Uni.lu HPC School 2018

PS13: Machine and Deep learning workflows on UL HPC platforms

Uni.lu High Performance Computing (HPC) Team

Dr. S. Varrette

University of Luxembourg (UL), Luxembourg

http://hpc.uni.lu

Latest versions available on Github:

UL HPC tutorials:	https://github.com/ULHPC/tutorials
UL HPC School:	http://hpc.uni.lu/hpc-school/
PS13 tutorial sources:	ulhpc-tutorials.rtfd.io/en/latest/deep_learning/

Summary

1 **Practical Session Objectives**

2 Deep Learning Analytics with Tensorflow

3 Training Neural Networks with Keras

Main Objectives of this Session

- (ultra brief) **review of Machine/Deep Learning**
  - ↪ using (excellent) material of

  `https://github.com/m2dsupsdlclass/lectures-labs`
- Preliminary installations
  - ↪ X11 forwarding and SOCKS5 proxy connection over SSH
  - ↪ Prepare virtualenv
  - ↪ Install Jupyter
  - ↪ Configure Jupiter Kernel and launch jupyter notebook
- MNIST ML and Deep ML using Tensorflow
- Training Neural Networks with Keras
  - ↪ Pretrained Models for Computer Vision

Hands-on 1: Preliminary installations

*We will rely on Python virtualenv - assumes you have followed PS9: Advanced Prototyping with Python

Your Turn!

Hands-on 1

ulhpc-tutorials.rtfd.io/en/latest/deep_learning/#1-preliminary-installations

- SSH Connect with X11 forwarding and SOCKS5 proxy connection
- Prepare virtualenv
- Installation of Jupyter Notebook
- Configure Jupiter Kernel and launch jupyter notebook

Summary

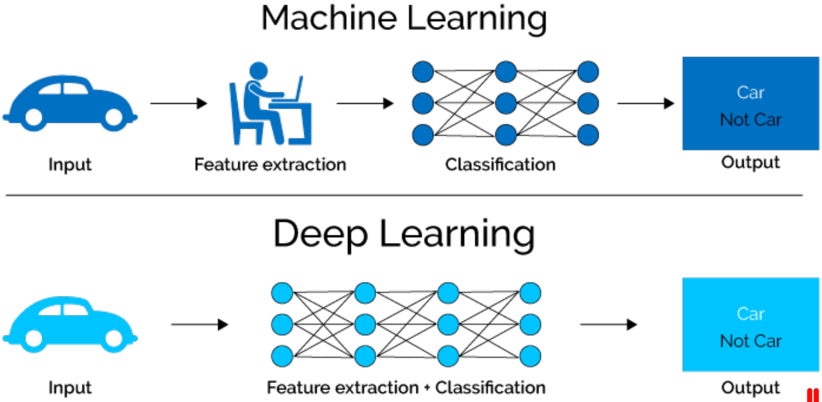
1 Practical Session Objectives

2 **Deep Learning Analytics with Tensorflow**

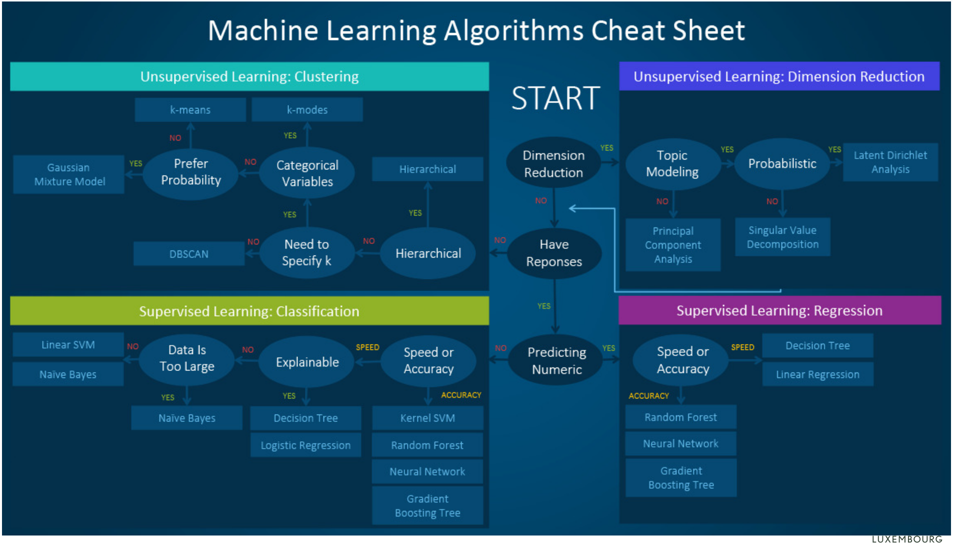
3 Training Neural Networks with Keras

Big data and Machine/Deep Learning

- **Out-of-scope of this tutorial**:
  ↪ Machine Learning (ML) / Deep Learning theoretical basis



