Advanced Artificial Intelligence

Lab 01

Outline

- Introduction to Python
- Introduction to Tensorflow
- Introduction to Pytorch
- Exercise

Introduction to Python

Python is a high-level programming language with **explanatory**, **object-oriented** and **dynamic data types**.

- Explanatory: no compilation in the development process.
- Object oriented: Python supports object-oriented style or code encapsulated in object programming technology.
- Dynamic data types: in Python, data types are determined during runtime, not declared in advance.

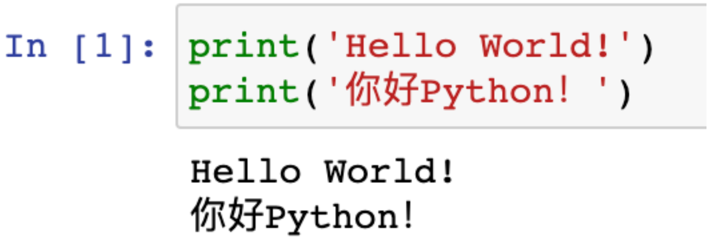
Introduction to Python

Advantages of Python:

- **Simplicity** -- Python is a language that represents the idea of simplicity. Reading a good Python program feels like reading English. This pseudo code nature of Python is one of its greatest advantages. It allows you to focus on solving problems rather than understanding the language itself.
- **High level language** - when you write a program in Python, you don't need to consider the low-level details such as how to manage the memory used by your program.
- **Scalability** - if you need a piece of your key code to run faster or want some algorithms not to be exposed, you can write some programs in C or C + +, and then use them in Python programs.
- **Standard code** -- Python uses forced indentation to make the code extremely readable.

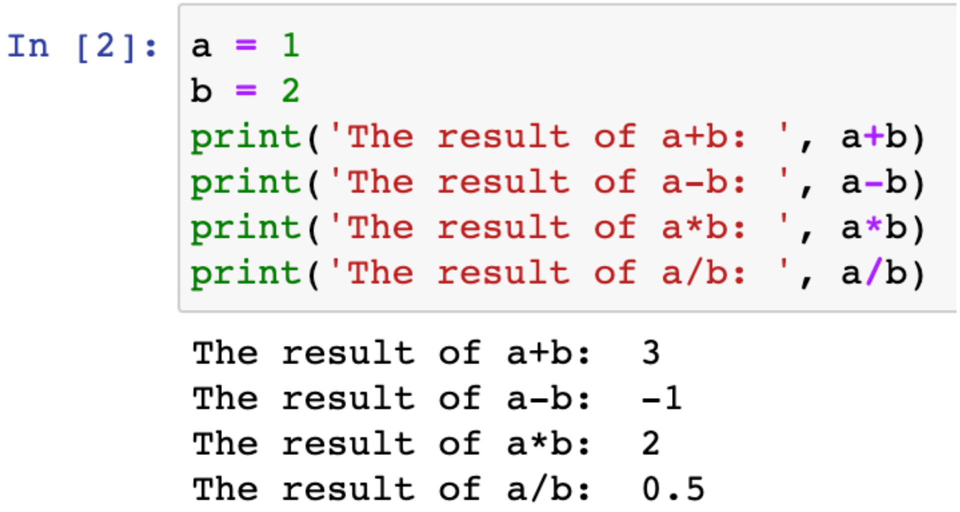
Introduction to Python

A simple Python Program:



```
In [1]: print('Hello World!')
        print('你好Python！')

Hello World!
你好Python！
```



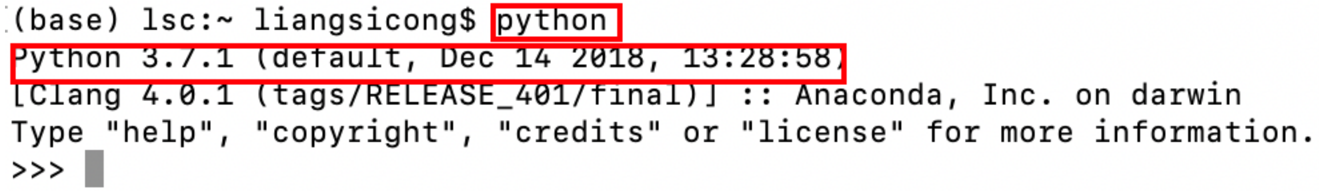
```
In [2]: a = 1
        b = 2
        print('The result of a+b: ', a+b)
        print('The result of a-b: ', a-b)
        print('The result of a*b: ', a*b)
        print('The result of a/b: ', a/b)

The result of a+b:  3
The result of a-b:  -1
The result of a*b:  2
The result of a/b:  0.5
```

