Open Software Project

Lecture 12: Backpropagation and MLP

Dongbo Min

Department of Computer Science and Engineering

Ewha Womans University, Korea

E-mail: dbmin@ewha.ac.kr

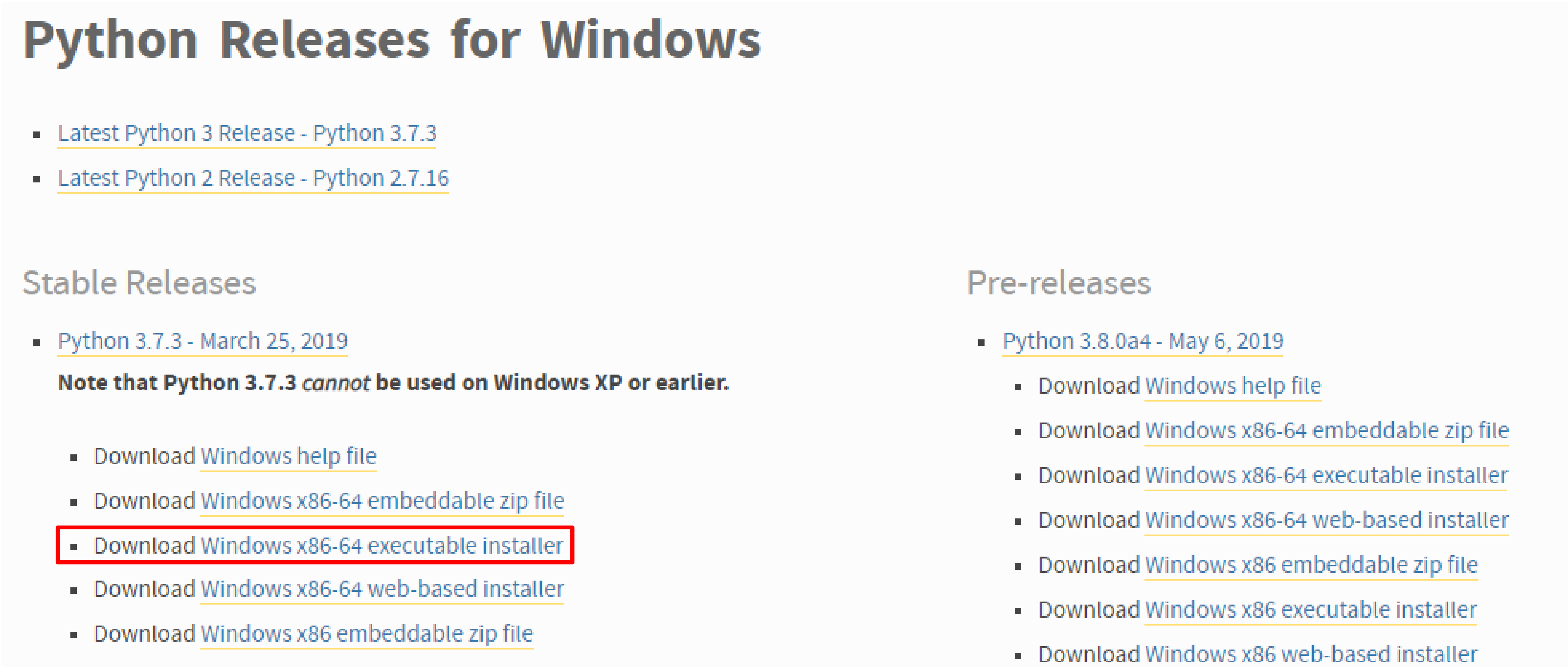
Contents

- Python and Jupyter Notebook tutorial

- Simple neural networks: Multi-layer perceptron (MLP)

Jupyter Notebook

- Install Python3
  - https://www.python.org/downloads/windows/



Python Releases for Windows

- Latest Python 3 Release - Python 3.7.3
- Latest Python 2 Release - Python 2.7.16

Stable Releases

- Python 3.7.3 - March 25, 2019

  Note that Python 3.7.3 *cannot* be used on Windows XP or earlier.

  - Download Windows help file
  - Download Windows x86-64 embeddable zip file
  - Download Windows x86-64 executable installer
  - Download Windows x86-64 web-based installer
  - Download Windows x86 embeddable zip file

