EXERCISE 03 - DEEP LEARNING LAB COURSE 2016

Scenario 1

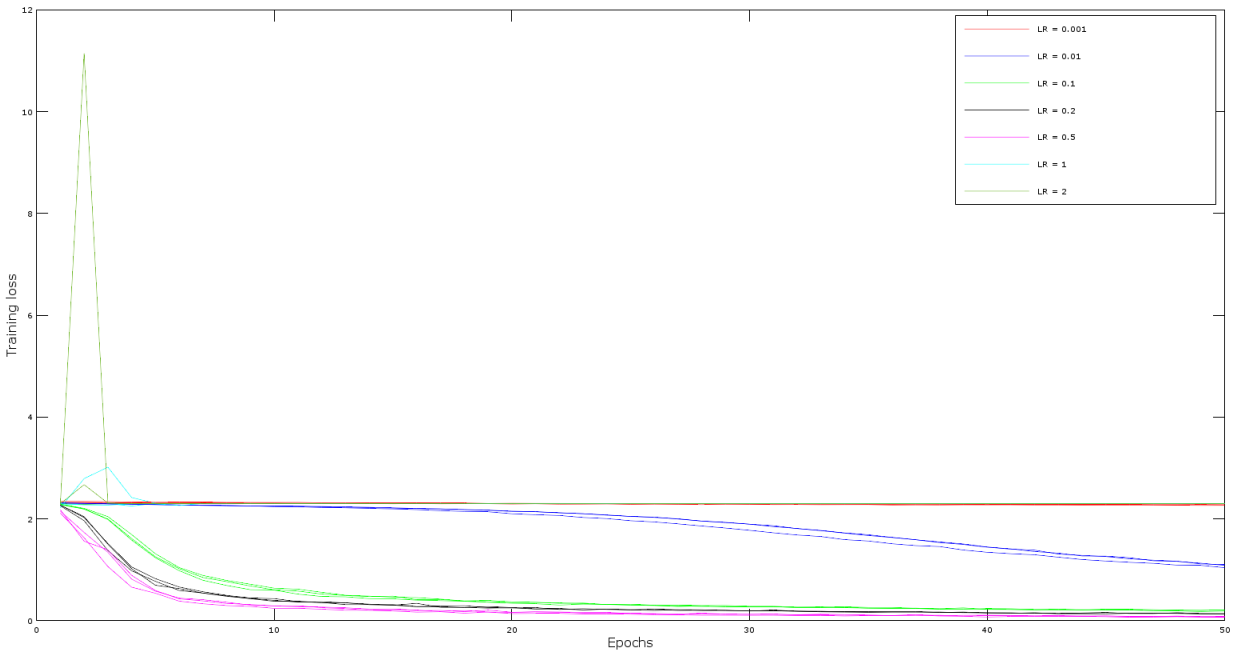

Figure 1.1 - Train loss

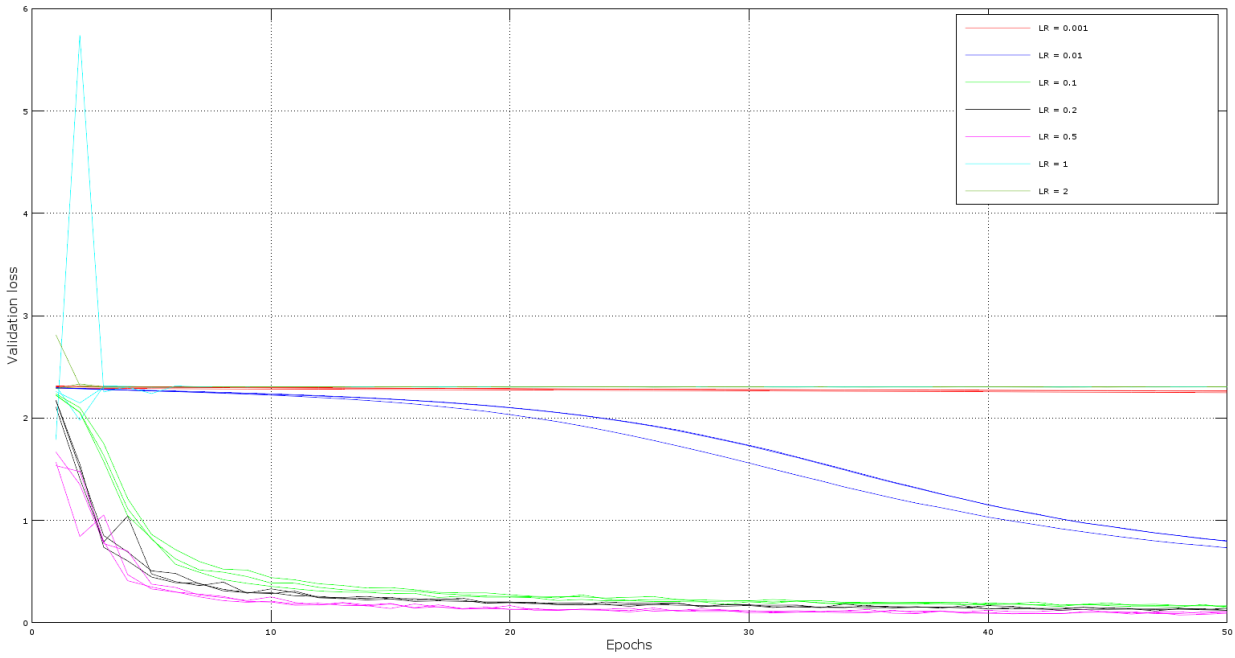

Figure 1.2 - Validation loss

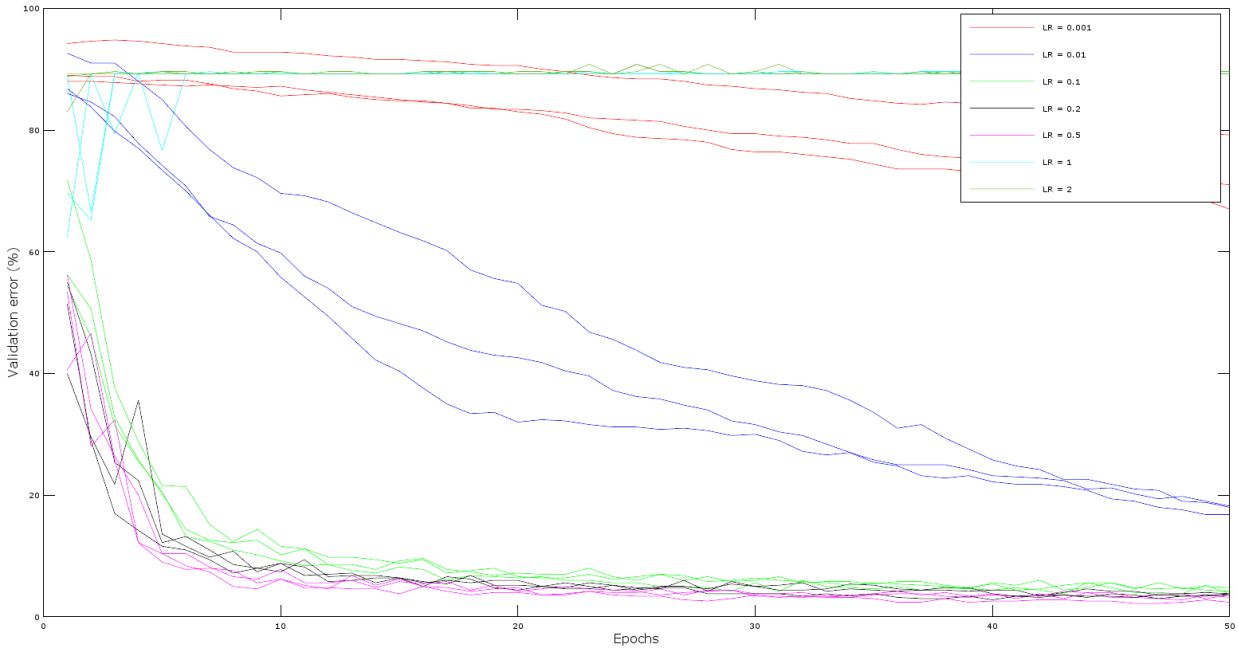

Figure 1.3 - Validation error

Judging by data from figures, we can see that neural network performs poorly with LR >= 1.0 and LR < 0.01. In cases with greater LR we can see a results of fast overfitting, when we have exceeded the local minimum of loss function and after what neural network basically stopped to learn anything from the data; cases with lower LR show that neural network didn't learn much since the beginning of training, what means that we will require a lot more epochs to reach the results of those with more optimal LR.

Q: Please describe ALL sources of stochasticity of the training process (why curves of different runs are not exactly the same).

A: Sources of stochasticity:

- Weights – initialized randomly
