Deep Learning

Hands-on computer session

Prerequisites

The hands-on computer session is based on the Keras library (which is based on python and the theano frameworks). It is recommended to the participant to install the library before the session.
Installation instruction:

- Theano
- Keras

Installation on linux (one possible way)

1. Install Python (>2.7) and numpy
2. Install Python-dev
3. Install Matplotlib
4. Install g++
5. Install scipy(from package repository)
6. Install pip
7. run : sudo pip install keras

Installation on windows (one possible way)

1. Install anaconda
2. run (in the command prompt) :
    a. pip install Theano
    b. conda install pydot
    c. pip install keras

Part I MNIST database

In a first time, we will consider the MNIST database which contains 60,000 28x28 grayscale images of the 10 digits, along with a test set of 10,000 images.
The file mnist.py contains a basic code for learning a neural network on this base :

- ***load*** which loads the data, splits them onto train and test part
- ***reshapeData*** which reshapes the data to the neural network format
- **plot_10_by_10_images** shows a sample of images (the images have to be in the initial format)
- **plot_mnist_digit** shows an image
- **plot_load_digit** loads an image form a file

- **predictionFormat** transforms the loaded image in the neural networks prediction format
- ***learnModel*** which learns a basic model
- ***evaluate*** which evaluate the models on the test set

In the learn model function, the neural networks shape is obtained by using the keras functions

- ***model.add(Dense(nbhiden, [input_shape=(nbinput,)], init='uniform')) :*** which adds an hidden layer to the network. nbhidden is the number of hidden nodes. [input_shape=(nbinput,)] corresponds to the input layer
- ***model.add(Activation(activationfunction)) :*** activation function of the layer. See here for the available activation functions
- ***model.add(Dropout(value)) :*** apply dropout to the layer (to avoid overfitting)

Questions

1. Describe the MNIST problem (classification ? regression ? numbers of inputs ? output ?)
2. Use the function plot_10_by_10_images to shows a set of examples
3. Describe the neural networks in the learnModel function
4. Learn and evaluate the proposed neural networks. Analyse the result.
5. Propose and apply your own neural networks. Try to achieve the best results as possible
6. Display the images that are predicted in the wrong class by your model
7. In an image editor, draw a digit try to recognize it with your model (use the function predict)

Part II IMDB database

We now consider the IMDB databse and the imdb.py file
Questions

1. Describe the problem
2. Describe the proposed neural networks
3. Since the data is too large for this session, make a function for computing a subsample of the training set
4. Learn, tune, and evaluate the model

Bonus

Apply the previous approach to the Reuters database (use the keras documentation).