Introduction to Python

Python installation：

- Python can be applied to multiple platforms, including Linux and Mac OS X.
- You can enter the "Python" command in the terminal window to check whether Python has been installed locally and the installed version of Python. Take Mac OS X as an example:



```
(base) lsc:~ liangsicong$ python
Python 3.7.1 (default, Dec 14 2018, 13:28:58)
[Clang 4.0.1 (tags/RELEASE_401/final)] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

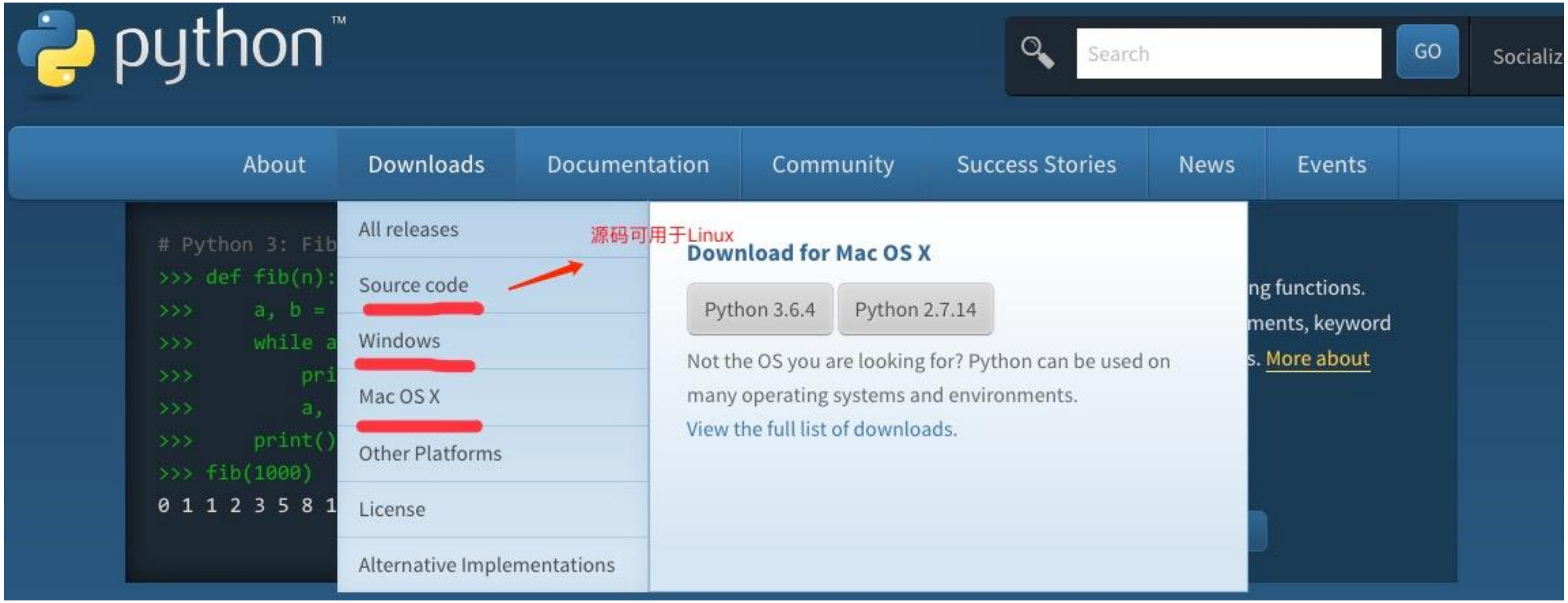
For the installation process, refer to: https://www.runoob.com/python/python-install.html

PS: since it has been officially announced that Python 2 update will be stopped, it is strongly recommended to install Python version above 3.6.

Introduction to Python

Python installation：

- Python official website： https://www.python.org/



源码可用于Linux

```
# Python 3: Fib
>>> def fib(n):
>>>     a, b =
>>>     while a
>>>         pri
>>>         a,
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 1
```

Download for Mac OS X

Not the OS you are looking for? Python can be used on many operating systems and environments.

View the full list of downloads.

Introduction to Python

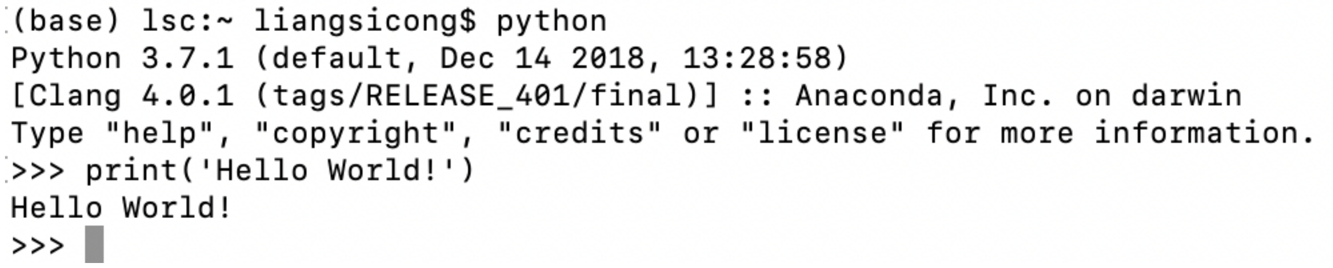
Python installation：

- Install Python using Anaconda (recommended)
- For beginners of learning python, installing Anaconda package management software is a good choice, which can reduce a lot of trouble in subsequent installation of Python packages. Writing code in Anaconda's own notebook is much better than using the command line.
- Download address of Anaconda: https://www.anaconda.com/download

Introduction to Python

Writing and execution of Python programs:

- Command line: open CMD (terminal on Mac OS computer) and enter ‘python’

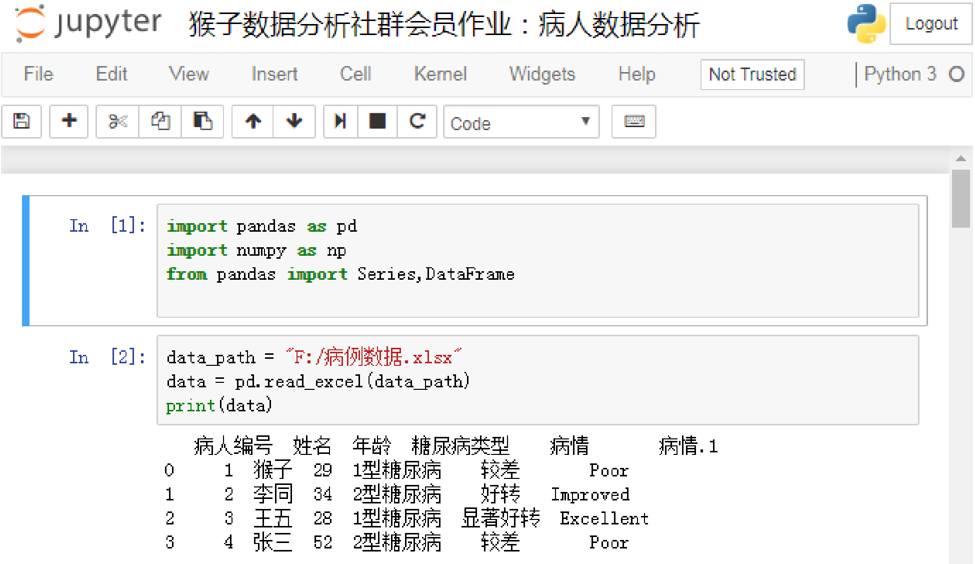