- Input data – can contain noise, which can be differently interpreted by network in each run
- Output data/predictiions(?)
- Loss function(?)
- SGD (?)
- Train/validation error(?)
- else?

Scenario 2

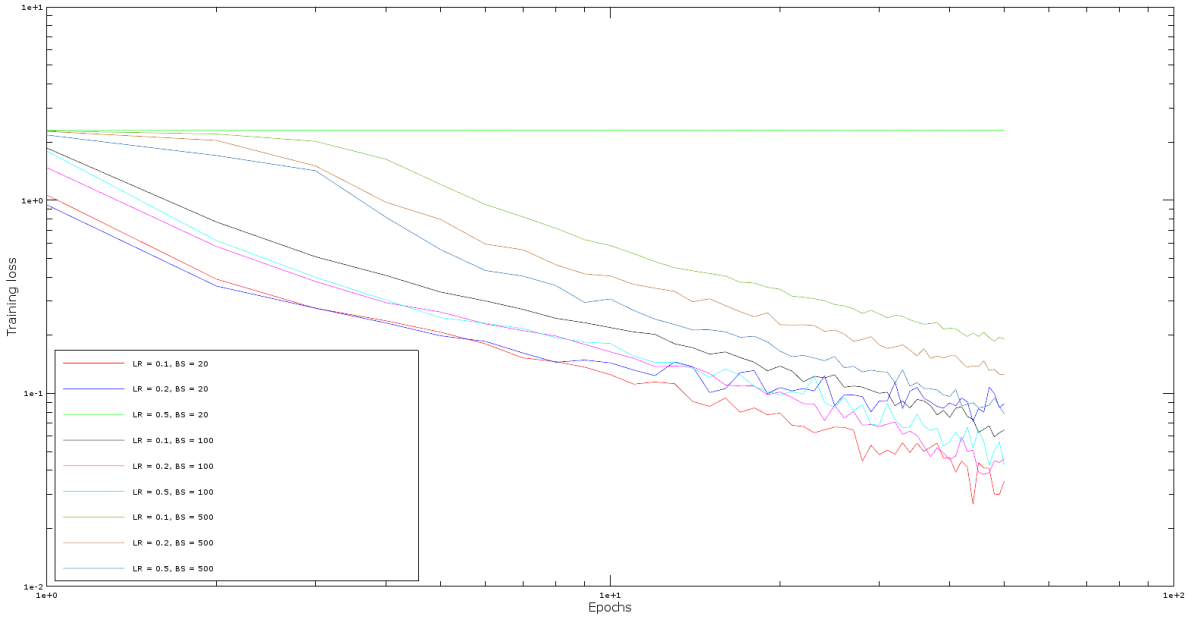

Figure 2.1 - Epoch/Training loss

Q: Investigate and describe which learning rates work best for each setting of batch size. Please make a guess how batch size and learning rate might be interdependent. Take into account that the greater the batch size the more steps SGD can perform within one epoch.

A: Smaller learning rates works better with smaller batch sizes, which also has a positive effect on network learning curves. I guess smaller batch size means that we take in less data to learn from and, using the same network setup, we can learn more information from it on each update.Also, during gradient descent, we do a smaller step, which reduces our chance of missing the local minimum.

