LAPPEENRANTA UNIVERSITY OF TECHNOLOGY
School of Engineering Science
Computational Engineering and Technical Physics

Documentation for practical assignment.

Course number: BM40A0701
Course name: Pattern Recognition
Project title: Digits 3-D
Date: 11.12.2017
Names,student numbers and roles of the authors in the project:
Margarita Pirumyan, 0524589, designing the idea, implementing code for neural network, documentation
Nikita Ashikhmin, 0524411, designing the idea, implementing code for preprocessing step, wrote testing part
Ivan Osipov, 0516131, designing the idea, wrote help for functions,wrote additional functions

Problem Statement.

Your task is to develop a learning system for hand-written digits 0, 1, ..., 9. The digits are “written” as free-hand strokes in the air with the index finger, and the 3-D location information of the finger tip has been read by using a LeapMotion sensor. The purpose of the system is to recognise the hand-written digits, that is, to classify them correctly based on the 3-D location time series.

Idea of the solution.

The main idea of the solution is that we reduce the problem of 3d digits recognition to 2d digits recognition. And it’s already known that this problem was solved with a good efficiency with the help of neural networks.

So, to recognize 3D digits we implemented 2 main steps:
1)Preprocessing step (3D to 2D)
2)Recognition of 2D digits using Neural Network with Backpropagation error algorithm

Implementation(Feature extraction and classification)

Preprocessing step (3D to 2D).

Input: 1000 samples of 3d vectors

For the first step we decided that 2dims of input data will be enough for the prediction, because this task can be reduced to the popular task of 2D digits recognition.

So, we took only 2 dims of each data sample, which are points in 2dims space and plotted them into figure, then we save them into the file system as jpeg images.

After this step we could work with sample as with image object. That‘s why further we implemented these steps:

- resizing (important to resize all objects to particular size for the reason that afterwards we need to give the images to the neural net. To train neural net we need a lot of samples in the same size)
- binarization
- reshaping (transforming array of pixels into the vector. Vector is an input for the neural net)

Output: 1000 samples of 2d digit images in the form of vector.

Vector consists of 0 and 1 values, it size is 1600(40x40) elements.

Recognition of 2D digits using Neural Network with Backpropagation error algorithm.

Input: 1000 samples of vectors of 1600 elements.

For the second step we used neural network with the Backpropagation algorithm.

Structure of the net:

There is only 1 hidden layer, but then it turned out that it's enough for recognition. There are 16 neurons in it and output layer consists of 10 neurons(because we need to recognize digits from 0..9).

Parameters:

Also we use sigma function as activation function because it gives the best results for training net.

Parametres for net:

```
l_rate = 0.1;   %learning coefficient
n_epoch = 200;
n_hidden = 16;  %num of neurons in hidden layer
a = 0.3;        %coeff for sigma function
```

There is a short scheme of our neural net structure on Figure1.