```
(base) lsc:~ liangsicong$ python
Python 3.7.1 (default, Dec 14 2018, 13:28:58)
[Clang 4.0.1 (tags/RELEASE_401/final)] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hello World!')
Hello World!
>>> 
```

Introduction to Python

Writing and execution of Python programs：

- Example of Jupyter notebook (recommended) supplied with Anaconda：



In [1]:
```
import pandas as pd
import numpy as np
from pandas import Series,DataFrame
```

In [2]:
```
data_path = "F:/病例数据.xlsx"
data = pd.read_excel(data_path)
print(data)
```

```
   病人编号  姓名  年龄  糖尿病类型    病情      病情.1
0     1  猴子  29  1型糖尿病    较差       Poor
1     2  李同  34  2型糖尿病    好转   Improved
2     3  王五  28  1型糖尿病  显著好转  Excellent
3     4  张三  52  2型糖尿病    较差       Poor
```

Jupyter Notebook official website： https://jupyter.org/

Introduction to Python

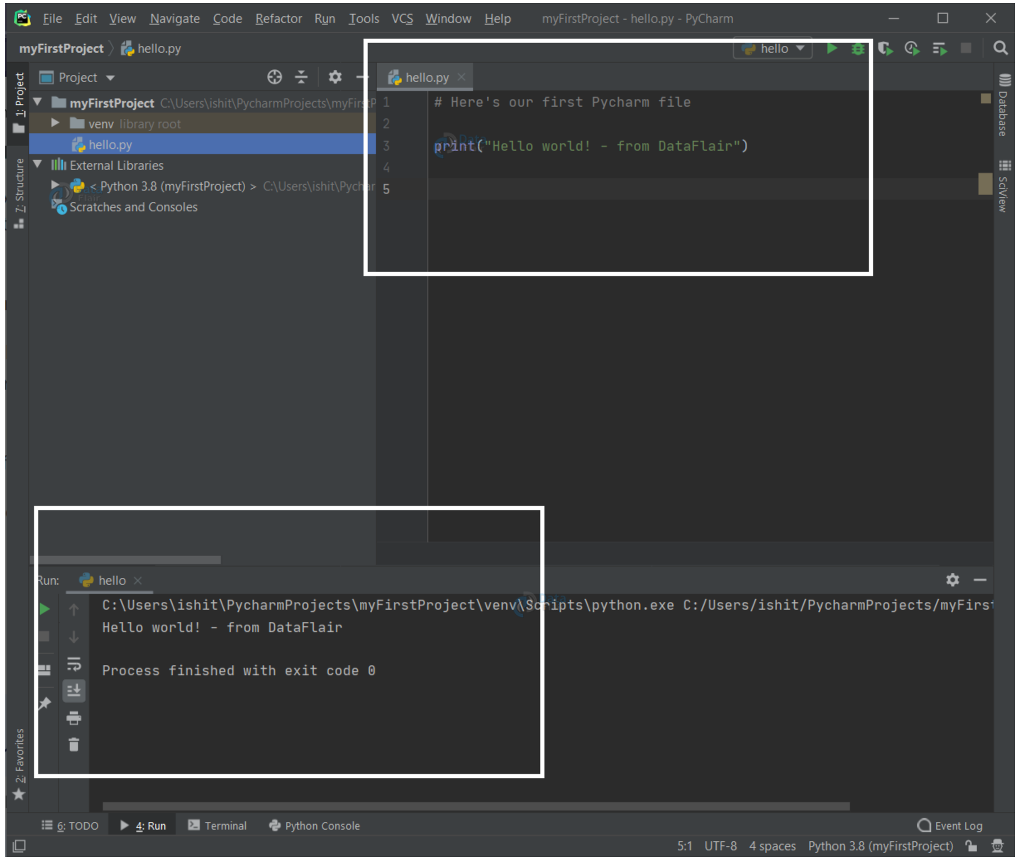
Writing and execution of Python programs：

- Integrated development environment (IDE): Pycharm (recommended)
- Pycharm is a python IDE created by JetBrains, which supports Mac OS, windows and Linux systems.
- Pycharm functions: debugging, syntax highlighting, project management, code jump, intelligent prompt, automatic completion, unit test, version control
- Pycharm download address: https://www.jetbrains.com/pycharm/download/
- Pycharm installation address: http://www.runoob.com/w3cnote/pycharm-windows-install.html

Introduction to Python

Writing and execution of Python programs：

- Pycharm example：


File Edit View Navigate Code Refactor Run Tools VCS Window Help
myFirstProject - hello.py - PyCharm
myFirstProject
hello.py
Project
myFirstProject C:\Users\ishit\PycharmProjects\myFirstP
venv library root
hello.py
External Libraries
< Python 3.8 (myFirstProject) > C:\Users\ishit\Pychar
Scratches and Consoles
hello
Run:
6: TODO
4: Run
Terminal
Python Console
Event Log
5:1 UTF-8 4 spaces Python 3.8 (myFirstProject)
Database
SciView
1: Project
Z: Structure
2: Favorites