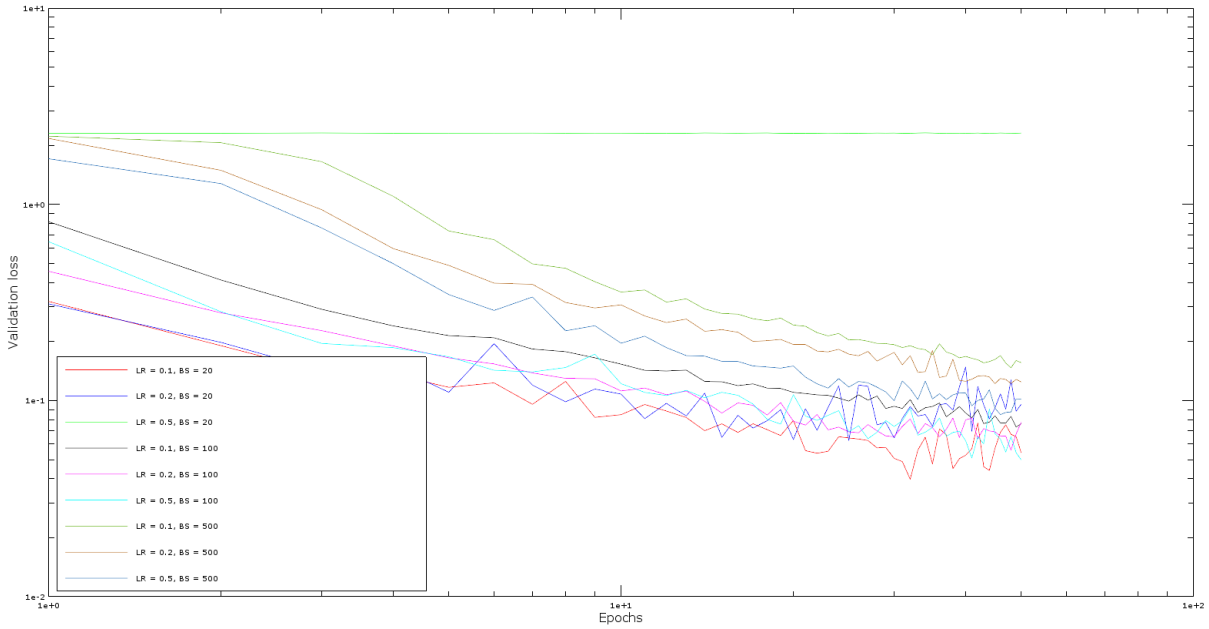

Figure 2.2 - Epoch/Validation loss

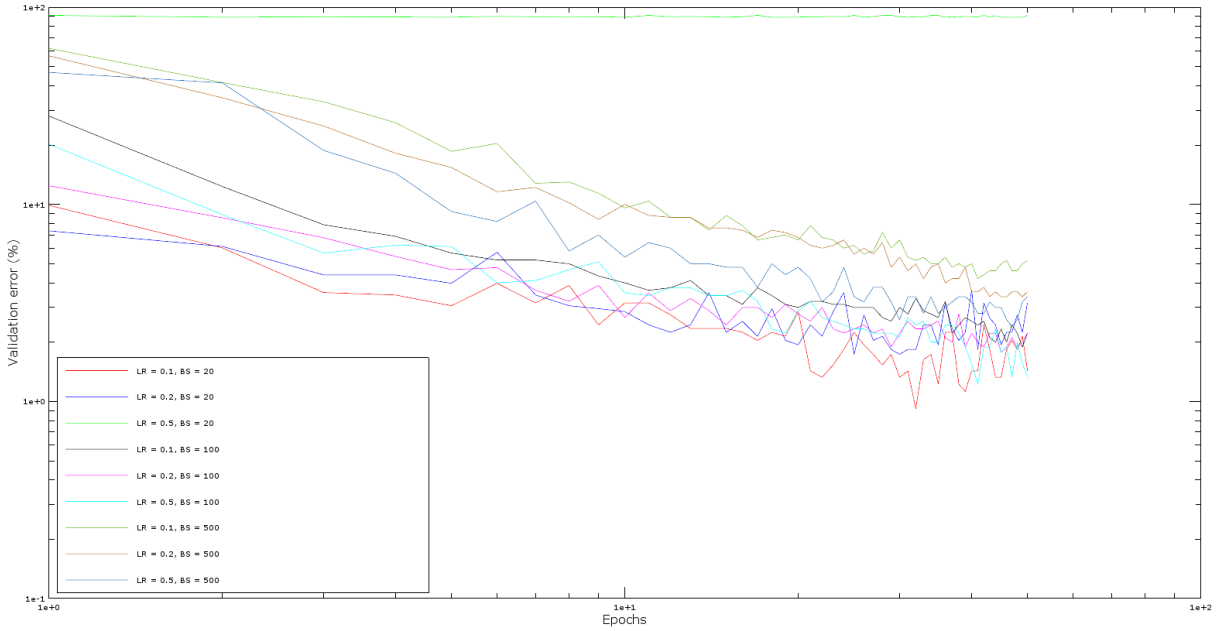

Figure 2.3 - Epoch/Validation error

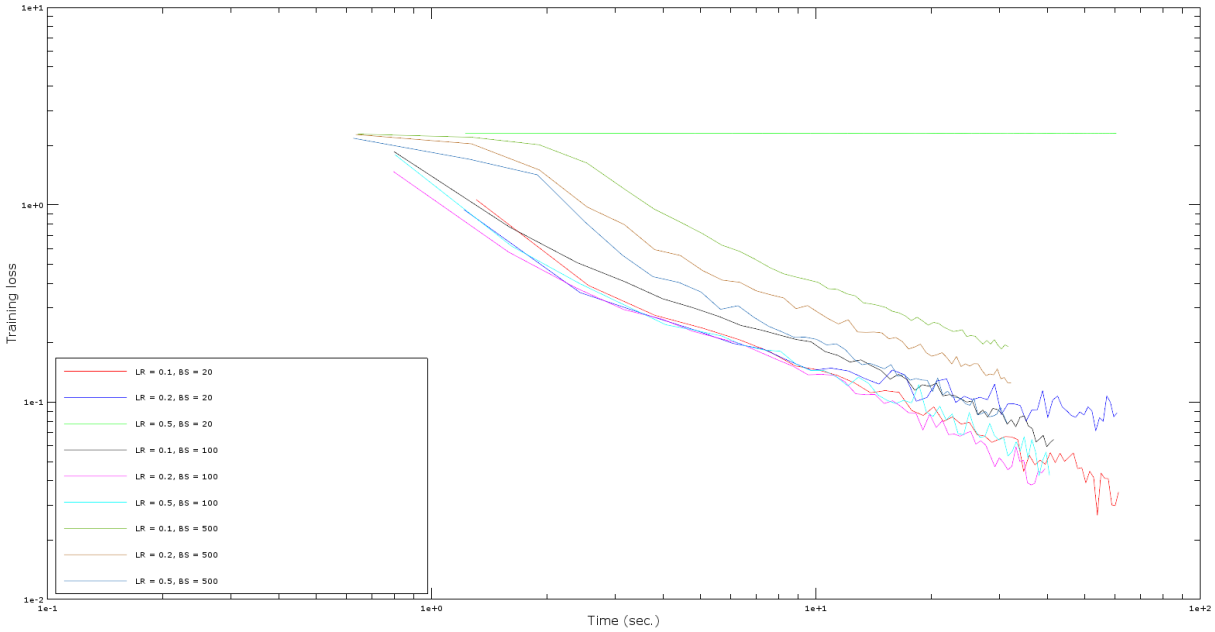

Figure 2.4 - Time/Training loss

Q: Please explain why curves ends at different positions on x-axis and why these positions are not different by a factor of 5 (e.g., some T, 5T and 25T for batch size 500, 100 and 20, respectively).

A: Because we use non-linear functions with multiple parameters for learning and loss is not governed by time or batch size only.

