Setting up your working environment
Python, Anaconda, Libraries
Several Examples as a First Foot in the Door!

EECE-6822 Machine Learning

Cris Ababei
Electrical and Computer Engr., Marquette University

1. Objective

The objectives of this activity include: (1) install Python on your laptop (i.e., Anaconda + Spyder) as well as several libraries; (2) go through some tutorials on first simple ML models (Keras TensorFlow and Pytorch). This will expose you to some initial concepts, which will be revisited later; and (3) go through some more tutorials as review of some basic probability, random variables. Note, that some of the concepts encountered here will be discussed in more detail later in the course; in this way we get repeated exposure and make use multiple times of techniques, in the spirit that "repetition is the mother of learning".

2. Prerequisite Readings

Murphy
- Probability review: Murphy 2.1-2.4, 2.6, 2.8, 3.1-3.2
- Statistics review, maximum likelihood: Murphy 4.2

Mostafa
- The learning problem: Ch. 1

Geron – Preface, Ch. 1
- General info on machine learning
- Examples of application
- Three types of ML
- A first simple example of LinearRegression class from SciKit-Learn

Raschka – Preface, Ch.1
- Types of ML (see pp. 3), with very nice figures
- Notational conventions

3. Python

In this course, we will use quite a few different software technologies. We need to install them in order to set-up a working environment that has all that is needed to work on examples and projects. It is required that you have your own laptop/computer. The main steps described here are for my specific case – a windows laptop.

If you are a Mac or Linux machine user, you may have an easier task setting-up all this – however, it will be up to you to figure that out. Do not expect detailed help from the instructor on aspects related to installing stuff on your laptop or on low-level debug of your code. If some of the online tutorials or examples are a bit outdated, it is expected that students will figure out how to make them work with the latest versions of IDEs and libraries. Please take ownership of the little "devil in the details" challenges and find workarounds!

We will use mostly Python. To execute Python code you can use Google Colab in the cloud or you can do it locally on your machine, using an IDE such as Anaconda Spyder (or anything else you may be familiar with already). We will use both working modes depending on the project.

The Colab approach is easy and convenient, but, requires an Internet connection at all times. What is Colab? Colab, or "Colaboratory", allows you to write and execute Python in your browser, with: Zero configuration required, access to GPUs free of charge, and easy sharing. One can run Jupyter Notebooks, which is very nice. *Jupyter Notebook* is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

To run Python programs on your local machine, you must install something like Anaconda. To set-up a Python development environment for machine learning and deep learning on your Windows machine, follow the following tutorial to install all you need:
https://machinelearningmastery.com/setup-python-environment-machine-learning-deep-learning-anaconda/

You must do that now in order to be able to continue with the examples below.

4. Prerequisites – Refreshing your Python

There are plenty of resources out there regarding Python coding and related math.
Two examples of nice collections of such resources were discussed in class.

Take some time to familiarize yourself with those so that you will know what's available and thus refer to later on as needed. Particularly, go through the Python Intro notebook:
readings/Python_Intro.ipynb
provided in the .ZIP file for this lecture.

5. Examples

Example 1: Keras TensorFlow

As part of the hands-on activities this week, you must do the following Python tutorials from mlm:

Tutorial 1: Your First Machine Learning Project in Python Step-By-Step
https://machinelearningmastery.com/machine-learning-in-python-step-by-step/

Tutorial 2: Your First Deep Learning Project in Python with Keras Step-by-Step
https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/

Example 2: Pytorch

In this example, we develop a Multilayer Perceptron (MLP) model in Pytorch, which does binary classification. It also requires to install Pytorch.

Tutorial 3: PyTorch Tutorial: How to Develop Deep Learning Models with Python
https://machinelearningmastery.com/pytorch-tutorial-develop-deep-learning-models

Then, you must use this example to develop a Pytorch version of the best model found through the exploration from Example 2. So, you essentially must implement in Pytorch the same "best" model discovered before and compare results obtained with the models in Keras + TF vs. Pytorch!

6. Assignment

You must re-do the second tutorial from Example #1 above. You must investigate several different network architectures and different number of units per layer in order to find out if the number of layers or the number of units per layer can be used to improve model performance:

- In the first phase of this experiment, you must study the dataset. To do that you must generate the scatter matrix plot and discuss it.
- In the second phase of this experiment, you shall investigate **four** network architectures with a number of layers = {3,4,5,6} (note that the case of 3 layers is the current example itself) and a number of 8 units on the middle layers.
- In the third phase of this experiment, you take the best model from the previous phase and investigate its performance for a number of units on middle layers = {4,8,16,24}.

7. Deliverables

You must write (**typed**) a report and upload it as a PDF file on D2L. The report should be named "**hw1_report_LastName.pdf**". You should also create a .zip archive with all your code and implementations of all parts of the assignment. Upload also this archive .zip file with the name "**hw1_implementation_code_LastName.zip**" to D2L. Hence, your D2L should contain two items: the report and the .zip file. **Do not include the report inside the .zip and upload only the .zip. They should be two separate items!**

The report should include the following sections and subsections:

1) Title + course info + your name
2) **Summary.** Describe in one paragraph what the objective of the assignment is.
3) **Part 1:**
    a. **Dataset Study.** Describe what a scatter matrix plot is. Include the one you generated and discuss what it tells you.
    b. **Architecture Exploration.** Describe the steps in conducting your experiments. You must create and include two plots in two separate figures: one that shows performance (i.e., model accuracy) as function of number of layers and one that shows performance as function of number of units per layer. Figures must be numbered and have captions!
4) **Part 2:**
    a. **Model Description.** Describe the Pytorch model you developed to achieve the same objective as the model from Example 2. Compare results and present a discussion of this comparison.
5) **Conclusion.** Present your conclusions and describe what issues you encountered and how you solved them.
6) **References.** Include all references that you used, as a numbered list. Cite them in the report itself; do not just list them! If your report has References that are not cited in the report, points will be deducted!