```
# Here's our first Pycharm file

print("Hello world! - from DataFlair")
```

```
C:\Users\ishit\PycharmProjects\myFirstProject\venv\Scripts\python.exe C:/Users/ishit/PycharmProjects/myFirst
Hello world! - from DataFlair

Process finished with exit code 0
```

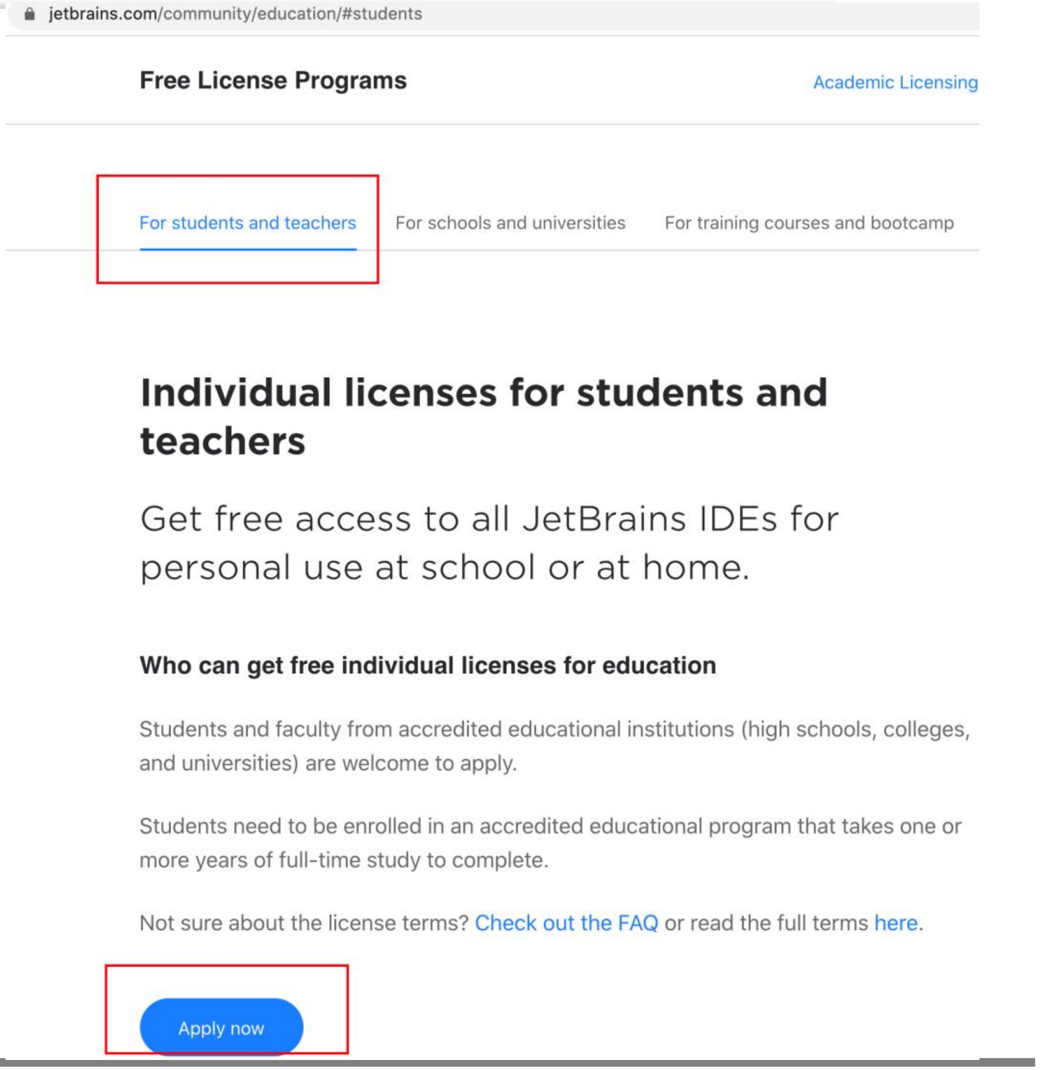
Introduction to Python

Application for Pycharm free license：

- Students can apply for the professional version of pychart free of charge, as follows:
    - Application address: https://www.jetbrains.com/student/
    - After completing the school email application, you will receive an activation email. After clicking the link to activate, you will receive an email containing the download address link, and you can enjoy all the professional software of JetBrains.
        - For example: 1234567@mail.sustech.edu.cn

Introduction to Python

Application for pycharm free license：



jetbrains.com/community/education/#students

Free License Programs

For students and teachers
For schools and universities
For training courses and bootcamp

Individual licenses for students and teachers

Get free access to all JetBrains IDEs for personal use at school or at home.

Who can get free individual licenses for education

Students and faculty from accredited educational institutions (high schools, colleges, and universities) are welcome to apply.

Students need to be enrolled in an accredited educational program that takes one or more years of full-time study to complete.

Not sure about the license terms? Check out the FAQ or read the full terms here.

Introduction to Python

Recommended materials for Python：

- https://www.python.org/
- https://www.runoob.com/python/python-tutorial.html
- https://www.w3school.com.cn/python/index.asp
- Major technical blogs / communities / forums, etc

Introduction to Tensorflow

TensorFlow is a free and **open-source software** library for machine learning and artificial intelligence developed by Google Brain Team.

It can be used across a range of tasks but has a **particular focus on training and inference of deep neural networks**.