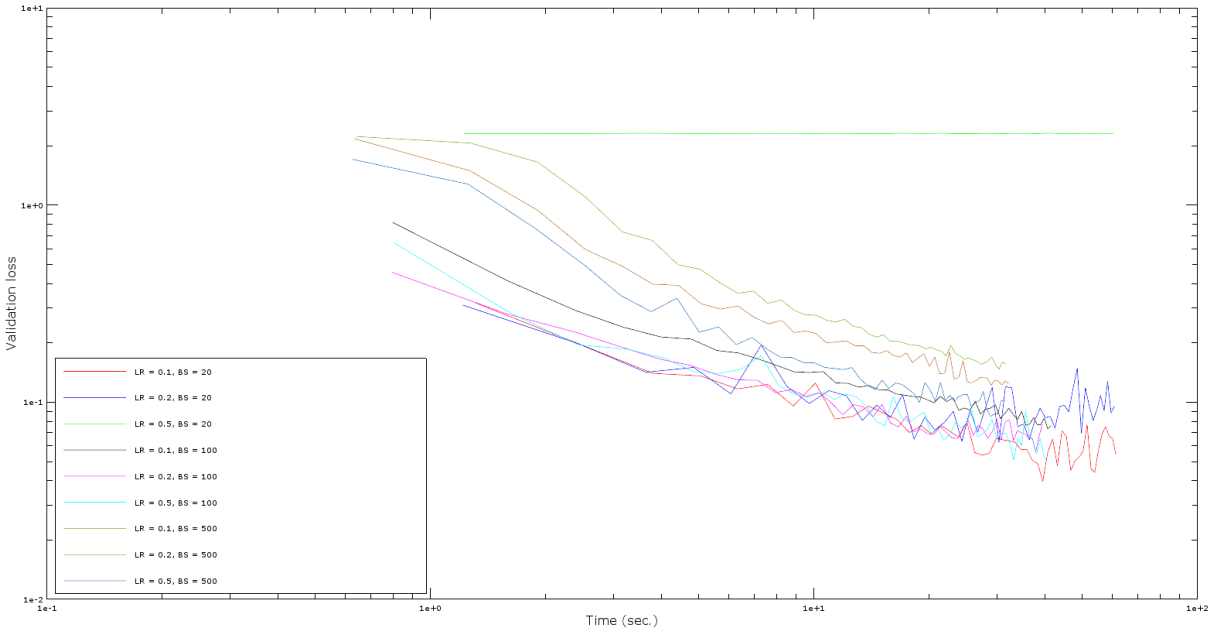

Figure 2.5 - Time/Validation loss

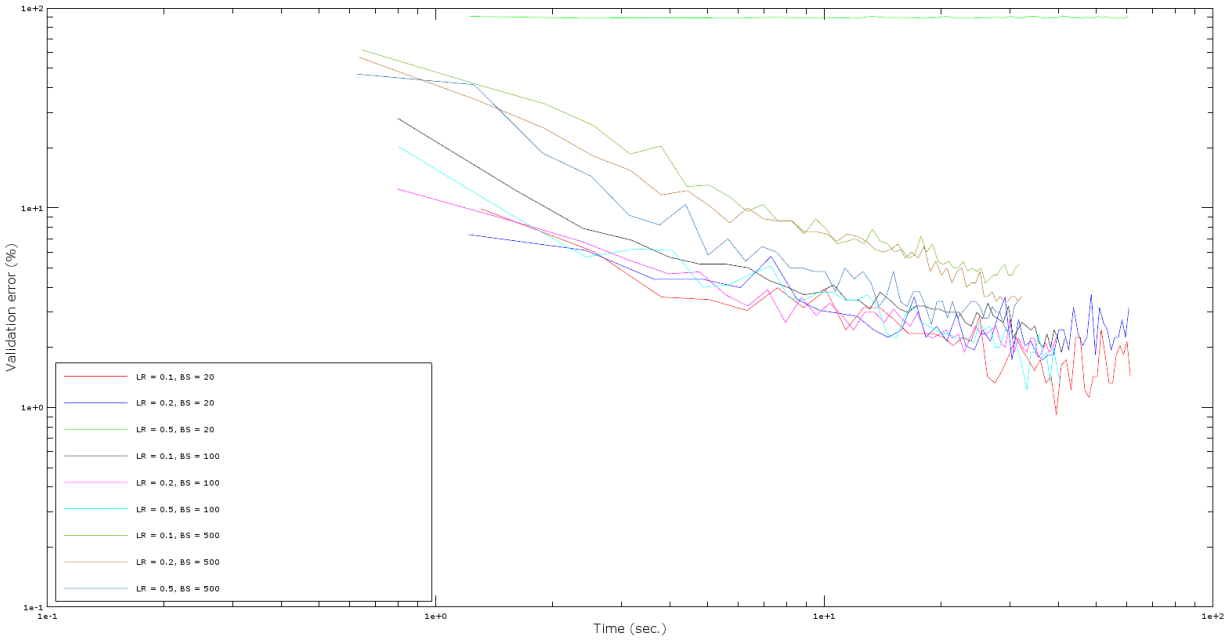

Figure 2.6 - Time/Validation error

Scenario 3

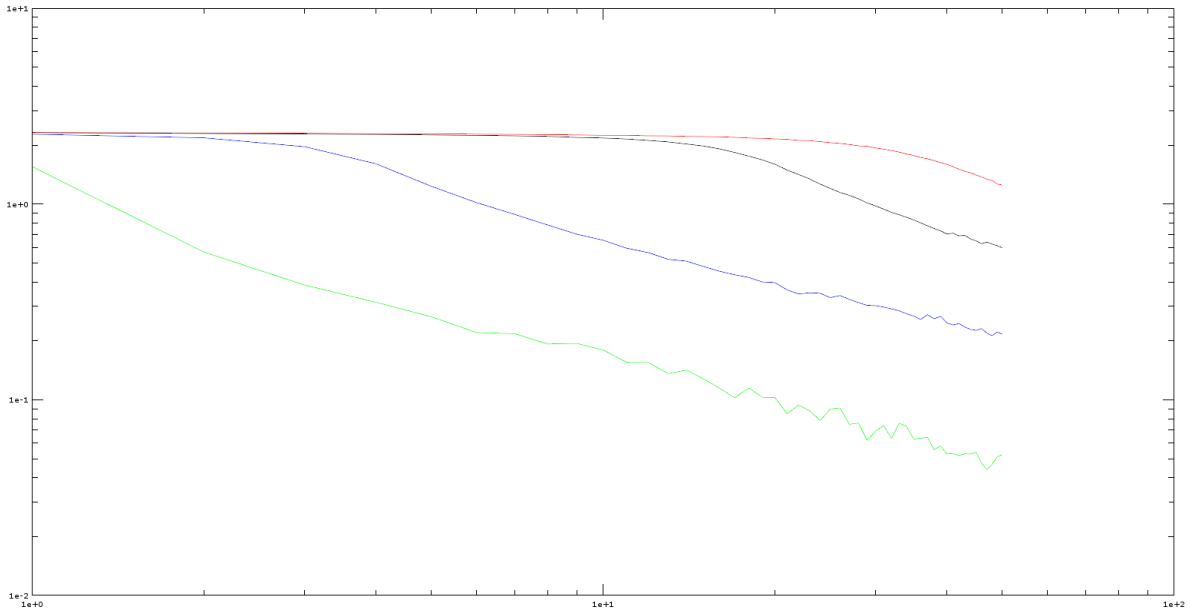

Figure 3.1 - Epoch/Training loss

Q: Investigate and describe which learning rates and momentum factors work best. Please don't forget to have a look at Lasagne's documentation of SGD with momentum. If momentum is useful, please describe why it might be the case.

A: Bigger momentum works the best with bigger learning rates, so, we can say it can be useful when we want a more "smoothed" results and are using LR =0.1. For LR= 0.01 or less would be useful to take a smaller momentum >= 0.5, which still can smooth some noise but less effective that with big learning rates.

