# Lab 10: Deep Learning Tutorial 1

By Sam Liu, last modified Ray Huang on 2020.04.26

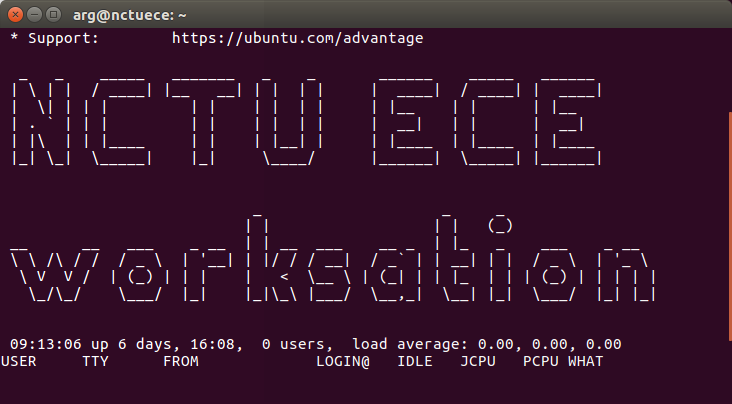
The objective of this tutorial is to introduce to the basic concept of DL, include the MLP model, forward/backward propagation. We also prepared some example code for basic Pytorch libraries in Jupyter notebook.

## Hardware and Software Setup

Access GPU machine with ssh

**laptop $ ssh [username]@140.113.148.xxx**

type the password then you will see like this

-

**ws : workstation**

make your own working directory

**ws $ mkdir [whatever]**

Clone the course repository

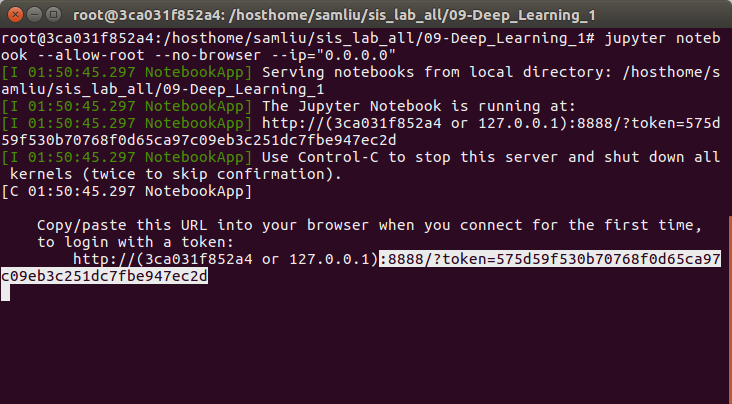
**ws $ cd [whatever]  
ws $ git clone https://github.com/Sensing-Intelligent-System/sis\_lab\_all\_2020**