Tensorflow allows us to draw a **calculation structure diagram**, which can also be called a series of human-computer interactive calculation operations, and then convert the edited Python file into more efficient C + + and calculate at the backend.

Introduction to Tensorflow

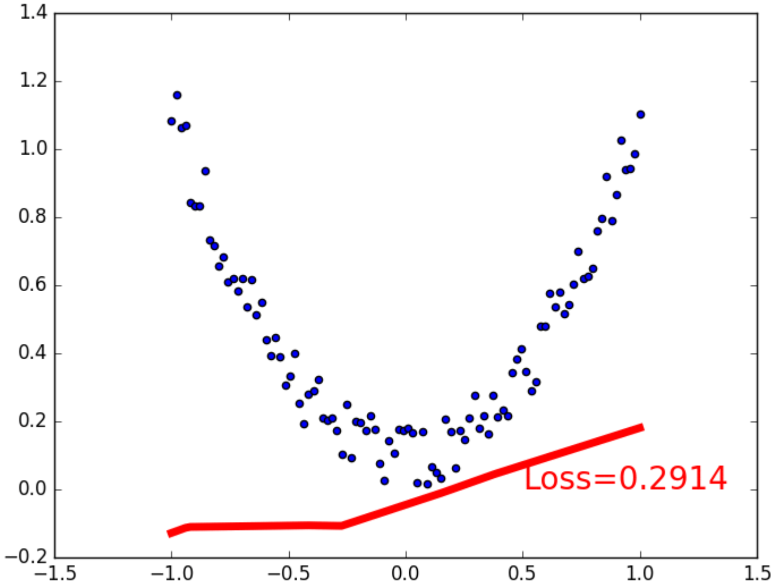
Tensorflow installation：

- Refer to: https://tensorflow.google.cn/install
- matters needing attention:
  - Tensorflow is provided as CPU version and GPU version. Tensorflow of GPU version can use the GPU provided by the computer / server to speed up the operation speed (only support the newer NVIDIA GPU of specific model);
  - Tensorflow has been updated to version 2.0 +, with major syntax changes compared with version 1. X. at the same time, many open-source tensorflow codes are also version 1. X.

Introduction to Tensorflow

Code example of Tensorflow：

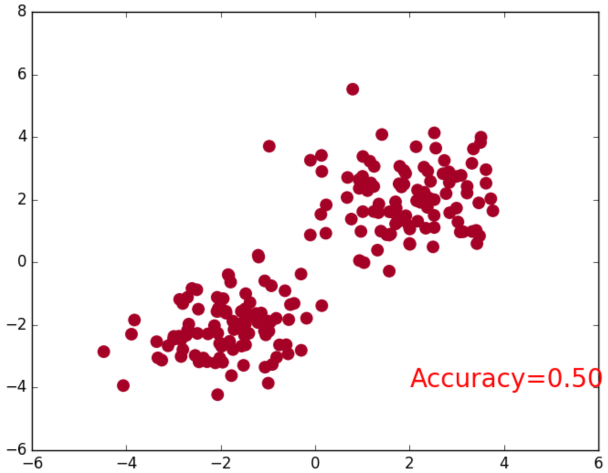
- What is regression task: continuous variable prediction, such as generating a function to fit existing data and predict future results



Introduction to Tensorflow

Code example of Tensorflow：

- What is classification task: discrete variable prediction, for example, generates a function that can distinguish different types of data, such as cat and dog classification



Introduction to Tensorflow

The difference between classification and regression lies in the type of output variables.

Regression: continuous variable prediction;

Classification: discrete variable prediction.

For example, predicting house prices is a regression task; Dividing things into categories, such as cats, dogs, pigs and cattle, is a classification task.

Introduction to Tensorflow

Code example of Tensorflow：

- Use Tensorflow to complete the classification of MNIST handwritten numeral dataset
- Training data: 55,000 image with 28x28 pixels

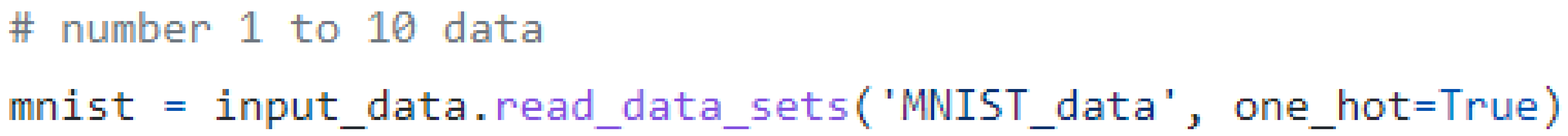
Introduction to Tensorflow

Usually, the process of completing a task includes:

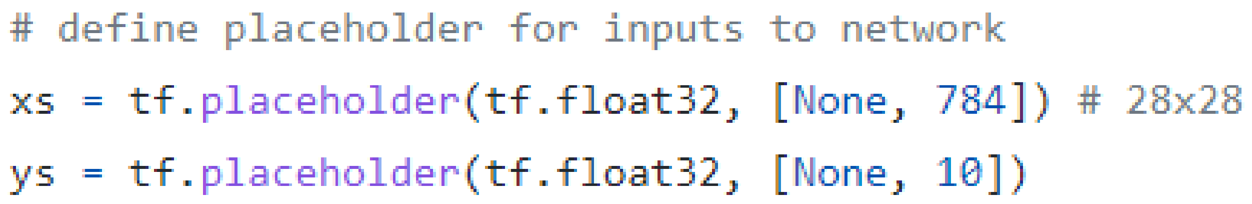
- **load data**
- **build neural network**
- **select appropriate loss function and optimizer**
- **train and predict**

Introduction to Tensorflow

load data



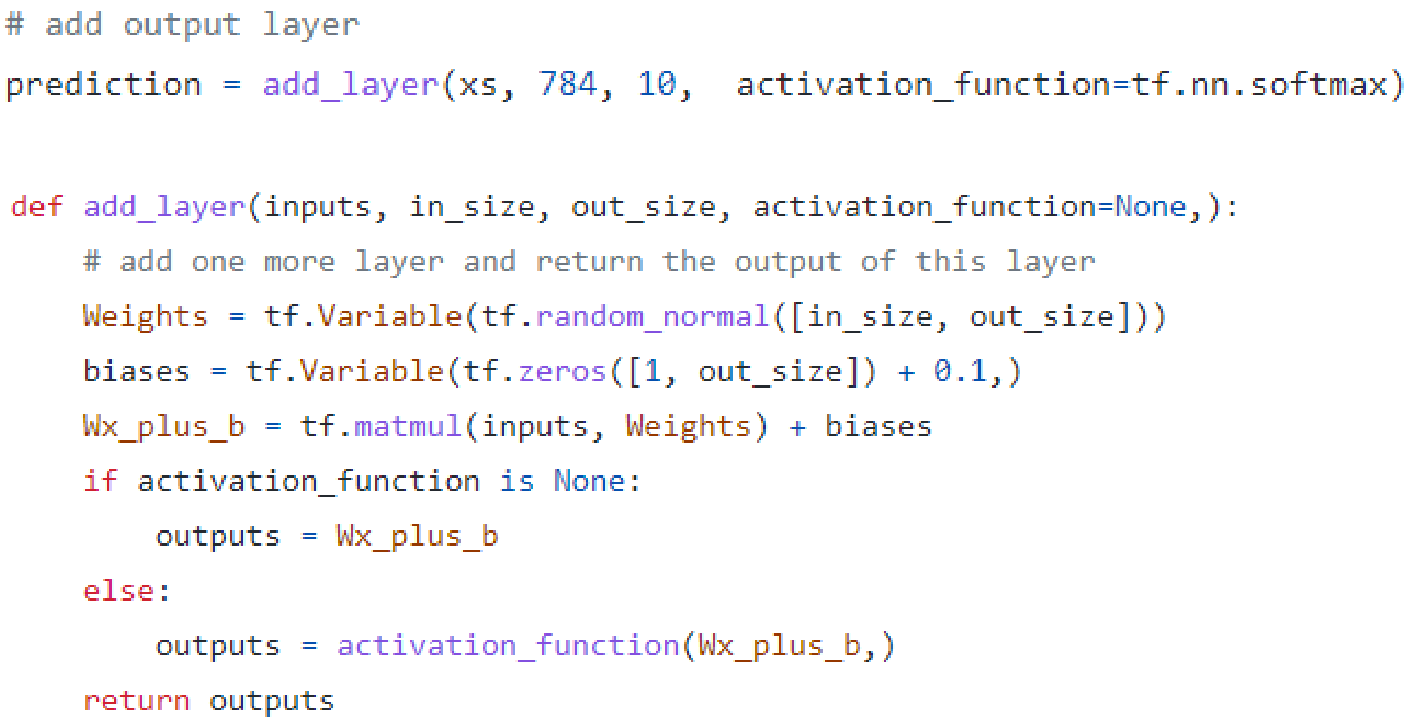
```
# number 1 to 10 data
mnist = input_data.read_data_sets('MNIST_data', one_hot=True)
```



