

The reader is advised to check out the corresponding function either of the source code file or of the Jupyter notebook.

2 Compute the gradients of the network parameters

i) State how you checked your analytic gradient computations and whether you think that your gradient computations were bug free. Give evidence for these conclusions. ii)

In order to check the quality of the computed gradients, we conduct the following checks:

1. Cross check with the numerically computed gradients for the first two data of the first batch.
2. Overfit our network in a small sample of the training data of the first batch.

For the first of these checks, our results using a threshold of 10^{-5} are the following:

```
Success!!
Average error on weights of first layer=  7.52283628908819e-09
Average error on bias of first layer= 4.602426280088195e-09
Average error on weights of second layer=  3.0747216543846533e-10
Average error on bias of second layer=  1.193638211533644e-11
```

For the second check, following the cross-entropy loss evolution on both the training and validation set we can observe overfitting, since the loss on the training set gets a small value, while in the validation set the same loss drops for a few number of epochs, and later starts to increase.

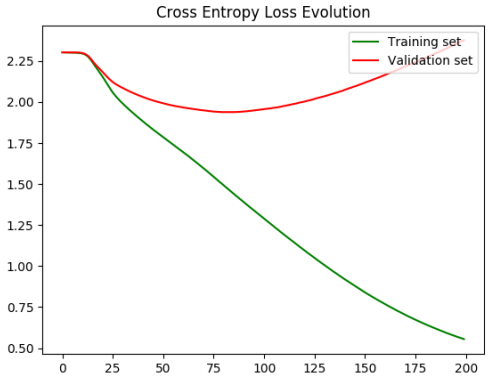


Figure 1: Overfitting in a small batch of the training data

3 Add momentum to your update step

After adding momentum updates to the training process, we conduct the same experiment with the previous exercise, using momentum factor values fo 0.5, 0.9 and 0.99 respectively. The cross-entropy loss for each of these settings, can be found in the following plot:

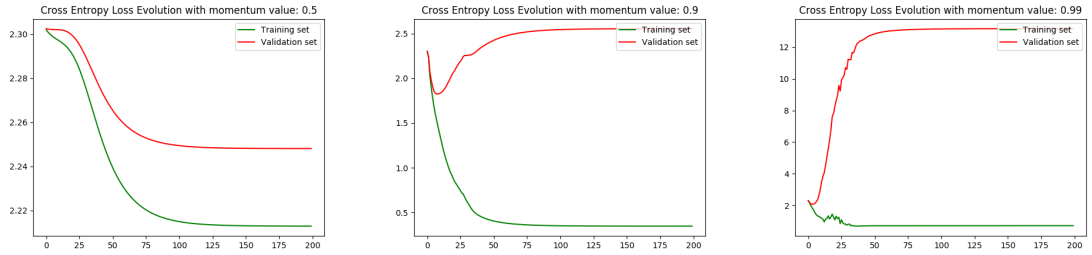


Figure 2: Cross entropy loss evolution with several values for the momentum update factor in a small sample of the training data

ii) Comment on how much faster the momentum term made your training.

As we can see from the above graphs, with the use of the momentum factor the training process speeds up. While for a small value (0.5) we could not overfit on the training data (under the scope that the the loss on the validation set is not increasing after a few number of epochs), by using high values for the momentum factor, the model not only overfits quickly on the training data, but also achieves the lowest training set loss value in a fewer number of training epochs compared to a training setting that makes no use of the momentum update approach.

4 Training your network

To define appropriate values for the learning rate η and the amount of the regularization λ, we firstly conduct a random search in a broad space, drawing values from
{0.00001, 0.00005, 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.2, 0.3, , 0.4 0.5, 0.6, 0.7, 0.8, 0.9, 1} for η, from where we defined the limits of the coarse search for appropriate values of the earning rate to be between 0.01 and 0.15.

State the range of the values you searched for lambda and eta, the number of epochs used for training during the coarse search and the hyper-parameter settings for the 3 best performing networks you trained.

Additionally to the aforementioned values for eta, different amounts of regularization were tested, drawn from $\{0,\ 10^{-6},\ 10^{-6},\ 10^{-5},\ 10^{-4},\ 10^{-3},\ 10^{-2},\ 10^{-1},\ 1,\ 10\}$. The best hyperparameter settings were the following:

```
BEST PERFORMANCE:  44.07
Best eta:  0.02878809988519304
Best lambda:  0.001
---------------------------------
SECOND BEST PERFORMANCE:  43.99
Second best eta:  0.030687177183315043
Second best lambda:  0.001
---------------------------------
THIRD BEST PERFORMANCE:  43.91
Third best eta:  0.02534053640834971
Third best lambda:  1e-05
```

State the range of the values you searched for lambda and eta, the number of epochs used for training during the fine search, and the hyper-parameter settings for the 3 best performing networks you trained.