**ws $ cd sis\_lab\_all\_2020 && git pull origin master**

Run the docker and jupyter notebook

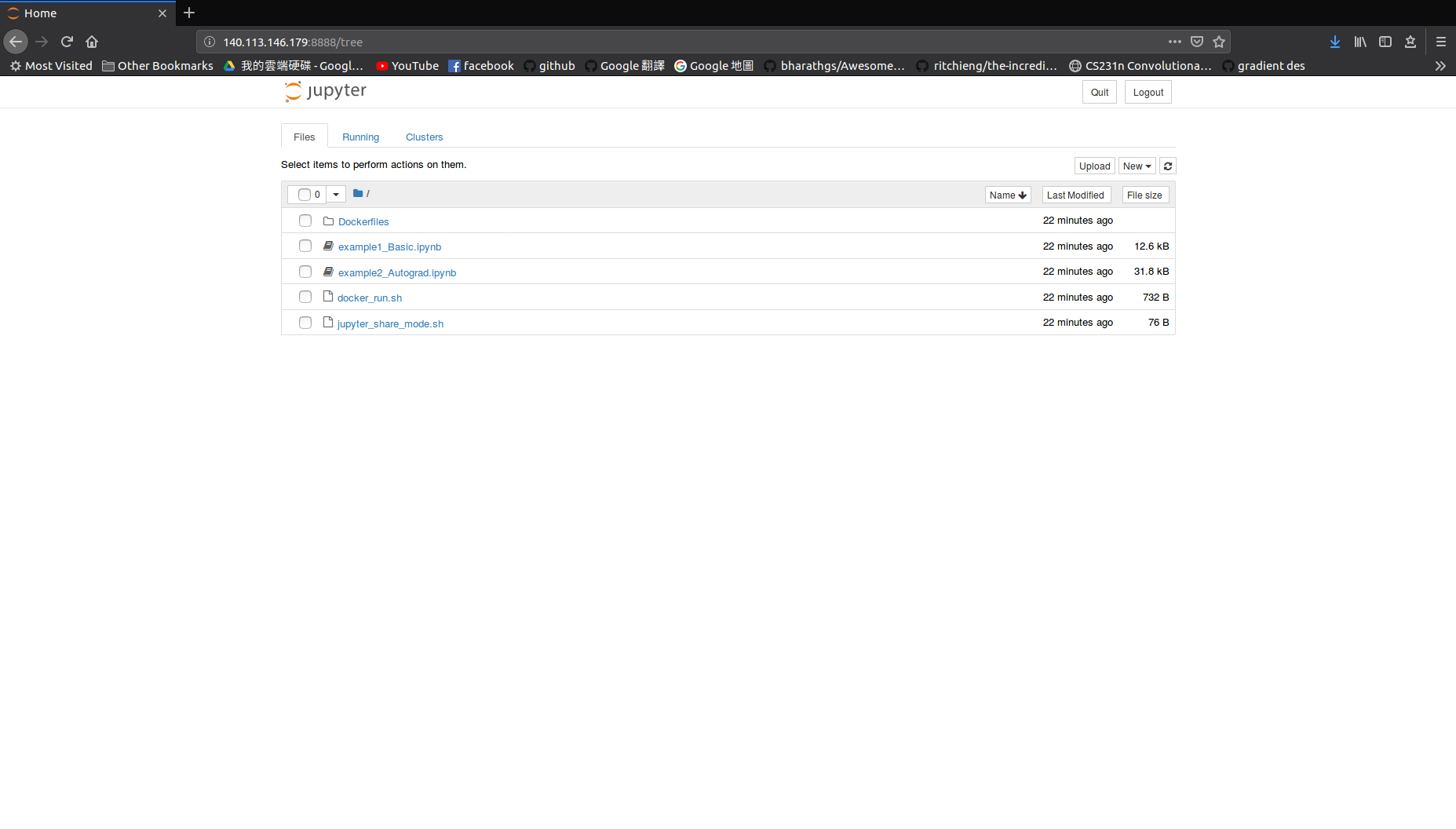
**ws $ source docker\_run.sh**

**container # source jupyter\_no\_broswer.sh**

****

Turn on the web browser and type the **workstation’s IP** and **token** from above