(need to use a diffent image, will add later)

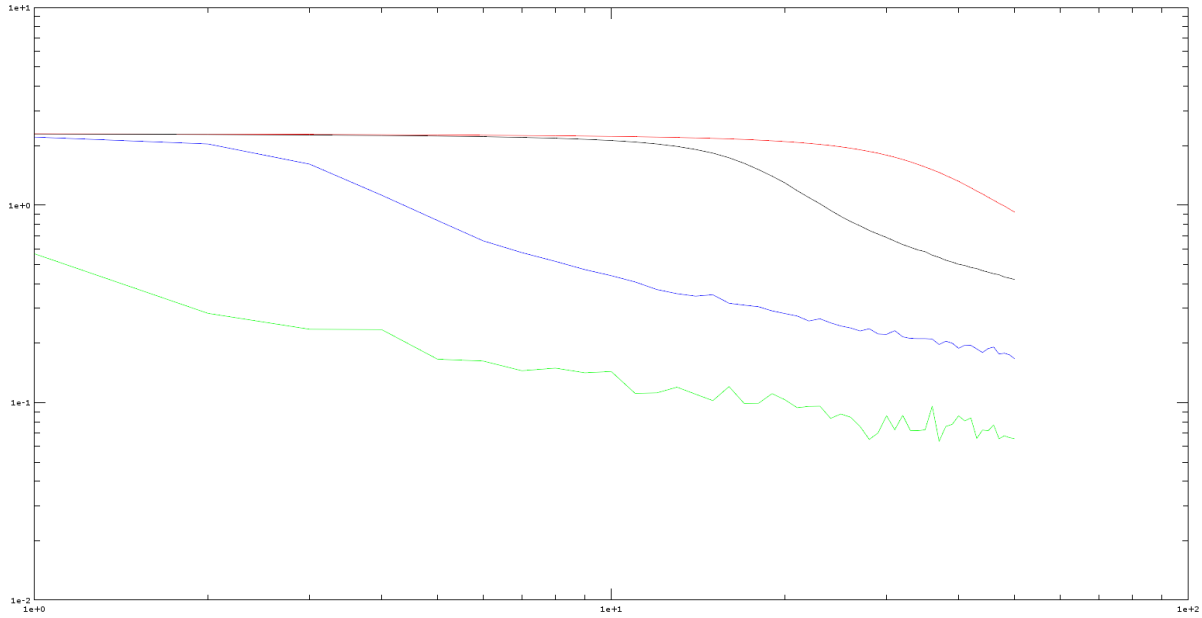

Figure 3.2 - Epoch/Validation loss

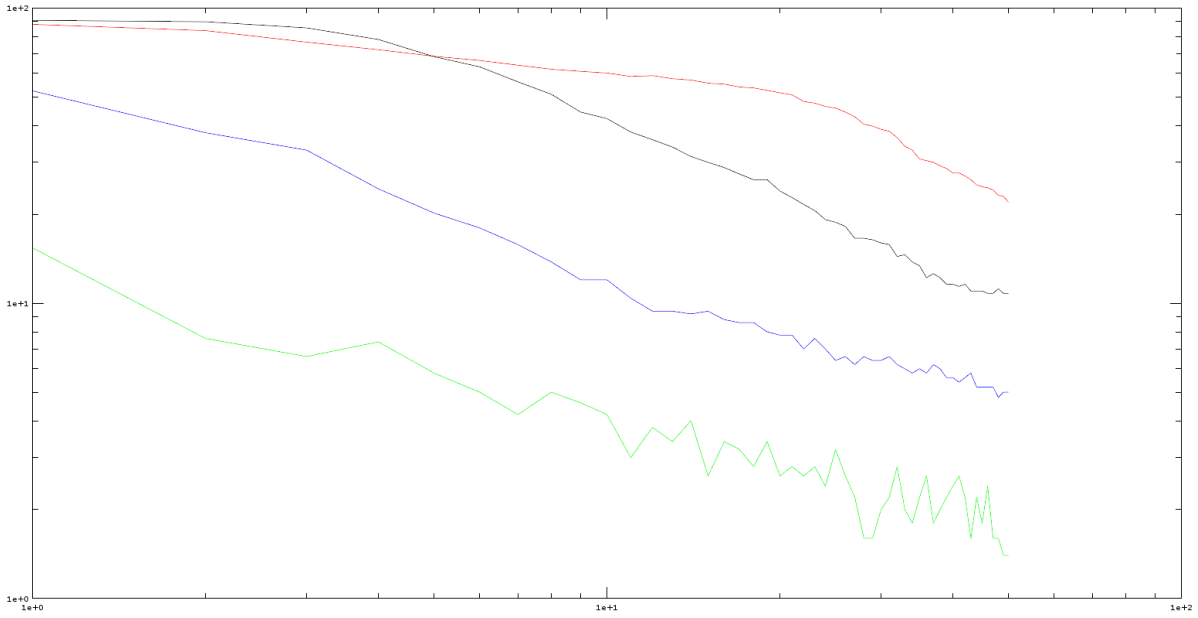

Figure 3.3 - Epoch/Validation error

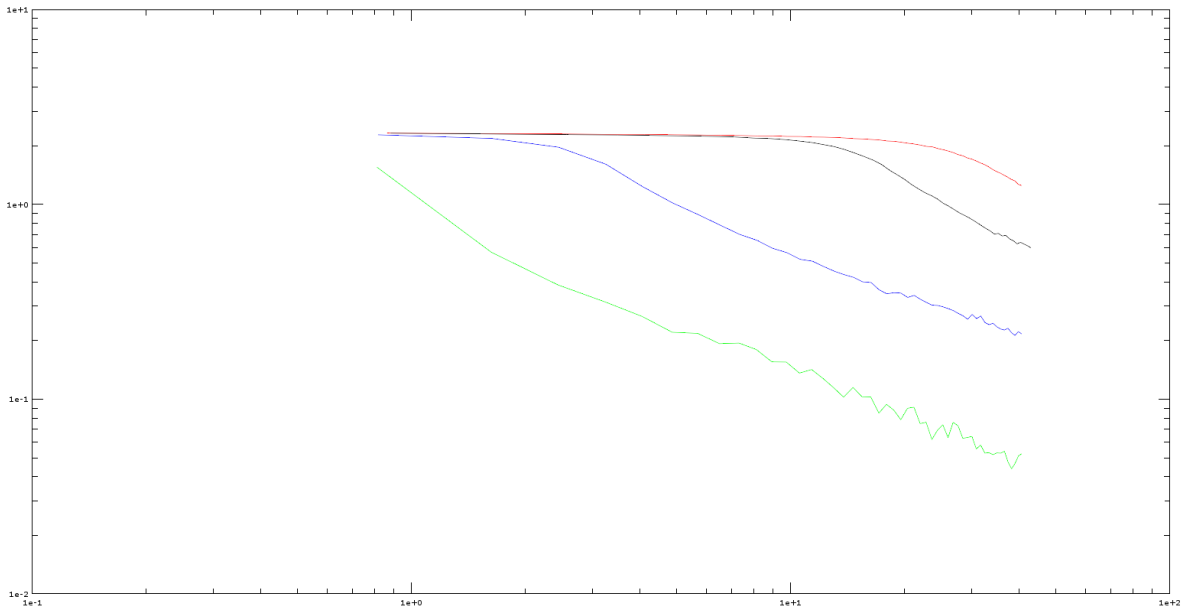

Figure 3.4 - Time/Training loss

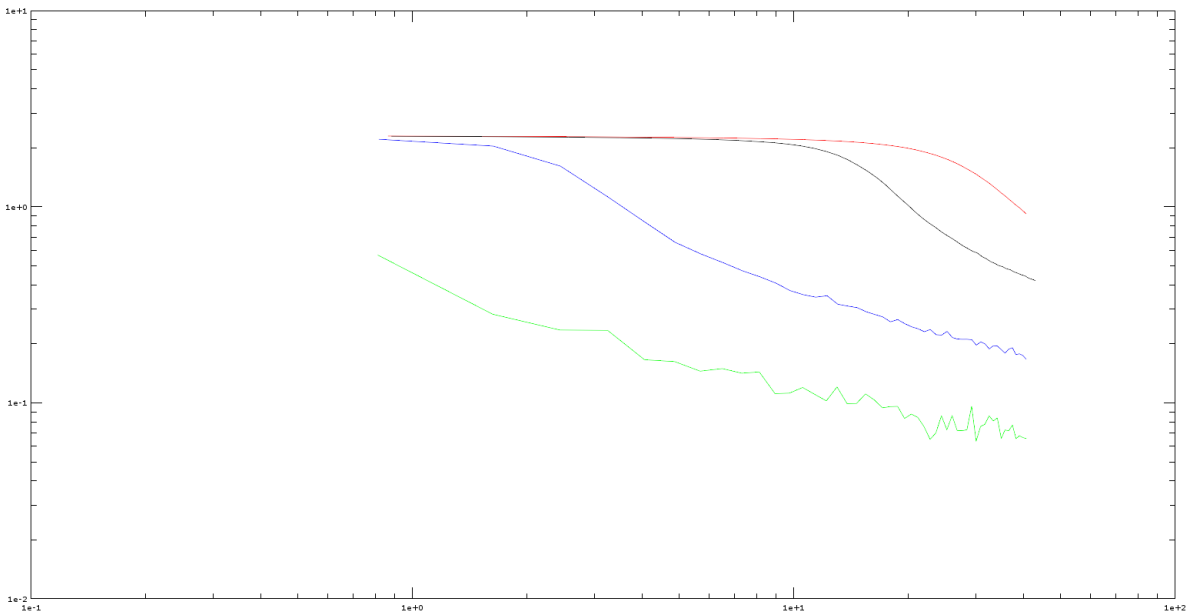

Figure 3.5 - Time/Validation loss