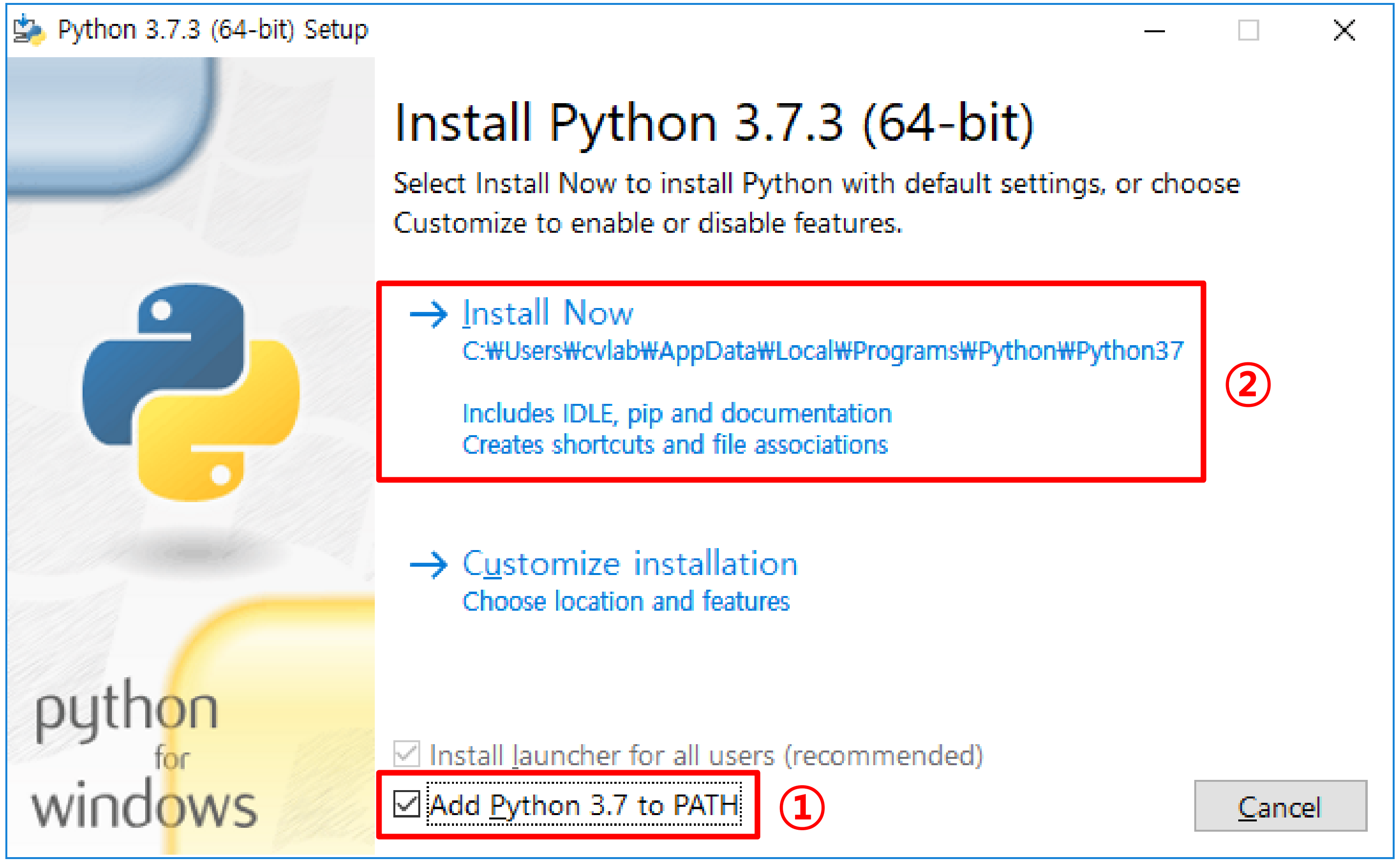
Pre-releases

- Python 3.8.0a4 - May 6, 2019
  - Download Windows help file
  - Download Windows x86-64 embeddable zip file
  - Download Windows x86-64 executable installer
  - Download Windows x86-64 web-based installer
  - Download Windows x86 embeddable zip file
  - Download Windows x86 executable installer
  - Download Windows x86 web-based installer

❖ **If your Windows is 32-bit, Download “Windows x86-executable installer”**

Jupyter Notebook

- Install Python3
  ① Check “Add Python 3.7 to PATH”
  ② Click Install Now



Python 3.7.3 (64-bit) Setup

Install Python 3.7.3 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

→ Install Now
C:₩Users₩cvlab₩AppData₩Local₩Programs₩Python₩Python37

Includes IDLE, pip and documentation
Creates shortcuts and file associations

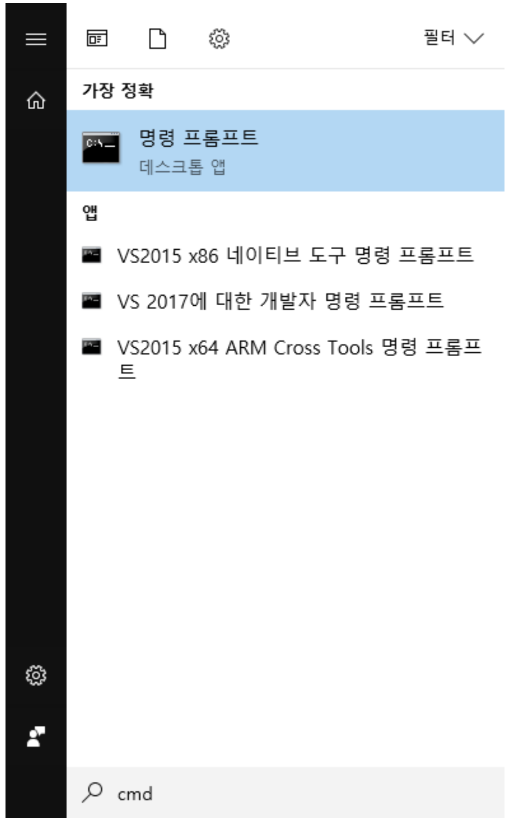
→ Customize installation
Choose location and features

☑ Install launcher for all users (recommended)