You will see like this



[**Lab 10: Deep Learning Tutorial 1**](#_74gram21yqax) **1**

[Hardware and Software Setup](#_chc57ow4x0s1) 1

[Overview](#_x5puhc1p2jjo) 2

[Usage of jupyter notebook](#_hqm01wfyfbgo) 3

[Topics and Activities](#_g0fw0bgwxhdt) 4

[Topic/Activity 1 Pytorch basic library](#_aadm982kyolo) 4

[Topic 1.1 Pytorch Tensor](#_rl15slrm10vt) 4

[Topic 1.2 Pytorch Autograd](#_9xf9nsimldmi) 4

[Topic/Activity 2 Network architecture](#_sf5mhl8ztpdm) 4

[Topic 2.2 Pytorch Neural Network](#_v4sjn3hb2lab) 4

[Topic 2.2 Pytorch Multilayer Perceptron](#_dlktwhuxe8nr) 4

[Reference](#_uwwglharlfb) 5

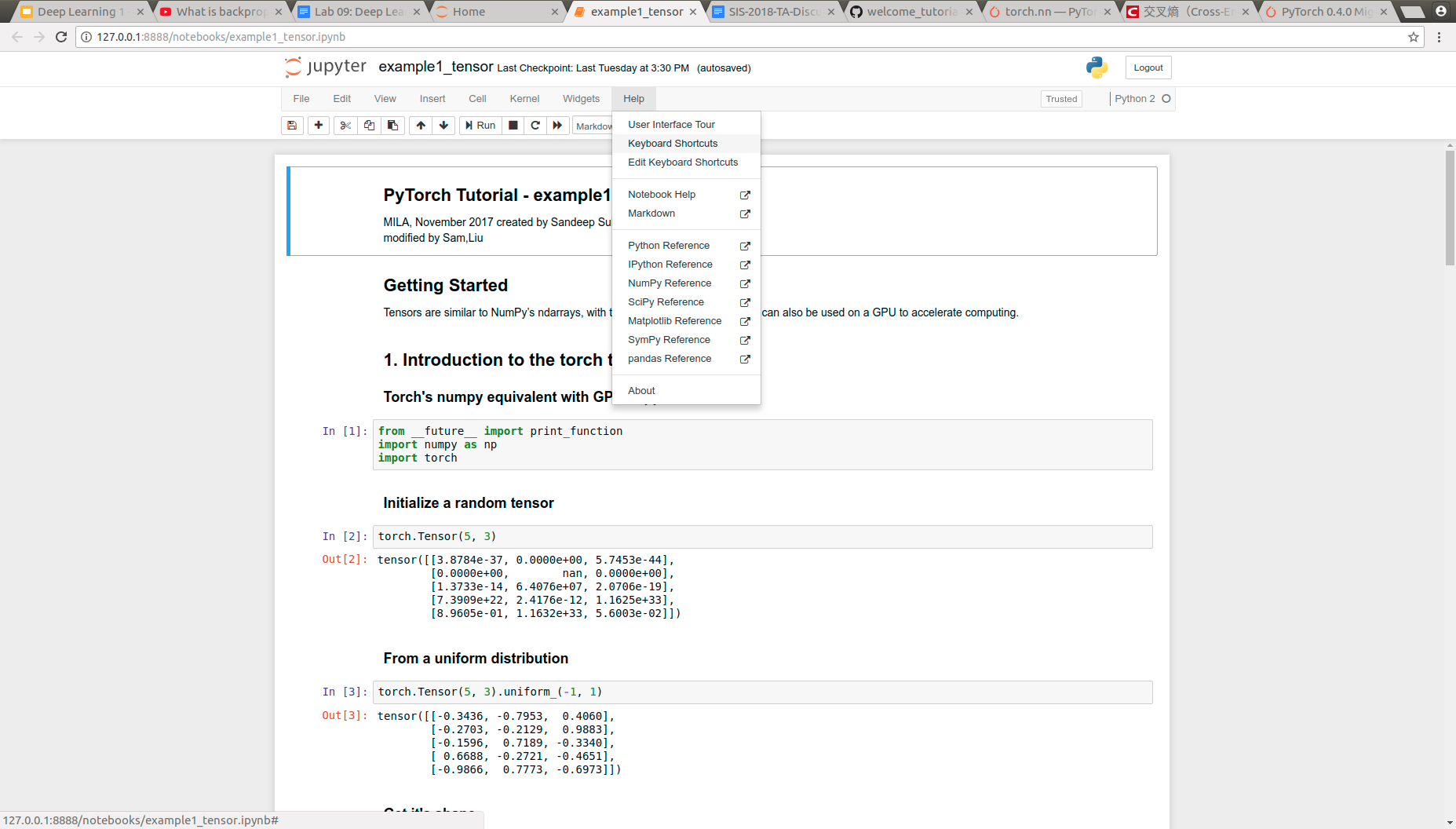
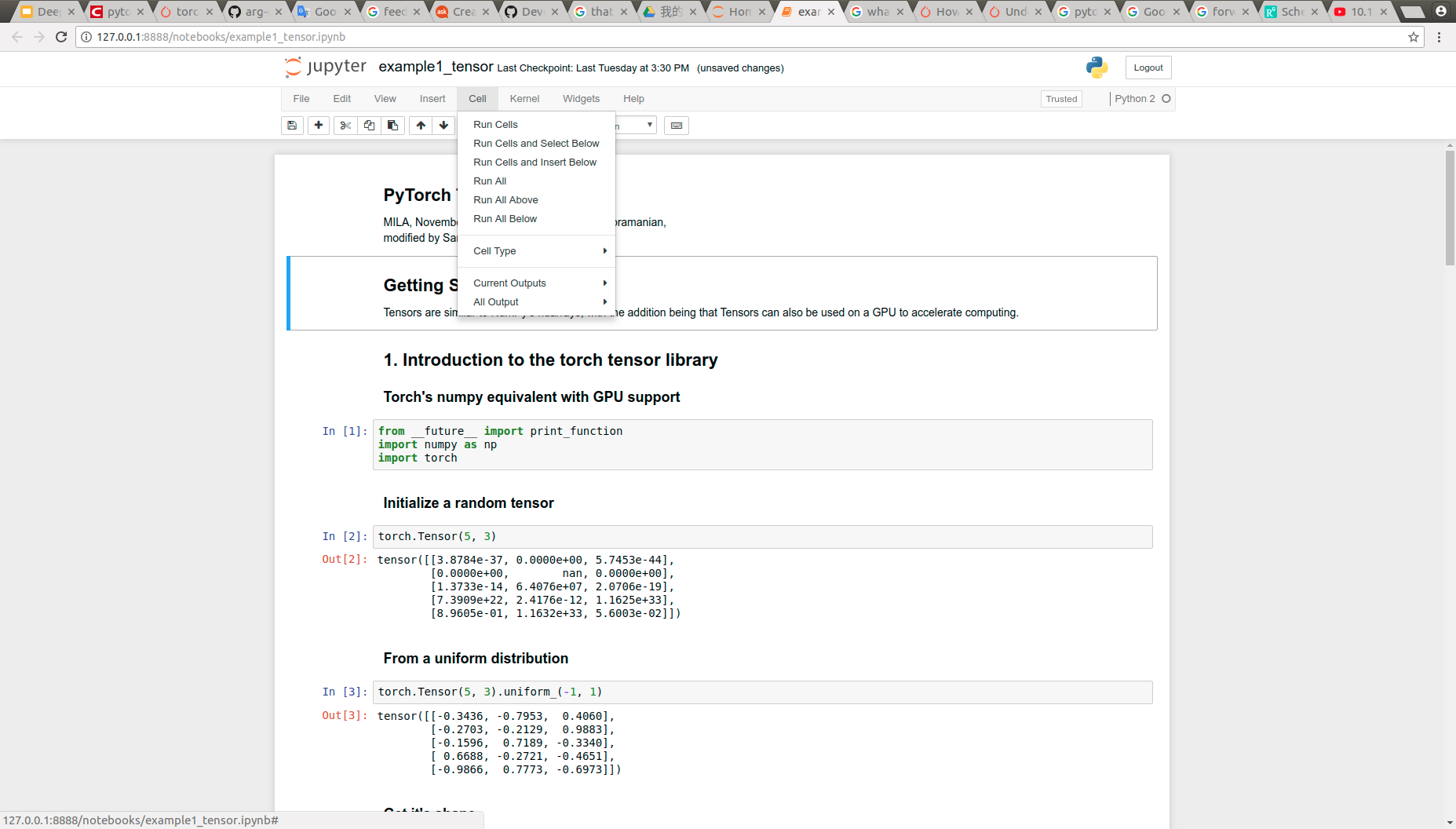
## Overview

