Ph.D. Course –Advanced Topics in Machine Learning – Deep Learning

August 24-28, 2015, DTU Compute, Section for Cognitive Systems

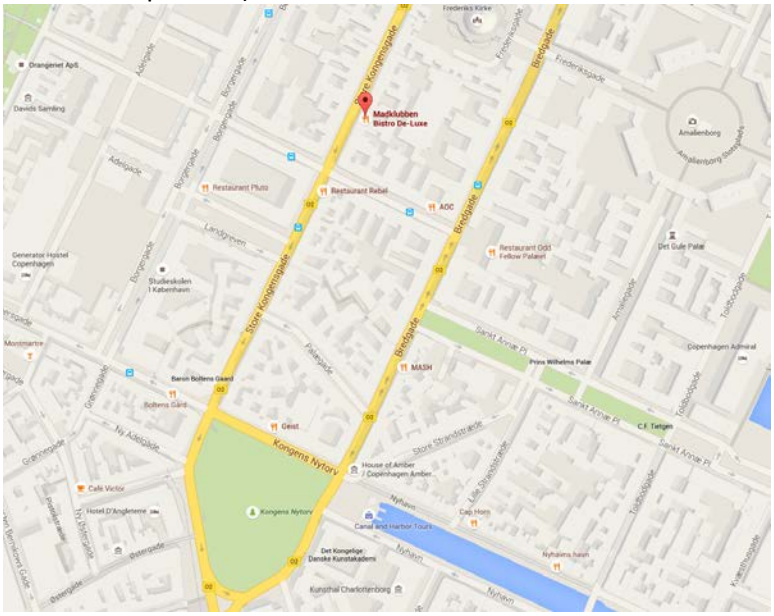
Course Program

Lectures will take place in building 324 auditorium 040 and 060 Monday-Friday from 9:00 AM – 4:40 PM (*click here for a map of the DTU campus*)

	Monday	Tuesday	Wednesday	Thursday	Friday
9:00-12:10 9:00 AM-12:10 AM	Lars Kai Hansen *Introduction to Neural Networks*	Bernd Dammann *Introduction to GPU programming* Anders Boesen Lindbo Larsen *Convnets: Power tools for your computer vision toolbox*	Ole Winther *Recurrent Neural Networks*	Charles Blundell *Probability & Neural Networks*	Tapani Raiko *Combining Supervised and Unsupervised Learning (and the Ladder Network)*
12:10-13:30 12:10 AM-1:30 PM	Lunch	Lunch	Lunch	Lunch	Lunch
13:30-16:40 1:30 PM-4:40 PM	Lars Kai Hansen *Introduction to Neural Networks*	Anders Boesen Lindbo Larsen *Convnets: Power tools for your computer vision toolbox*	Ole Winther *Recurrent Neural Networks*	Charles Blundell *Probability & Neural Networks*	Tapani Raiko *Combining Supervised and Unsupervised Learning (and the Ladder Network)* Toke Jansen *Machine learning for image editing*
			18:00 Dinner Madklubben *(Store Kongensgade 66, Copenhagen)*		

For details regarding transportation and accommodation, see also the practical information at the course homepage (http://www.compute.dtu.dk/courses/02901).

The course includes a dinner Wednesday evening at 6:00 pm at the restaurant Madklubben (http://www.madklubben.dk/bistro-de-luxe/) located in downtown Copenhagen at Store Kongensgade 66 close to Kongens Nytorv (see also the map below).



Please bring your own laptop computer at the course!
The course will use Python, Theano and Lasagne. We highly recommend you complete the installation prerequisites in the following pages before the course.

Installation Prerequisites

1. Introduction

Python (release 2.7.x) will be used as the programming language for this course. It is a powerful high-level programming language with the ability to program object-oriented using a vast amount of scientific libraries. If you are unfamiliar with Python then please refer to one of the tutorials[1] to get started.

The libraries that will be used mostly for this course are NumPy (`http://www.numpy.org`), Theano (`https://theano.readthedocs.org/en/latest/`), Lasagne (`http://lasagne.readthedocs.org/en/latest/`), and matplotlib (`http://matplotlib.org`).

- NumPy is a widely used scientific computing library for implementing multi-dimensional array objects, linear algebra functions, and generating random numbers.
- Theano is build upon Numpy to evaluate linear algebra functions on GPU (extreme speedups) and efficient symbolic differentiation.
- Lasagne is build upon Theano with the purpose of implementing deep learning models more efficiently.
- matplotlib is a plotting library.

We strongly encourage you to look into these libraries in order to understand how to implement basic linear algebraic expressions and neural networks[2].

2. Installing Python

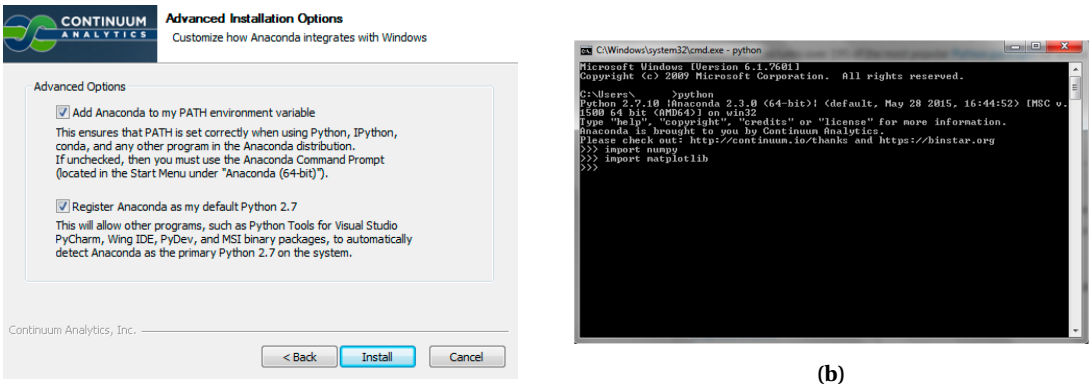
Anaconda (`https://store.continuum.io/cshop/anaconda/`) is a scientific Python distribution bundling all the essential Python libraries into one installer file.