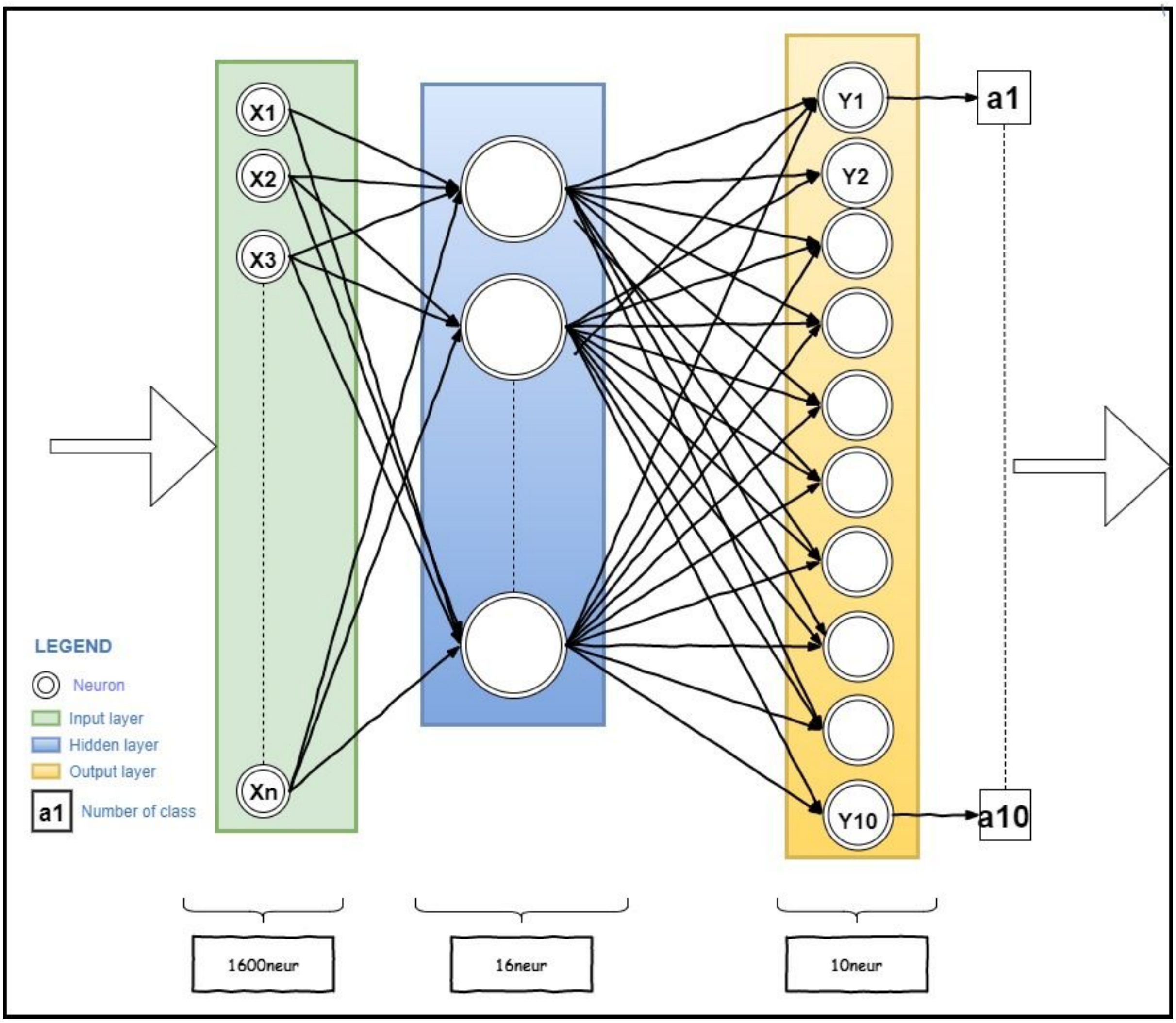

Figure1.

<u>Training of net consists from several steps:</u>

- initialization of variables
- setting the parameters for net
- building net and estimating the output and error of it for each sample (Forward step)
- Backpropagating step: we correct the weights of net based on the error values moving back on layers
- Continue to make last two steps until we reach 3 states:
  1)The error in the epoch will decrease significantly
  2)The number of max epochs is reached
  3)The difference between epochs doesn't change more than eps

<u>Testing of net:</u>

- It’s just the implementation of forward step to the new(test) data

Validation:

We tested our net with cross-validation approach(or leave-one-out approach).

Output: predicted digit number for each sample

Results

First testing of program.

Train data = 700 samples
Test data = 300 samples

Parametres for net:
l_rate = 0.1;   %learning coefficient
n_epoch = 200;
%only one hiden layer exists
n_hidden = 16; %num of neurons in hidden layer
a = 0.3;   %coeff for sigma function

```
Results for train data:
True predicted (%)=
  99.2857
False predicted (%)=
   0.7143
```

```
Results for test data:
True predicted (%)=
  90.3333
False predicted (%)=
   9.6667
```

Cross validation result (1000 samples, k_folds = 10):

Final True predicted (%)=

```
  90.1000

Final False predicted (%)=
    9.9000
```

References

1. Mathematical methods of learning by use of precedents (theory of machine learning) K.V. Vorontsov, 140pages http://www.machinelearning.ru/wiki/images/6/6d/Voron-ML-1.pdf