# Assignment 3

Implement a classifier based on a fully connected feed forward neural network (multi-layer perceptron), for hand written digits.

The classification algorithm must use a neural network of 3 layers(input: 784 neurons, hidden: 100, output: 10). The neurons from the hidden layer will use the sigmoid activation function. The neurons from the last layer can also use sigmoid, but softmax is recommended. The output of the neural network will be given by the neuron's number form the final layer with the greatest output value.

For maximum points, you have to:

- The neural network needs to be able to obtain at least 95% accuracy on a test set
- The used cost function must be cross-entropy
- The last layer has to use the softmax activation function
- Weights must be initialized in a proper manner to avoid saturation of the neurons.
- You use at least one of the following techniques:
    - L2 regularization + momentum
    - Dropout + any data augmentation
    - RMSProp

For training, use the MNIST dataset. This can be found, in a format that can be easily worked with in python, at the following url: http://deeplearning.net/data/mnist/mnist.pkl.gz

The dataset is split in 3 sets: training_set, validation_set, test_set. Each of these 3 sets contains two vectors of equal length:

1. A set of digits written as a vector of length 784. The digits from the mnist dataset have the shape 28x28 pixels and are represented as a vector ( each of the 28 lines from the 28x28 matrix are written one after each other, thus forming a vector of 784 elements). Each pixel from the matrix has a value between 0 and 1, where 0 represents white, 1 represents black and the value between 0 and 1 is a shade of grey.
2. A label for each element from the first vector: a number between 0 and 9 representing the digit from the image

The 3 sets have the following meaning:

- training_set (used for training your model); 50000 elements
- validation_set (usually used to adjust hyper-parameters and to perform a first evaluation of the resulted model); 10000 elements
- test_set (dataset used for testing. Use it only after you've fine-tuned the algorithm using the validation set. Do not use it for fine-tuning); 10000 elements

The dataset was saved using the cPickle python module. (it is very used for datasets serialization)

To load the dataset, use the following code.

```
import cPickle, gzip, numpy
f = gzip.open('mnist.pkl.gz', 'rb')
train_set, valid_set, test_set = cPickle.load(f)
f.close()
```