Machine Learning Cheat sheet



Machine/Deep-Learning Frameworks

- Pytorch
  - ↪ Python version of Torch open-sourced by Facebook in 2017.
  - ↪ Torch is a computational framework with an API written in Lua that supports machine-learning algorithms.
  - ↪ PyTorch offers dynamic computation graphs, which let you process variable-length inputs and outputs.
- TensorFlow
  - ↪ open source software library from Google for numerical computation using data flow graphs,
  - ↪ thus close to the Deep Learning book way of thinking about neural networks.
- Keras,
  - ↪ high-level neural networks API,
  - ↪ written in Python and capable of running on top of TensorFlow.
- Caffee
  - ↪ a well-known and widely used machine-vision library that ported Matlabs implementation of fast convolutional nets to C and C++.
  - ↪ You'll also have to consider its successor, Caffee 2,

Machine/Deep-Learning Frameworks

- Offer various **Package Design Choices**
  - ↪ **Model specification**:
    - ✓ Configuration file (Caffe, DistBelief, CNTK) vs. programmatic generation (Torch, Theano, Tensorflow)
  - ↪ For programmatic models, choice of high-level language:
    - ✓ Lua (Torch)
    - ✓ vs. Python (Theano, Tensorflow)
    - ✓ vs others (Go etc.)

In this talk

- We chose to work with python because of rich community and library infrastructure.

TensorFlow vs. Theano

- Theano is another deep-learning library with pythonwrapper
  - ↪ was inspiration for Tensorflow
- Theano and TensorFlow are very similar systems.
  - ↪ TensorFlow has better support for distributed systems though,
  - ↪ development funded by Google, while Theano is an academic project.

What is TensorFlow ?

- TensorFlow is a deep learning library recently open-sourced by Google.
  - ↪ library for numerical computation using **data flow graphs**.
    - ✓ **Nodes** represent mathematical operations,
    - ✓ **edges** represent the multidimensional data arrays (**tensors**) communicated between them.
- Flexible architecture allowing to deploy computation anywhere:
  - ↪ to one or more CPUs or GPUs in a desktop, server,
  - ↪ or mobile device with a single API.
- TensorFlow was originally developed within the Google Brain Team

Hands-on 2: Tensorflow

- you are ready to play with tensorflow
- provided tutorial is self-explicit and make use of Jupyter Notebook

Hands-on 2

ulhpc-tutorials.rtfd.io/en/latest/deep_learning/#2-mnist-machine-learning-ml-and-deep-ml-using-tensorflo

- Run a very simple MNIST classifier **Step 1**
  ↪ MNIST: computer vision dataset (images of handwritten digits)
- Run a deep MNIST classifier using convolutional layers **Step 2**
  ↪ compare results with best models

Summary

1 Practical Session Objectives

2 Deep Learning Analytics with Tensorflow

3 Training Neural Networks with Keras

Hands-on 3: Keras

- you are ready to play with Keras
- provided tutorial is self-explicit and make use of Jupyter Notebook of **EXCELLENT** labs available at

https://github.com/m2dsupsdlclass/lectures-labs

Hands-on 3

`ulhpc-tutorials.rtfd.io/en/latest/deep_learning/#3-training-neural-networks-with-keras`

- Install keras
- Training Neural Networks with Keras
- Pretrained Models for Computer Vision

Thank you for your attention...

Questions?

http://hpc.uni.lu

High Performance Computing @ uni.lu

Prof. Pascal Bouvry
Dr. Sebastien Varrette
Valentin Plugaru
Sarah Peter
Hyacinthe Cartiaux
Clement Parisot

University of Luxembourg, Belval Campus
Maison du Nombre, 4th floor
2, avenue de l'Université
L-4365 Esch-sur-Alzette
mail: hpc@uni.lu

1. **Practical Session Objectives**
2. **Deep Learning Analytics with Tensorflow**
3. **Training Neural Networks with Keras**