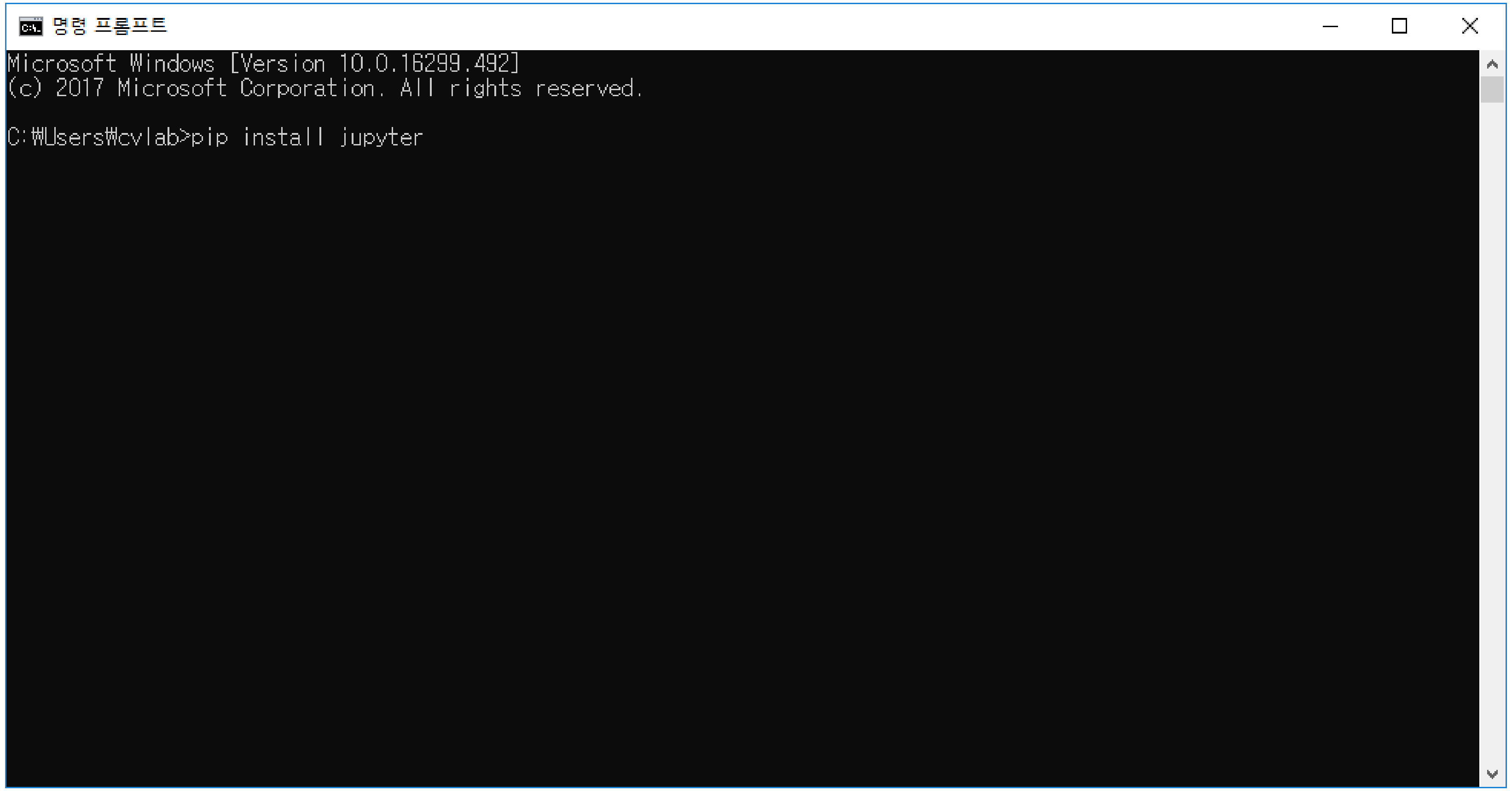
☑ Add Python 3.7 to PATH

Jupyter Notebook

- Install Jupyter Notebook
  - Ctrl + ESC -> type “*cmd*”
  - Type the following command “*pip install jupyter*”


필터
가장 정확
명령 프롬프트
데스크톱 앱
앱
VS2015 x86 네이티브 도구 명령 프롬프트
VS 2017에 대한 개발자 명령 프롬프트
VS2015 x64 ARM Cross Tools 명령 프롬프트

cmd

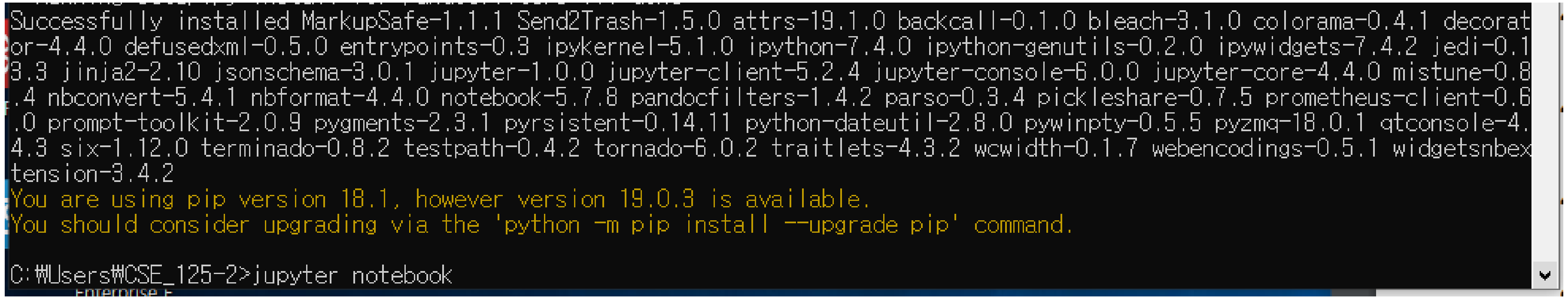


```
Microsoft Windows [Version 10.0.16299.492]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\cvlab>pip install jupyter
```