```
# define placeholder for inputs to network
xs = tf.placeholder(tf.float32, [None, 784]) # 28x28
ys = tf.placeholder(tf.float32, [None, 10])
```

Introduction to Tensorflow

build neural network

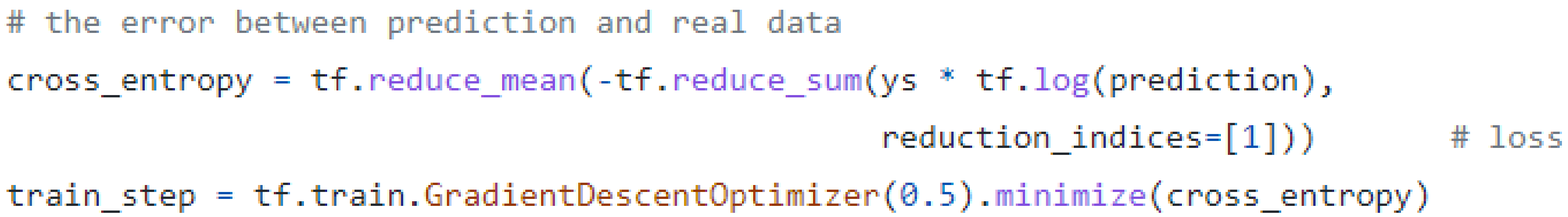