From observing the results of the coarse search, various combination for λ and η were tested:

- η's drawn from [0.017, 0.019] and $\lambda = 10^{-6}$
- η's drawn from [0.027, 0.029] and $\lambda = 10^{-6}$
- η's drawn from [0.022, 0.027] and $\lambda = 10^{-5}$
- η's drawn from [0.017, 0.019] and $\lambda = 10^{-4}$
- η's drawn from [0.028, 0.032] and $\lambda = 10^{-3}$

In total, 70 experiments were conducted at this step. The 3 best settings observed are the following:

```
BEST PERFORMANCE:  44.44
Best eta:  0.01713848118474131
Best lambda:  0.0001
---------------------------------
SECOND BEST PERFORMANCE:  43.91
Second best eta:  0.018394727031491028
Second best lambda:  1e-06
---------------------------------
THIRD BEST PERFORMANCE:  43.71
Third best eta:  0.029193158018971287
Third best lambda:  0.001
```

Since not much of the experiments achieved validation-set accuracy performance around 44% , a second round of another experiments was conducted, leading to the following results:

```
BEST PERFORMANCE:  44.16
Best eta:  0.018920249916784752
Best lambda:  0.0001
---------------------------------
SECOND BEST PERFORMANCE:  43.89
Second best eta:  0.02899354379782664
Second best lambda:  0.001
---------------------------------
THIRD BEST PERFORMANCE:  43.73
Third best eta:  0.026925585997753746
Third best lambda:  1e-05
```

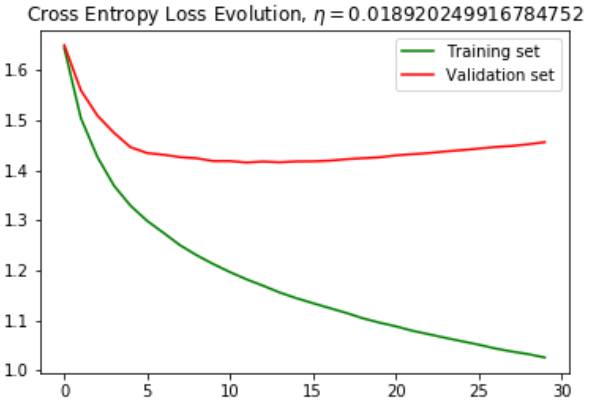
The best 3 settings of both rounds, are presented in the following table:

Performance rank	η	λ	Validation set accuracy
1	0.01713848118474131	0.0001	**44.44**%
2	0.018920249916784752	0.0001	**44.16** %
3	018394727031491028	10^{-6}	**43.91**%

For your best found hyper-parameter setting (according to performance on the validation set), train the network on all the training data (all the batch data), except for 1000 examples in a validation set, for ∼30 epochs. Plot the training and validation cost after each epoch of training and then report the learnt network's performance on the test data. Exercise

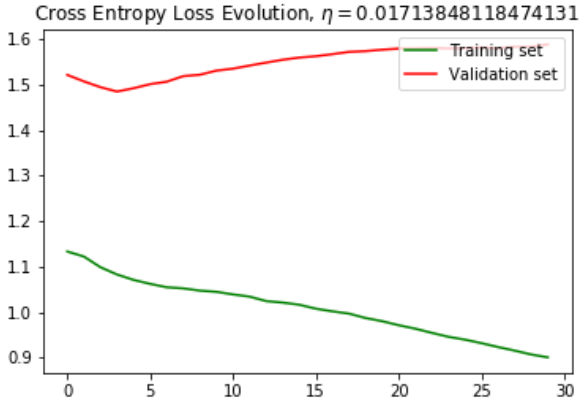
We make us of the 2 best performances of the fine search along with the best performance of the coarse search, which succeeded to exceed 44% accuracy performance on the validation set, after a training process of 10 epochs.

1. Experiment no.1: $\eta = 0.018920249916784752$, $\lambda = 0.0001$



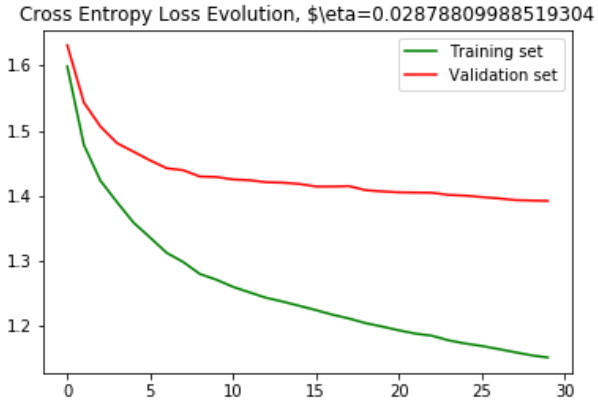
The test-set accuracy performance of this setting is 50.23%

2. Experiment no.2: $\eta = 0.01713848118474131$, $\lambda = 0.0001$



The test-set accuracy performance of this setting is 49.14%

3. Experiment no.3: $\eta = 0.02878809988519304$, $\lambda = 0.0001$
For this setting, the best model in terms of validation set accuracy performance was tracked, in order to avoid phenomenas as overfitting which might be the case in experiment 1.



The test-set accuracy performance of this setting is 51.53%