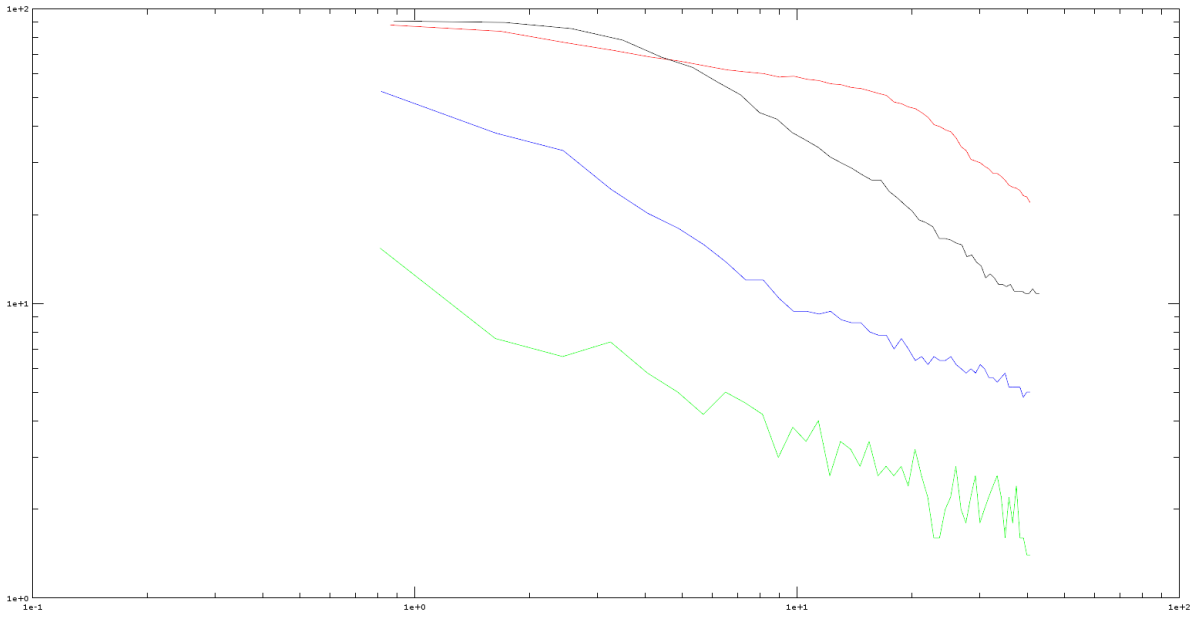

Figure 3.6 - Time/Validation error

Scenario 4

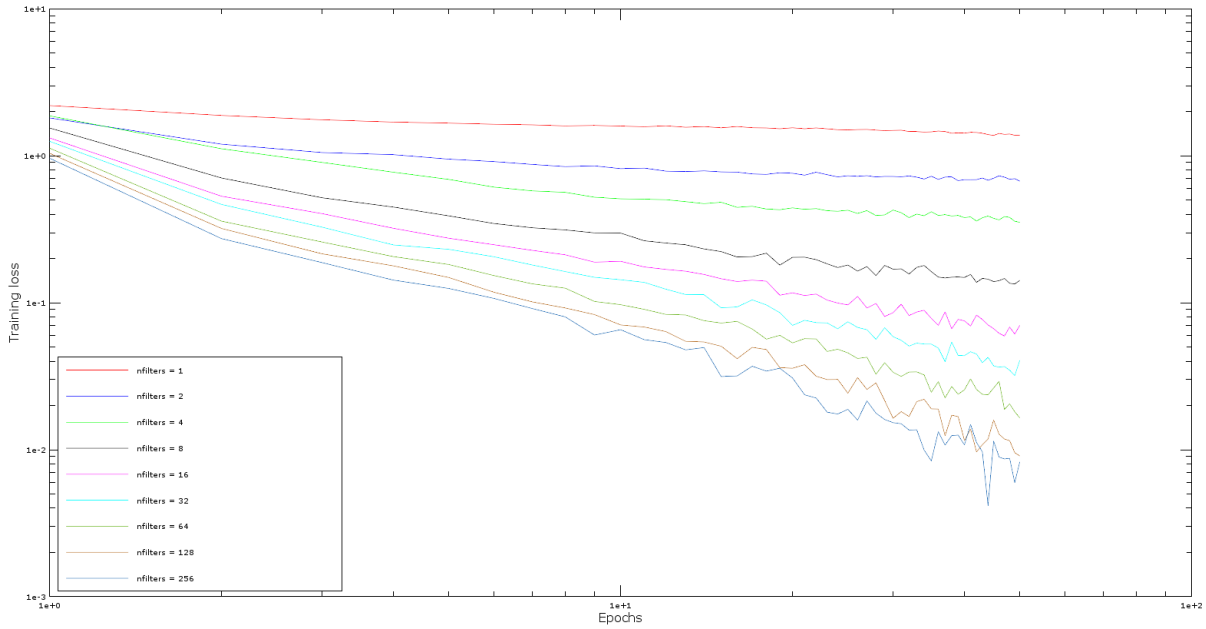

Figure 4.1 - Epoch/Training loss

Q: Investigate and describe how training curves change w.r.t. the number of filters used. Which number of filters you would select if in all cases you are allowed to train your network for 50 epochs.