Jupyter Notebook

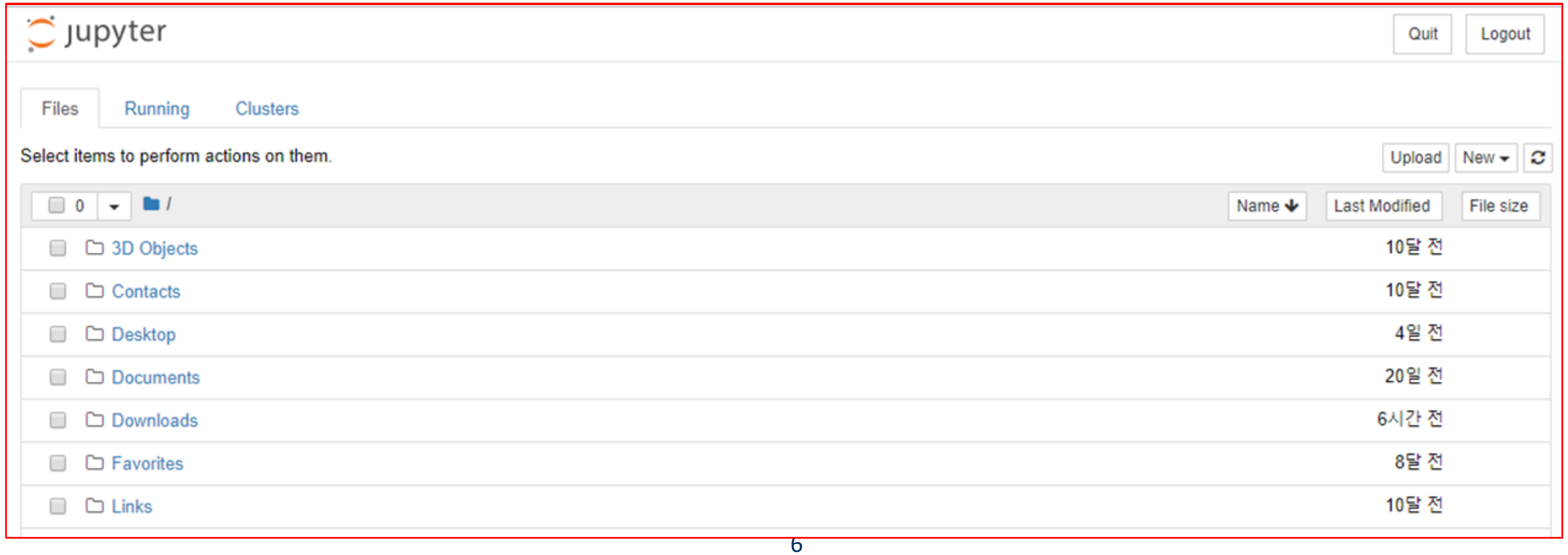
- Install Jupyter Notebook
  - Run the following command “*jupyter notebook*”



```
Successfully installed MarkupSafe-1.1.1 Send2Trash-1.5.0 attrs-19.1.0 backcall-0.1.0 bleach-3.1.0 colorama-0.4.1 decorat
or-4.4.0 defusedxml-0.5.0 entrypoints-0.3 ipykernel-5.1.0 ipython-7.4.0 ipython-genutils-0.2.0 ipywidgets-7.4.2 jedi-0.1
3.3 jinja2-2.10 jsonschema-3.0.1 jupyter-1.0.0 jupyter-client-5.2.4 jupyter-console-6.0.0 jupyter-core-4.4.0 mistune-0.8
.4 nbconvert-5.4.1 nbformat-4.4.0 notebook-5.7.8 pandocfilters-1.4.2 parso-0.3.4 pickleshare-0.7.5 prometheus-client-0.6
.0 prompt-toolkit-2.0.9 pygments-2.3.1 pyrsistent-0.14.11 python-dateutil-2.8.0 pywinpty-0.5.5 pyzmq-18.0.1 qtconsole-4.
4.3 six-1.12.0 terminado-0.8.2 testpath-0.4.2 tornado-6.0.2 traitlets-4.3.2 wcwidth-0.1.7 webencodings-0.5.1 widgetsnbex
tension-3.4.2
You are using pip version 18.1, however version 19.0.3 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:₩Users₩CSE_125-2>jupyter notebook
```