```
# add output layer
prediction = add_layer(xs, 784, 10,  activation_function=tf.nn.softmax)

def add_layer(inputs, in_size, out_size, activation_function=None,):
    # add one more layer and return the output of this layer
    Weights = tf.Variable(tf.random_normal([in_size, out_size]))
    biases = tf.Variable(tf.zeros([1, out_size]) + 0.1,)
    Wx_plus_b = tf.matmul(inputs, Weights) + biases
    if activation_function is None:
        outputs = Wx_plus_b
    else:
        outputs = activation_function(Wx_plus_b,)
    return outputs
```

Introduction to Tensorflow

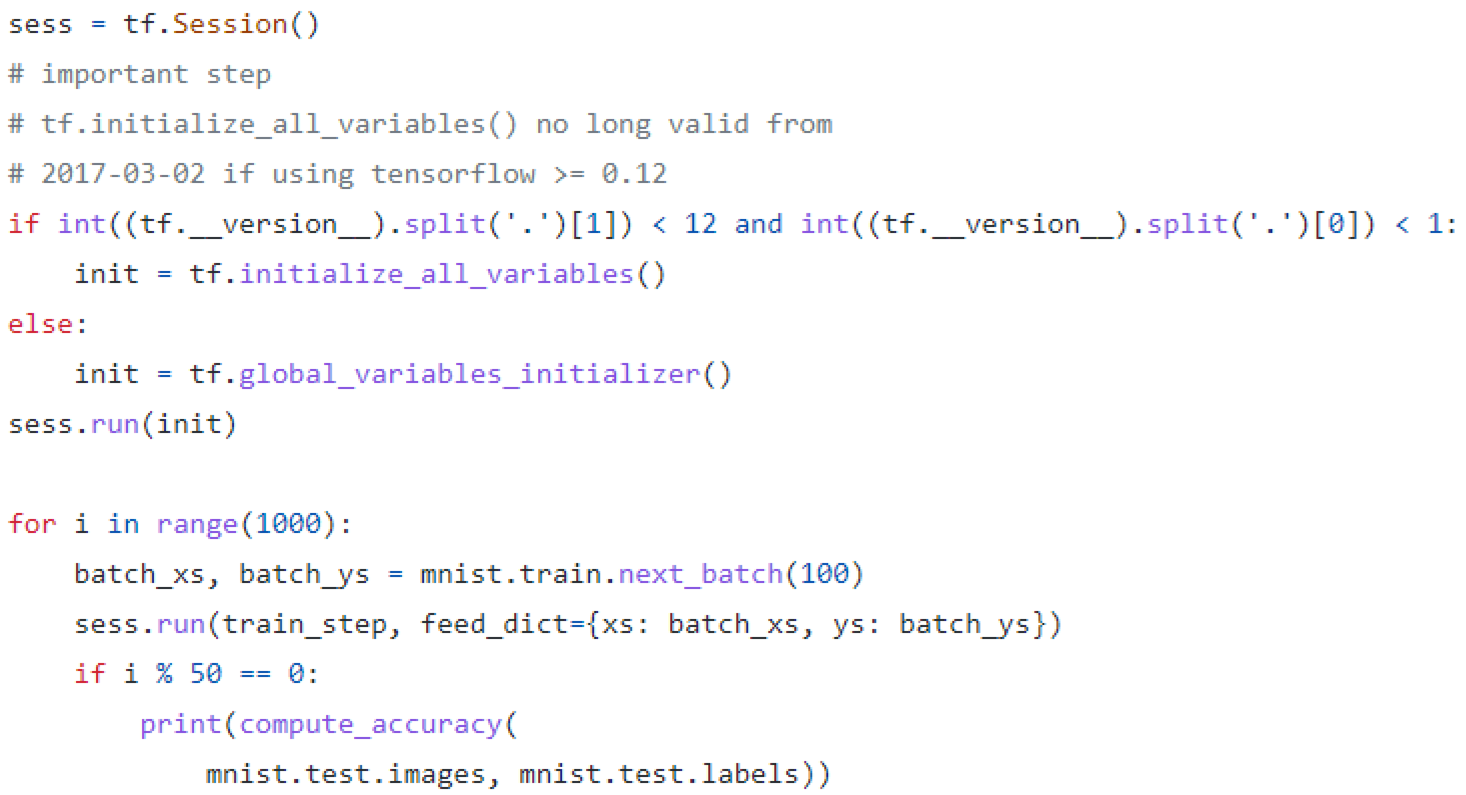
select appropriate loss function and optimizer