A: Judging by curves, we can see that the more filters you use, the lesser loss you get, however the output become more “noisy” as near to the end, graphs of functions with nfilters>= 128 start to drastically jitter, which means that network might be giving the predictions inccorectly. For the case of 50 epoch training I would peek nfilters=64 or nfilters=32 since these graphs have a better training loss than of those with smaller number of filters and are more “stable” than of more filters.

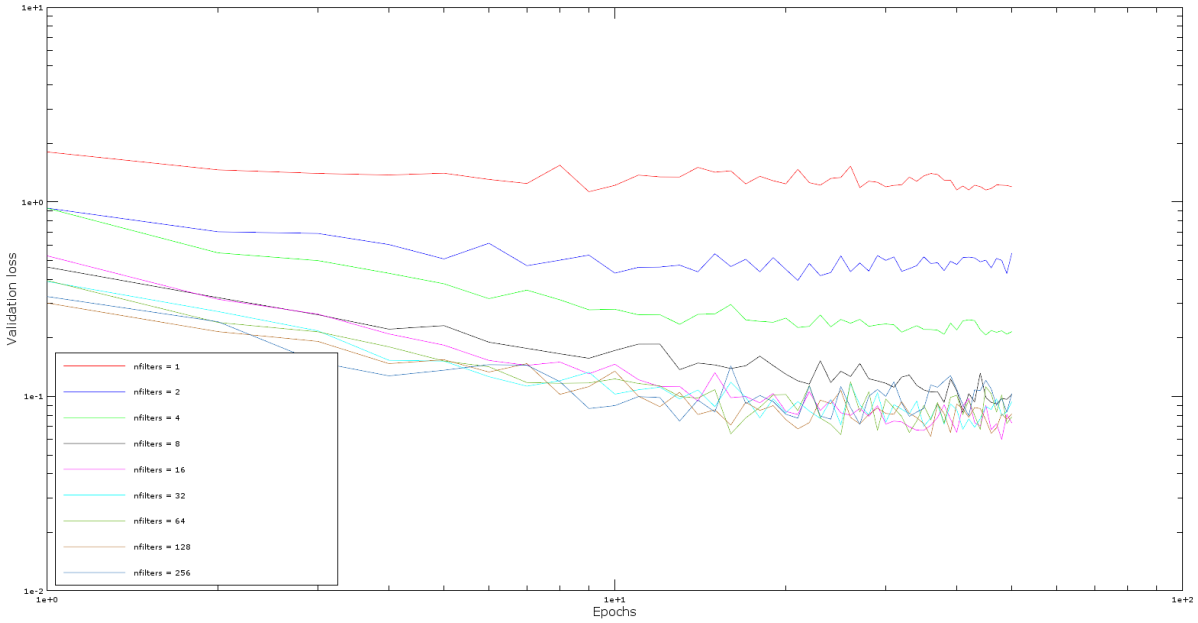

Figure 4.2 - Epoch/Validation loss

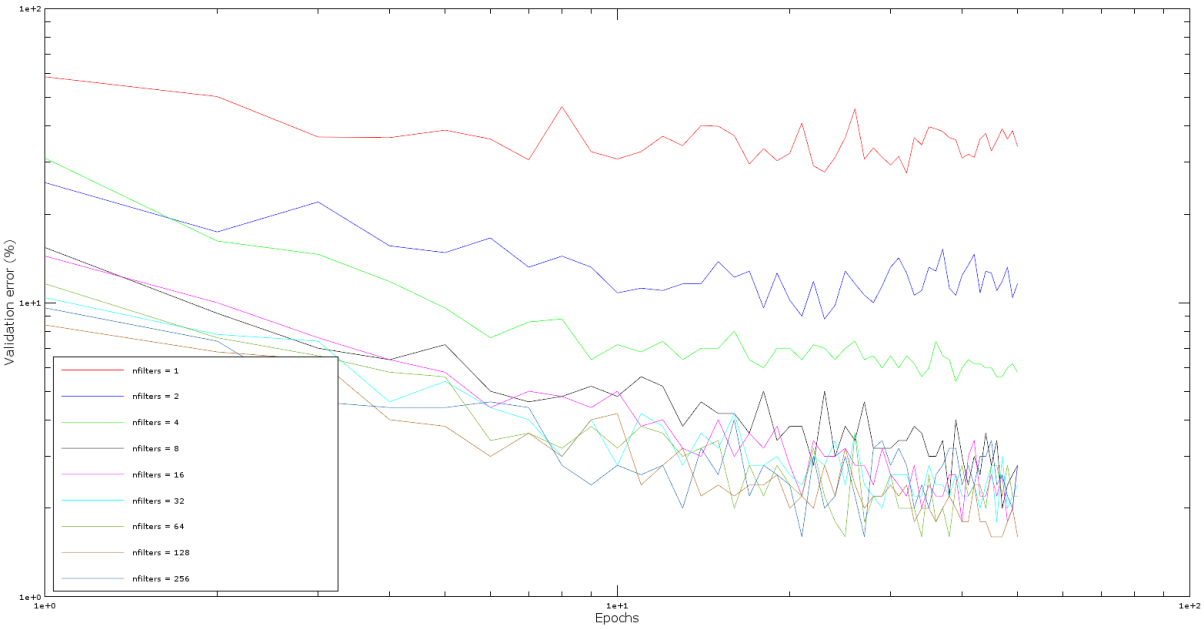

Figure 4.3 - Epoch/Validation error

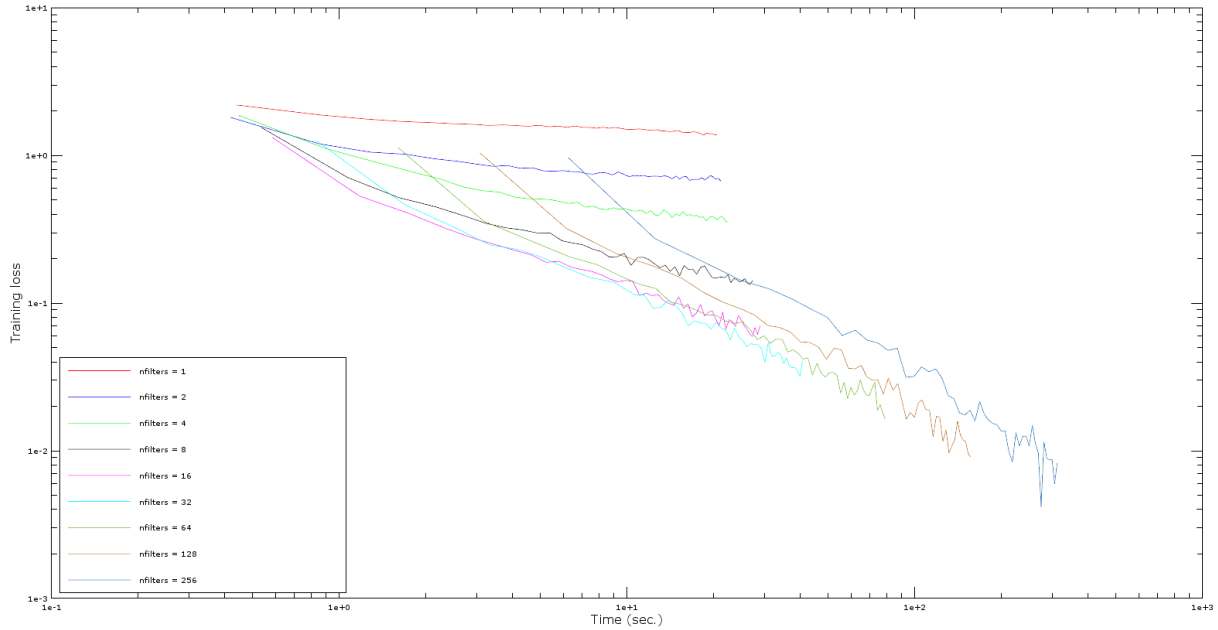