  - Now you can see the following screen

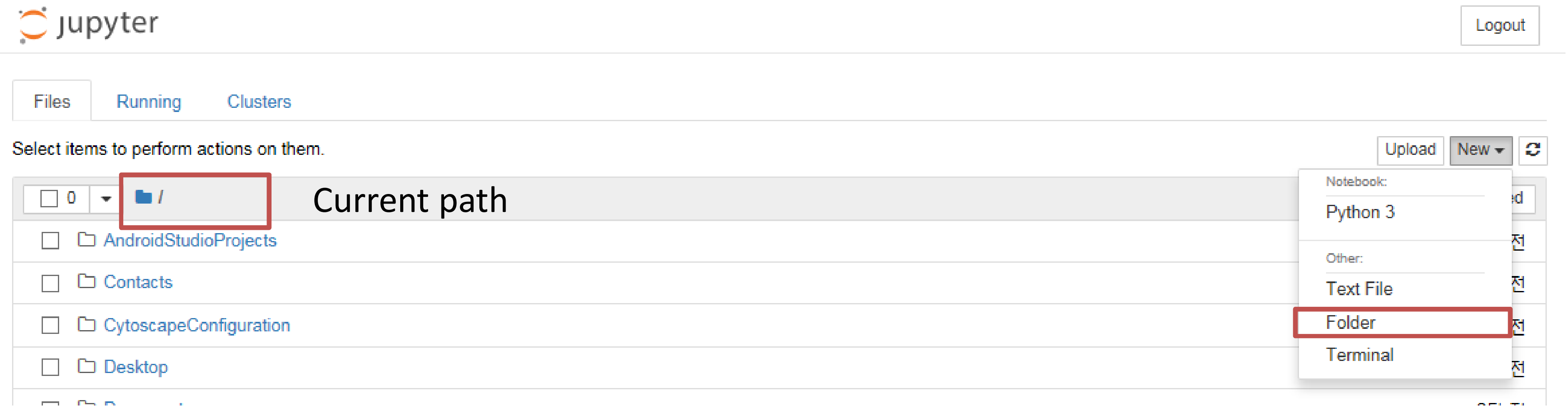


Select items to perform actions on them.

Name	Last Modified	File size
3D Objects	10달 전	
Contacts	10달 전	
Desktop	4일 전	
Documents	20일 전	
Downloads	6시간 전	
Favorites	8달 전	
Links	10달 전	

Create Directory

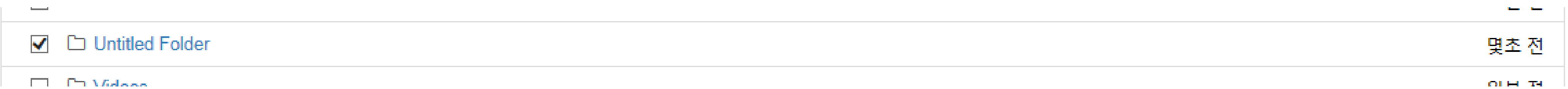
Click the "New" drop-down button in the upper right corner and select "Folder"



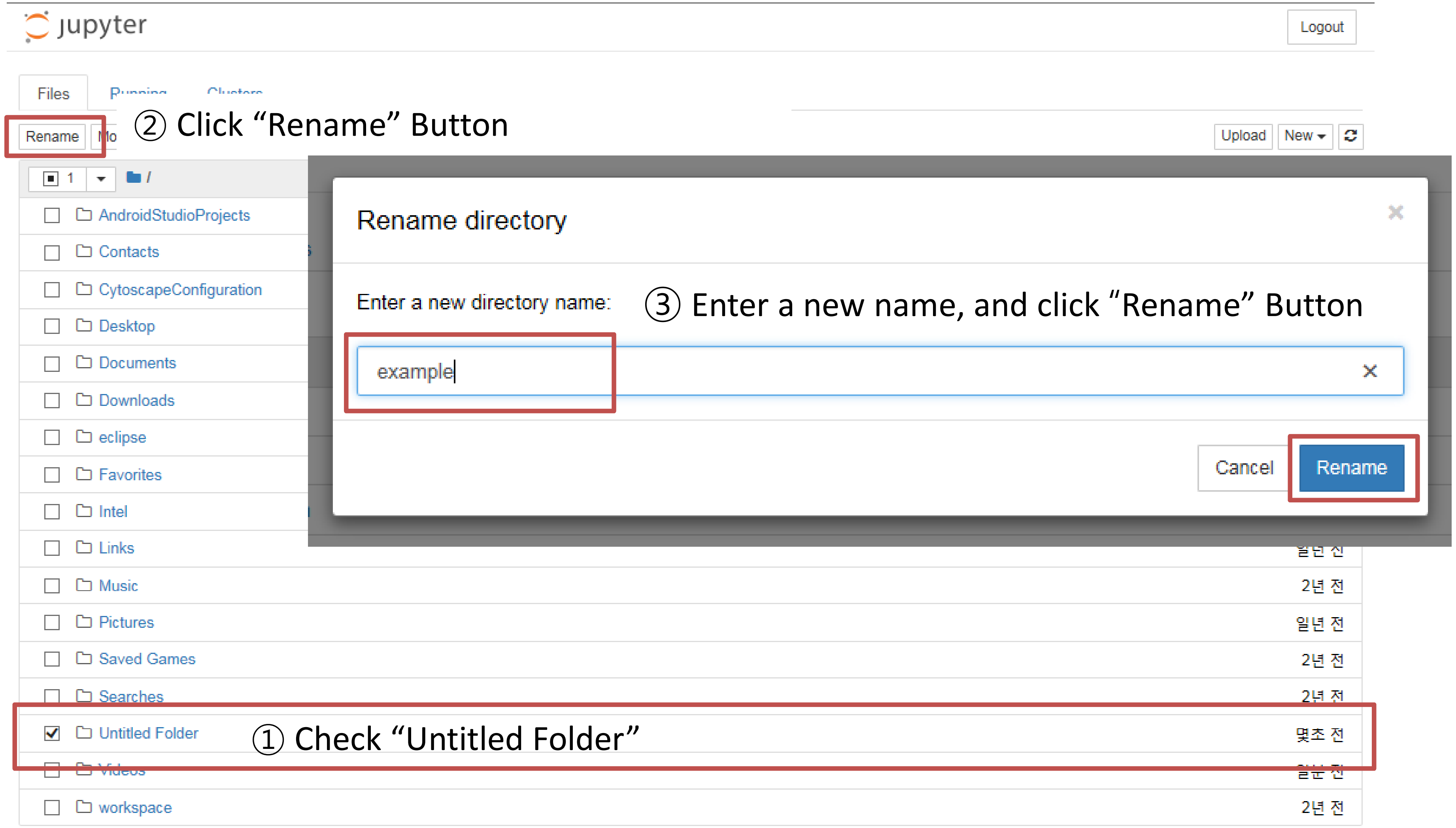
Select items to perform actions on them.