- Download the Anaconda bundle for Python 2.7 from `http://continuum.io/downloads`.
- Execute the installer and make sure that you tick each check box (cf. Fig. 1a).
- Python and the essential libraries are now installed and you can check this by starting a terminal and importing (cf. Fig. 1b).

The Anaconda distribution comes with an IDE (Spyder), that enables code completion, project files and much more. Start Spyder by searching through your programs/applications folder and execute the executable (cf. Fig. 2).

[1] e.g. `https://docs.python.org/2/tutorial/`

[2] See `http://lasagne.readthedocs.org/en/latest/user/tutorial.html` to get a hands-on on how to implement a basic multi-layered perceptron model.



(a) **(b)**

Figure 1: *While installing the Anaconda distribution, make sure that each check box is ticked, since this will enable the ability to call Python from a terminal.*

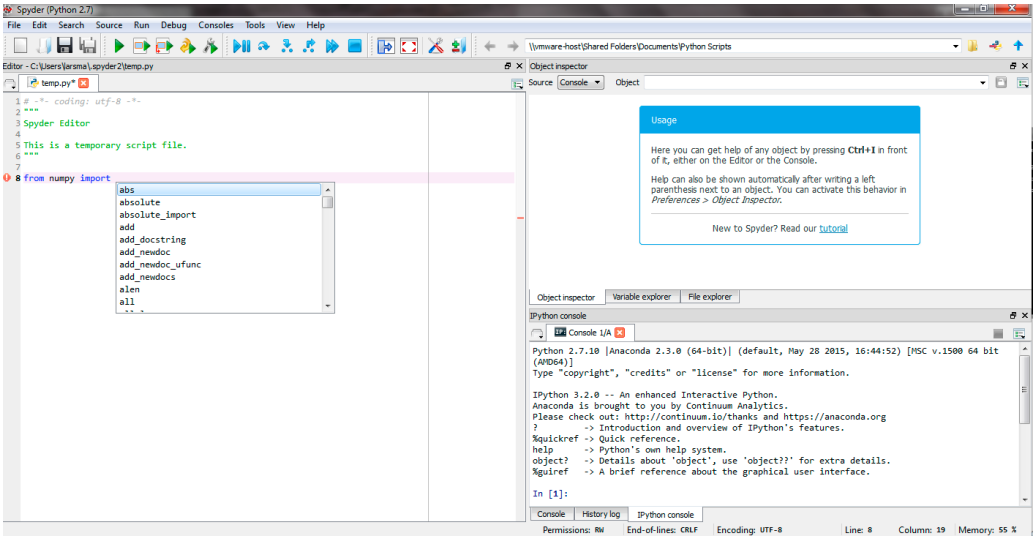


Figure 2: *Spyder IDE helps you structuring your code into a project and code completion.*

3. Installing Theano

The prerequisites for installing Theano can be tricky. To install this library we refer to the official installation guide (`http://deeplearning.net/software/theano/install.html`).

To check whether Theano is installed correctly, follow the steps to import Theano with the complementary tensor module. A good toy example to get started is to implement a simple logistic regression model (cf. `http://deeplearning.net/software/theano/tutorial/examples.html` and Fig. 3).

log_reg.py

```
import numpy
import theano
import theano.tensor as T
rng = numpy.random
N = 400
feats = 784
D = (rng.randn(N, feats), rng.randint(size=N, low=0, high=2))
training_steps = 10000
# Declare Theano symbolic variables
x = T.matrix("x")
y = T.vector("y")
w = theano.shared(rng.randn(feats), name="w")
b = theano.shared(0., name="b")
print "Initial model:"
print w.get_value(), b.get_value()
# Construct Theano expression graph
p_1 = 1 / (1 + T.exp(-T.dot(x, w) - b))   # Probability that target = 1
prediction = p_1 > 0.5                    # The prediction thresholded
xent = -y * T.log(p_1) - (1-y) * T.log(1-p_1) # Cross-entropy loss function
cost = xent.mean() + 0.01 * (w ** 2).sum()# The cost to minimize
gw, gb = T.grad(cost, [w, b])             # Compute the gradient of the cost
                                          # (we shall return to this in a
                                          # following section of this tutorial)
# Compile
train = theano.function(
          inputs=[x,y],
          outputs=[prediction, xent],
          updates=((w, w - 0.1 * gw), (b, b - 0.1 * gb)))
predict = theano.function(inputs=[x], outputs=prediction)
# Train
for i in range(training_steps):
    pred, err = train(D[0], D[1])

print "Final model:"
print w.get_value(), b.get_value()
print "target values for D:", D[1]
print "prediction on D:", predict(D[0])
```

Figure 3: *Importing the Theano modules and implementing a simple logistic regression example.*

4. INSTALLING LASAGNE

When Theano is installed properly, the installation of Lasagne is quite simple (cf. `http://lasagne.readthedocs.org/en/latest/user/installation.html`). Test that Lasagne is installed properly by importing it: *import lasagne*.