# Job for the Lab 5

Purpose of work: study of the hyperparameters of the neural network, understanding impact of different hyperparameters on accuracy.

# Installing jupyter notebook and preparing it to work with neuralnets

Preparing the environment:

1) Download and install Miniconda from <https://conda.io/miniconda.html>

1.1) Don't forget to add conda in %PATH

2) Save environment.yml in working directory

3) Execute in shell / cmd: "conda env create -f environment.yml"

3.1) Wait until all dependencies will be resolved.

Note: It's possibly if some of dependencies won't be resolved (Ex. cython).

If so, download missed ones by yourself using conda.

4) Open "$HOME/.keras/keras.json" for Linux or "%USERPROFILE%\.keras\keras.json" for Windows

5) Change "backend" value from "tensorflow" to "theano"

Running Jupyter notebook:

1) Open shell / cmd

2) $ activate neuralnets

3) $ jupyter notebook

If you have a video card from NVIDIA and it is supported by CUDA, then you can speed up calculations

by using a video card:

\* Download CUDA Toolkit from <http://developer.nvidia.com/cuda-downloads>

# Work with jupyter

There are 2 parts of work in files named Lab1-Part1 and Lab1-Part2 respectively. Both parts represent work with collections of training and test data.

Part 1 represent recognition of basic math functions with illustrations of neural net vision of the functions for training.

Part 2 represent recognition of simple images for making following work.

Data usage represented at <https://keras.io/datasets/> and depends on variant.

The number of your variant is N mod 4 + 1, where N is your number in the group.

|  |  |  |  |
| --- | --- | --- | --- |
| Var | Part1 func | Part2 data | Hyperparameters\* |
| 1 | Absolute(Sin(x)) X: -6.3..6.3 Y: 0..1.2 | CIFAR10 | Layers count, neurons count per layer |
| 2 | Cos(x) X: -9..9 Y: -1..1 | CIFAR100 | Learn rate, regularization L1 |
| 3 | Absolute(Sin(x)) X: -6.3..6.3 Y: 0..1.2 | Handwritten digits | Regularization L2, output layer activation type |
| 4 | Cos(x) X: -9..9 Y: -1..1 | Fashion articles | Layer activation type, loss function type |

\* If you cannot reach the required accuracy you may change other hyperparameters.

There are represented such hyperparameters as

* Layer count
* Neurons count per layer (actually it’s not hyperparameter but structure parameter)
* Learn rate
* Regularization L1 and L2
* Output layer activation type
* Layer activation type
* Loss function type
* Epoch count

1) By changing these hyperparameters try to reach max accuracy value(at least 0.95) for Part2 model with fixed epoch count 20  
2) Change 1st hyperparameter’s value from min to max with minimal step depends on your variant   
3) Show impact on result using graphs  
4) Describe impact of each hyperparameter on accuracy.  
5) Set hyperparameter value back to one which produced max accuracy  
6) Repeat 2-5 steps for second hyperparameter

Make a report including:

* Each hyperparameter description and its impact on accuracy.
* Hyperparameters’ values which were used to reach accuracy value 0.95
* Graphs for these hyperparameters’ values