Current path

A new folder is created with the name "Utitled Folder"



Create Directory



② Click “Rename” Button

Rename directory

Enter a new directory name:

③ Enter a new name, and click “Rename” Button

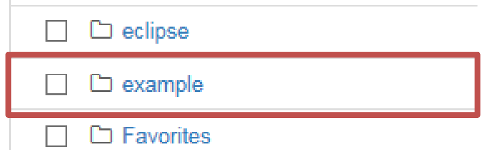
example

Cancel Rename

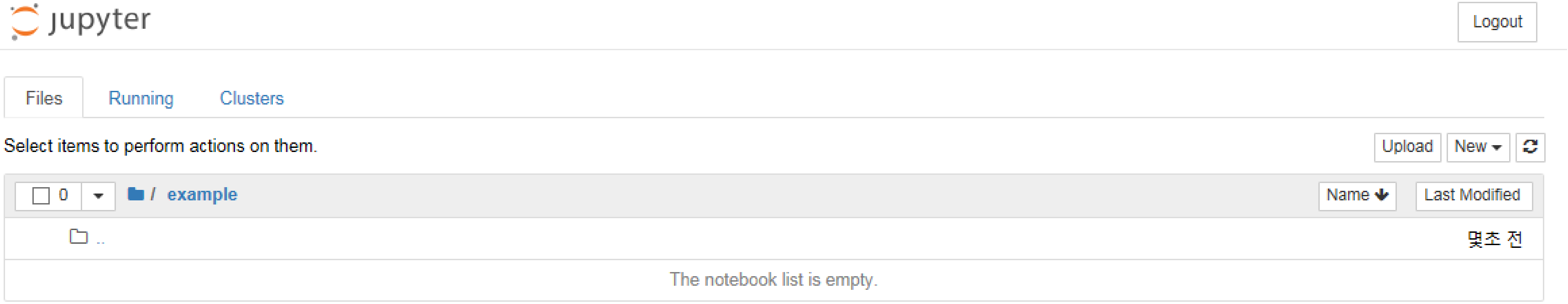
□	AndroidStudioProjects	
□	Contacts	
□	CytoscapeConfiguration	
□	Desktop	
□	Documents	
□	Downloads	
□	eclipse	
□	Favorites	
□	Intel	
□	Links	일년 전
□	Music	2년 전
□	Pictures	일년 전
□	Saved Games	2년 전
□	Searches	2년 전
☑	Untitled Folder	몇초 전
□	Videos	일년 전
□	workspace	2년 전

① Check “Untitled Folder”

Create a new notebook



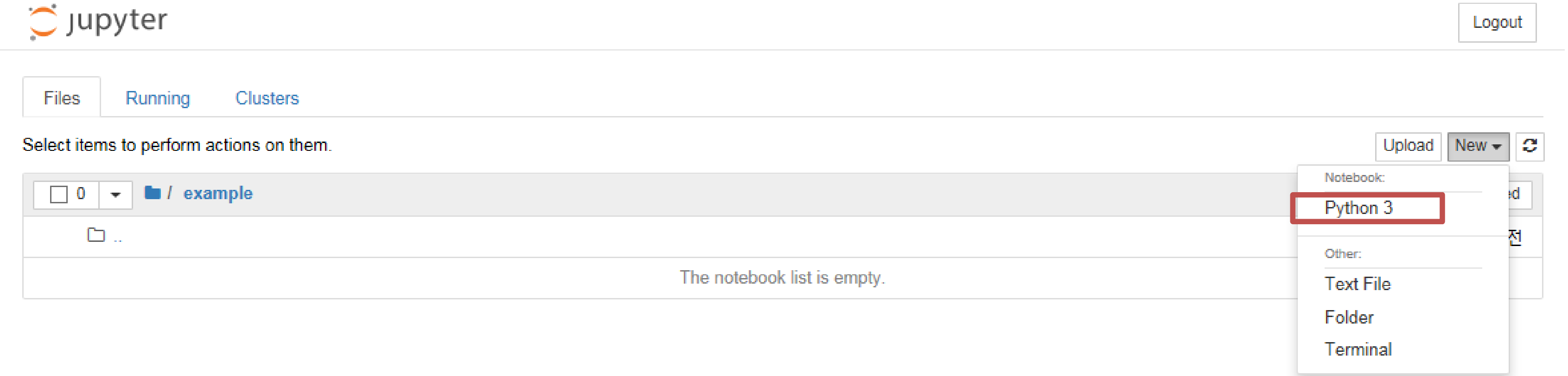
Click renamed folder



Select items to perform actions on them.

The notebook list is empty.