Estimated Time to Finish: 1.5 hours

After completing this tutorial you should

* understand the basic deep learning architecture, concept of forward/backward propagation.
* understand how to use Pytorch module to implement the regression model.

## Usage of jupyter notebook

1. Keyboard shortcuts: (Help → Keyboard shortcuts)  
   
2. Run cell: [Shift+Enter]
3. 

## Topics and Activities

### Topic/Activity 1 Pytorch basic library

Pytorch provides many convenient module like Tensor, nn, optim for creating a neural network quickly. The most important module about data computation in pytorch is [torch.Tensor](https://pytorch.org/docs/stable/tensors.html). All data forwarded into/output from neural network is shaped into ‘Tensor’ object, either are the parameters(weight) stored in Network layer. In addition, Tensor has many powerful features and functions to implement the computation behind neural network. Let's see the example1 and 2 to make friends with Tensor!

#### Topic 1.1 Pytorch Tensor

Turn on jupyter notebook, and click example1\_tensor.ipynb

(Estimated Time to Finish: 5 minutes)

#### Topic 1.2 Pytorch Autograd

Turn on jupyter notebook, and click jupyter notebook

(Estimated Time to Finish: 5 minutes)

### Topic/Activity 2 Network architecture

After we are familiar with Tensor, we know computational graphs and autograd in torch.Tensor are very powerful paradigm for defining complex operators and automatically taking derivatives; however for large neural networks raw autograd can be a bit too low-level. Pytorch provide many tools to help us build a network in code, such as [**torch.nn**](https://pytorch.org/docs/stable/nn.html) package. For example, we only use torch.nn module to design our network layer and package them sequentially to a model. All of parameters(weight) in network are also packaged in the model, so it’s more convenience to manage the parameters in high-level.

In example3, we introduce the network declaration with torch.nn module, also show the backward propagation process in this tutorial, then we will implement the linear regression with Pytorch in example4. there are two sections in that jupyter notebook: one does all things ‘manually’, and second one uses readable code to complete mission, it's important paradigm that show the processing to define a network and its data flow(forward/backward propagation).

#### Topic 2.2 Pytorch Neural Network

Turn on jupyter notebook, and click example3\_nn.ipynb

(Estimated Time to Finish: 15 minutes)

#### Topic 2.2 Pytorch Multilayer Perceptron

Turn on jupyter notebook, and click example4\_mlp.ipynb

(Estimated Time to Finish: 15 minutes)

**Disscussion:**

1. **What's the reason that pytorch need to use ‘Tensor’ instead of ‘Numpy’?**
2. **What's the mathematics behind the backward propagation, and what role does it play in deep learning?**
3. **In example3, could you generalize the software architecture in Pytorch that used the feed-forward neural network to solve problem?**
4. **What do you think about ‘loss curve visualization’ in network training process? Is it an important process or just a redundant work? (No standard answer)**

## Reference

Pytorch official tutorial: <https://pytorch.org/tutorials/>

Mila-udem pytorch tutorial: <https://github.com/mila-udem/welcome_tutorials/tree/master/pytorch>