```
# the error between prediction and real data
cross_entropy = tf.reduce_mean(-tf.reduce_sum(ys * tf.log(prediction),
                                              reduction_indices=[1]))       # loss
train_step = tf.train.GradientDescentOptimizer(0.5).minimize(cross_entropy)
```

Introduction to Tensorflow

train and predict



```
sess = tf.Session()
# important step
# tf.initialize_all_variables() no long valid from
# 2017-03-02 if using tensorflow >= 0.12
if int((tf.__version__).split('.')[1]) < 12 and int((tf.__version__).split('.')[0]) < 1:
    init = tf.initialize_all_variables()
else:
    init = tf.global_variables_initializer()
sess.run(init)

for i in range(1000):
    batch_xs, batch_ys = mnist.train.next_batch(100)
    sess.run(train_step, feed_dict={xs: batch_xs, ys: batch_ys})
    if i % 50 == 0:
        print(compute_accuracy(
            mnist.test.images, mnist.test.labels))
```

Introduction to Tensorflow

Complete code refer to：

https://github.com/MorvanZhou/tutorials/tree/master/tensorflowTUT/tf16_classification

Introduction to Tensorflow

Recommended materials for Tensorflow：

- https://tensorflow.google.cn/
- https://www.w3cschool.cn/tensorflow_python/
- https://mofanpy.com/tutorials/machine-learning/tensorflow/

Introduction to Pytorch

Introduction of Pytorch：

- Pytorch is a derivative of torch in Python. Because pytorch is a neural network library using the torch language, torch is easy to use, but Lua is not particularly popular. All development teams transplanted Lua's torch to the more popular language python.
- If you've known before **Numpy** (a famous package that provides mathematical calculation in Python) can quickly become familiar with pytorch, because pytorch is a module that can be used to replace numpy in the field of neural network.

Introduction to Pytorch

Introduction of Pytorch：

- According to Pytorch official declaration, the biggest advantage of his family is that the **neural network established is dynamic**. Compared with the static tensorflow, he can deal with some problems more effectively, such as the output of RNN change time length.
- The code written by pytorch is **better understood** than tensorflow.

Introduction to Pytorch

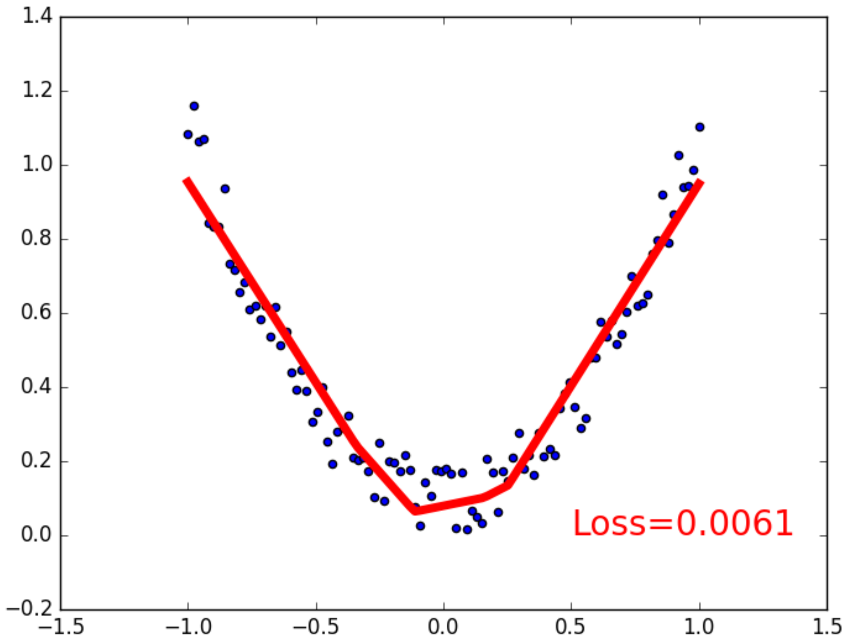
Pytorch installation：

- Refer to： https://pytorch.org/get-started/locally/
- matters needing attention:
  - Pytorch is provided as CPU version and GPU version. Pytorch of GPU version can use the GPU provided by the computer / server to speed up the operation speed (only support the newer NVIDIA GPU of specific model);

Introduction to Pytorch

Code example of Pytorch：

- Use pytorch to fit a quadratic function



Complete code refer to： https://github.com/MorvanZhou/PyTorch-Tutorial/blob/master/tutorial-contents/301_regression.py

Introduction to Pytorch

Recommended materials for Pytorch：

- https://pytorch.org/tutorials/
- https://www.w3cschool.cn/pytorch/
- https://mofanpy.com/tutorials/machine-learning/torch/

Conclusion

Server: Anaconda, (miniconda); Pytorch.

Clint: Pycharm.

Jupyter notebook

Exercise

After completing the installation of all tools according to the PPT above,

- refer to https://mofanpy.com/tutorials/machine-learning/torch/regression/ Or other online tutorials to implement a simple regression model based on tensorflow / pytorch
- refer to https://mofanpy.com/tutorials/machine-learning/torch/classification/ Or other online tutorials to implement a simple classification model based on tensorflow / pytorch