Click the "New" drop-down button in the upper right corner and select "Python 3"

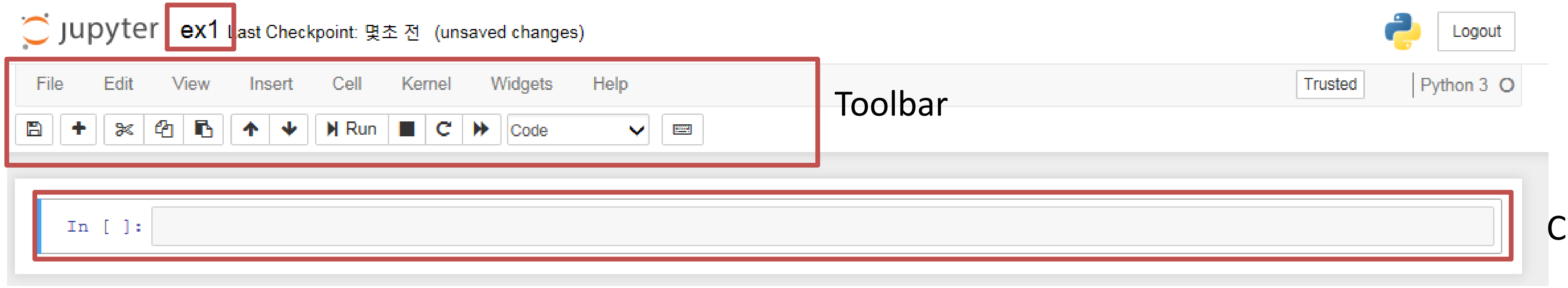


Select items to perform actions on them.

The notebook list is empty.

Create a new notebook

Current file name, click here to change



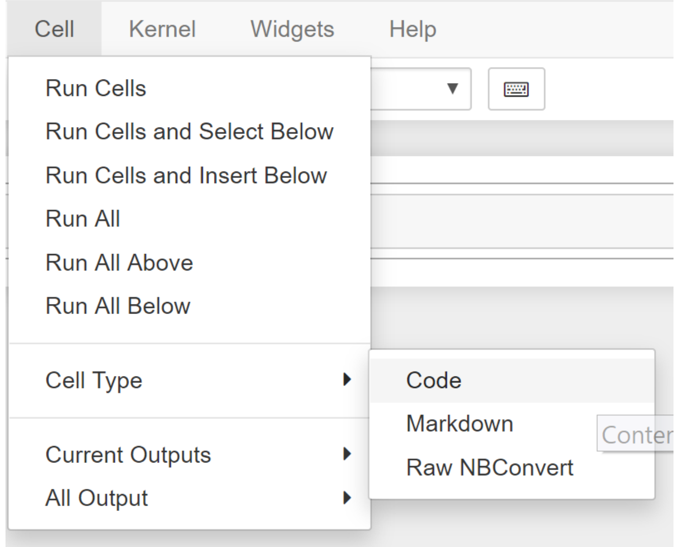
Toolbar

Cell

- kernel:
“computational engine” that executes the code contained in a notebook document.

- cell:
container for text to be displayed in the notebook or code to be executed by the notebook’s kernel.

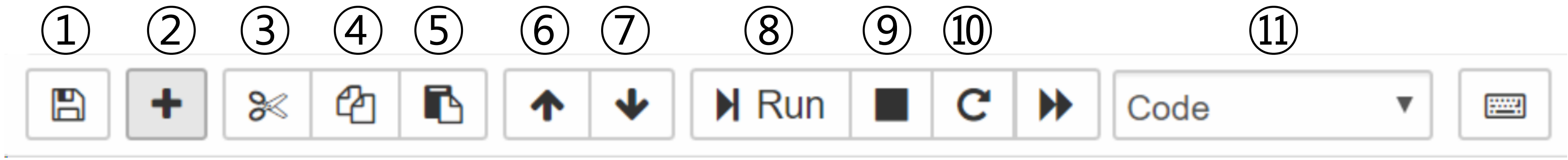
Cell



A **code** cell contains code to be executed in the kernel and displays its output below.

A **Markdown** cell contains text formatted using Markdown and displays its output in-place when it is run.

Notebook Interface



① save and checkpoint (Extension: .ipynb)

② insert new cell below

③④⑤ cell cut/copy/paste

⑥⑦ Move the selected cell position up/down

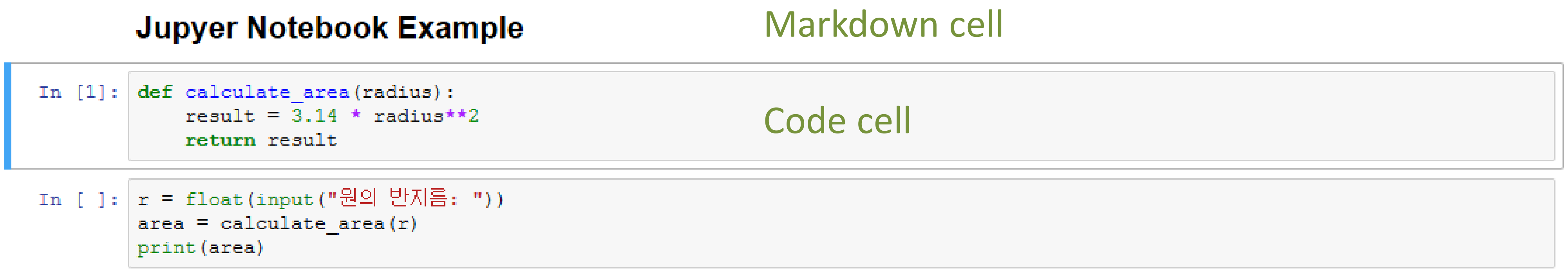
⑧ Run cell

⑨ interrupt the kernel

⑩ restart the kernel

⑪ Change cell type (markdown cell / code cell / …)

