Exercise Session: Machine Learning in Power Systems

Smart Distribution Systems (B-KUL-H00P3A)
Prof. Geert Deconinck

March 23, 2017

Description

This exercise session will familiarize the students with basic concepts from machine learning and teach the students how these concepts can be used within power systems for *forecasting* and *clustering*.

The students will learn to:

- set up a machine learning environment, using state-of-the-art tools, such as keras, tensorflow and theano in Python;
- implement and train a two-layer neural network using Keras;
- use this neural network to make prediction about the wholesale electricity price.

1 Introduction to neural networks

here we will explain them some basic concepts about NN

- different activation functions
- back propagation

2 Setting up a machine learning environment

here we will explain how the student can setup all tools on Linux Ubuntu 16.04

- Python 2.7
- Keras (Tensorflow backend)
- matplotlib
- Jupyter notebook [Jup]

To do: check with Veronica how we can automate this procedure so student don't lose too much time and replicate it on their own computers.

Installation script created, details described in appendix. Doubt: possible to get sudo access in the computer rooms?

We will also make a Github account for the course so student can clone our repostory with data and additional examples.

3 Exercises

3.1 Data analysis of the data set (1h)

here the student will have to *visualize* the data set

The data set is a pandas python structure containing wind, solar, temperature and wholesale electricity prices for Belgium from 2013 tot 2016.

3.2 Implement your first neural network (1h)

we will show the students how to implement a two layer neural network.

3.3 Scenario reduction (?)

- Use clustering to reduce the dataset to a smaller number of scenarios
- This could be used in the later session on optimization as an input for a stochastic optimization, where the weights of the scenarios are derived from the number of states in the given clusters
- If too difficult, then just mention the stochastic optimization as an application

4 Final report

The data set for the final report is split in two parts: a training set and validation set. The students only receive the training set and will have to implement their own strategy for forecasting the electricity price.

After two weeks the students have to submit their neural network. The course assistants will validate their model using the validation set and make a ranking based a mean-squared error metric. A final grade will be provided depending on this ranking.

The students have to:

- team up and form groups of maximum 4 students
- submit a report of no more than 5 pages (including plots) explaining their forecasting strategy and visualization of the data set (pdf format). It is recommended that students use the following overleaf template:
  - https://www.overleaf.com/8396605zwdzvdrygrvf#/29773845/
- submit the model containing their neural network + weights (in Pickle format)
- submit the report and data set to a dropbox link on Toledo (B-KUL-H00P3A). The final day for submission is
  - **Final date (two weeks after exercise session)**.

5 Feedback

Please provide feedback on this course session so we can keep improving the format (link to surveymonkey)

The final grading (ranking) will be made available on Toledo two weeks after submission.

Have fun!

A Installation procedure

A prerequisite is that you have Python installed. Most Linux distributions come with Python installed.
The commands described below can be executed by opening a terminal (try the shortcut ctrl-alt-T)

A.1 Cloning the repository

We clone the GitHub repository of this exercise session to get all the files we need. 'Cloning' simply copies a given repository to a local folder. This is done by executing the following command, which will copy the repository to your desktop.

```
git clone <INSERT GITHUB URL> ~/Desktop
```

A.2 Installation

There are two ways to set up your machine. You can either execute the installation script, or go through the steps manually.

A.2.1 Automatic

The repository you just cloned, contains an installation script. First, you will have to make it executable.

```
sudo chown +x ~/Desktop/sds/install_script.sh
```

Now you can execute it to take care of the installation for you.

```
~/Desktop/sds/install_script.sh
```

A.2.2 Manual

First, we will create a virtualenv. Virtualenv provides you with a virtual environment for Python, allowing you to install the exact dependencies you want for a certain application. The following commands install virtualenv, create a virtual environment named 'sds' and activate this environment.

```
sudo apt-get install python-pip python-dev python-virtualenv
virtualenv --system-site-packages ~/sds
source ~/sds/bin/activate
```

Next, we install tensorflow, a library created by Google for numerical computations in the context of machine learning [Ten].

```
pip install --upgrade tensorflow
```

Furthermore, we install keras, a high-level neural networks library [Ker]. This library runs on top of tensorflow, which we installed before.

```
sudo pip install keras
```

Finally, we install jupyter [Jup].

```
sudo pip install jupyter
```

References

[Jup] Jupyter Notebook iPython. A simple way to share jupyter notebooks. https://nbviewer.jupyter.org/github/ipython/ipython/blob/4.0.x/examples/IPython

[Ker] Keras. Deep learning library for theano and tensorflow. https://keras.io/.

[Ten] TensorFlow. An open-source software library for machine intelligence. https://www.tensorflow.org/.