Figure 4.4 - Time/Training loss

Q: Please describe the difference between the two plots. Which number of filters you would use when your time budget is limited to 10 seconds.

A: In this case I would use functions which have a better (lesser) loss and are not very “noisy”. Such functions would be with nfilters =[16, 32, 64].

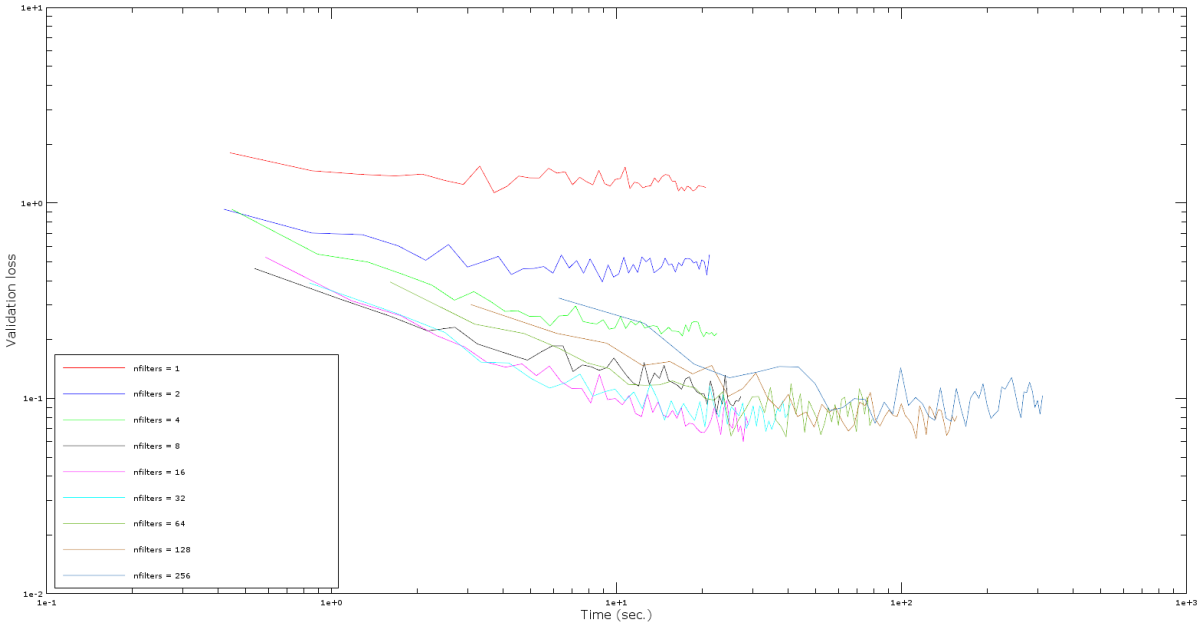

Figure 4.5 - Time/Validation loss

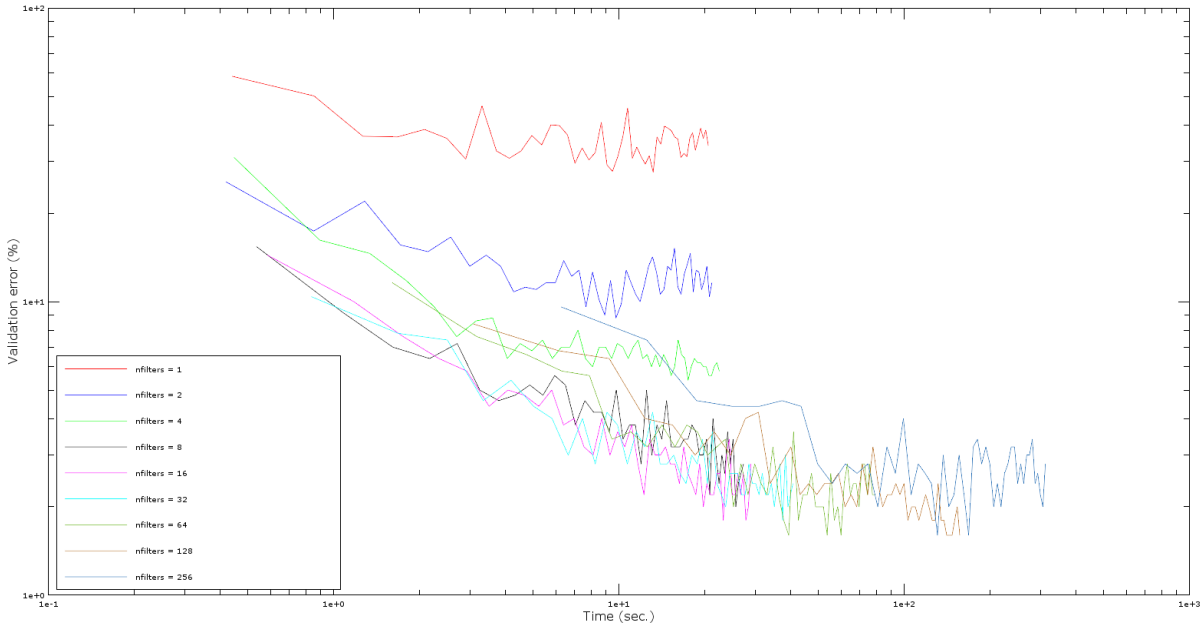

Figure 4.6 - Time/Validation error

Q: What this plot suggests?

A: This plot suggests that for this data set use of intermediate number of filters gives almost the same results as big number of filters but with smaller noise and faster calculations.