Neuronale Netze – Exercise 2 – Logistic Neurons

1.1 Goals

- Be familiar with Python and its scientific frameworks (`Numpy`, `Scipy`, `Matplotlib`, `Scikit-learn`..).
- Be familiar with the NN Praktikum framework design.
- Implement a classifier using **Logistic Neurons** (Logistic Regression) to recognize a handwritten digit whether it is a seven (7) or not.

1.2 Repository

- GitHub: `https://github.com/thanhleha-kit/NNPraktikum.git` (branch: Ex1, sample code: Ex2)

1.3 Data

1.3.1 MNIST - handwritten digit recognition dataset

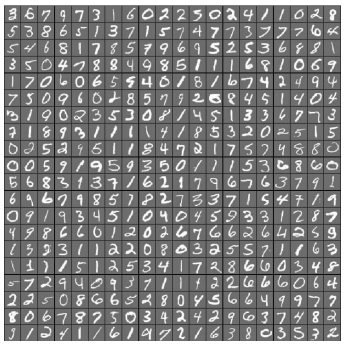

Figure 1: Example digits from MNIST dataset

- Grayscale images of handwritten digits (0–9), 28x28 pixels.
- 60000 images for training, 10000 for testing.

1.3.2 This MNIST subset

- 3000 images for training, 1000 for validation, 1000 for testing.
- In csv format, one row is one image (instance)
  - First field is the label of the digit (0–9)
  - Remaining fields (784 fields) are pixel values (0 means White, 255 means Black)

1.4 Scientific Python Packages

1.4.1 Numpy & Scipy

- `Numpy`: Numerical python library, useful in vector and matrix operations, matlab-like.
- `Scipy`: Scientific computing python library, contains `Numpy`.
- `http://www.numpy.org/andhttp://www.scipy.org/`
- Install: See `http://www.scipy.org/install.html`
- Good tutorial: `http://www.engr.ucsb.edu/~shell/che210d/numpy.pdf`

1.4.2 Matplotlib

- Python 2D plotting library.
- We will use a part of `Matplotlib`, called `Pylab`.
- `http://matplotlib.org/`
- Install: See `http://www.scipy.org/install.html`
- Good tutorial: `https://scipy-lectures.github.io/intro/matplotlib/matplotlib.html`

1.5 Scikit-learn

- A Python Machine Learning framework, easy-to-use and covers full of ML algorithms.
- We will use it for comparison purpose, doing evaluation and printing performance.
- `http://scikit-learn.org/stable/`
- Install: `http://scikit-learn.org/dev/install.html`
- Tutorial: `http://scikit-learn.org/stable/tutorial/index.html`

1.6 Your coding tasks

- Install python and required scientific packages (numpy, scipy, matplotlib, scikit-learn).
- Check out or download the data and framework (if you haven't done it yet).
- Complete the code for an asbtract Logistic Neuron layer (will be useful for next exercises).
- Complete the code for training, classifying and evaluating of the 1-Logistic-Neuron classifier.
- Run your classifier implementation in `Run.py`.
- Bonus:
  - Try different learning rates and (unofficially) report about the performance
  - Try different "epochs" value and (unofficially) report about the performance
  - Use matplotlib to draw a graph of performance on different epochs

1.7 Notes

- Your classification will be a very simple layer with one Logistic neuron taking $784 + 1$ input values and produce 1 output value.
- But it still needs to conform the general abstract layer design (n output values = n Logistic neurons).
- The output is the probability of "being a Seven". And the final decision (classify method) would be based on that.