Example



Jupyer Notebook Example

Markdown cell

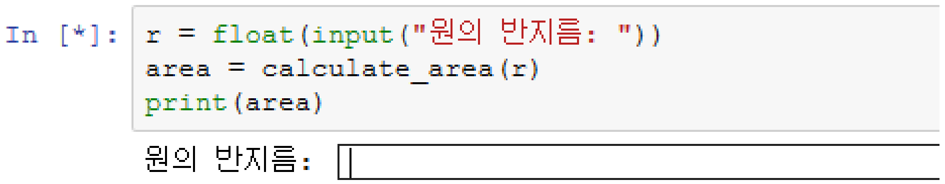
```
In [1]: def calculate_area(radius):
            result = 3.14 * radius**2
            return result
```

Code cell

```
In [ ]: r = float(input("원의 반지름: "))
        area = calculate_area(r)
        print(area)
```

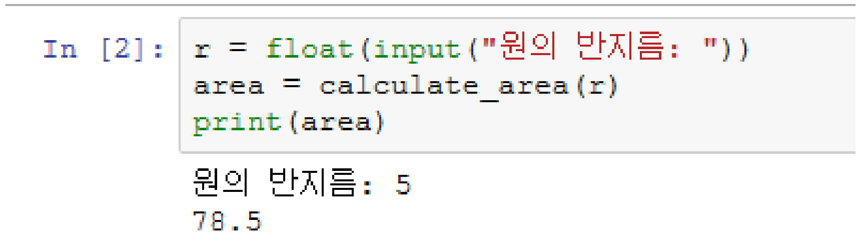
Run selected cell by pressing ⏯ Run button or by pressing shift+Enter

Execution Result



```
In [*]: r = float(input("원의 반지름: "))
        area = calculate_area(r)
        print(area)

        원의 반지름: 
```

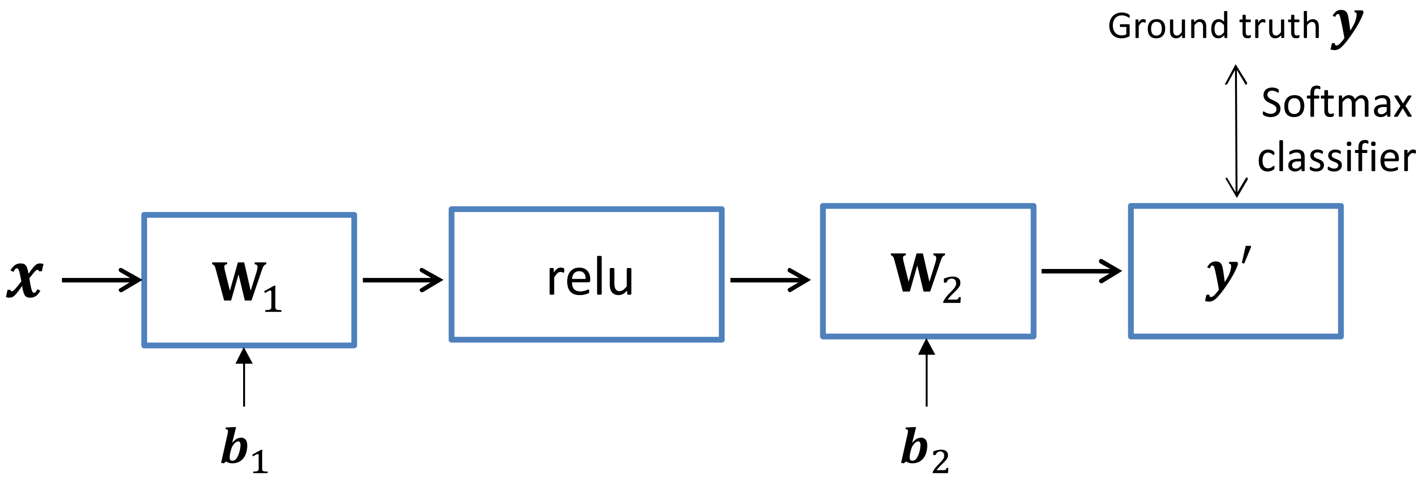


```
In [2]: r = float(input("원의 반지름: "))
        area = calculate_area(r)
        print(area)

        원의 반지름: 5
        78.5
```

2-layer Neural Network - Practice

- Implement and train 2-layer Neural Network.



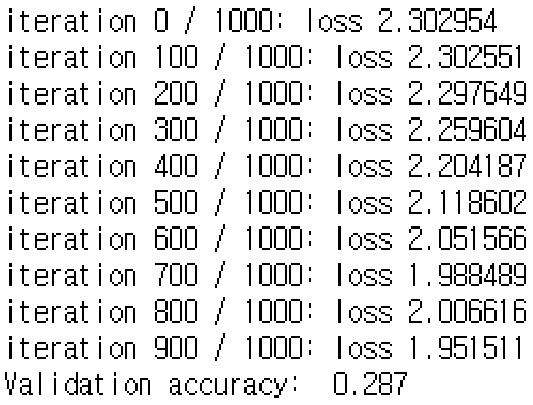
  - Perform the classification using “CIFAR-10” dataset
  - Two weights $\mathbf{W}_1, \mathbf{W}_2$ with biases $\boldsymbol{b}_1, \boldsymbol{b}_2$
  - Predicted output $\boldsymbol{y}' = \mathbf{W}_2\big(\mathrm{relu}(\mathbf{W}_1\boldsymbol{x} + \boldsymbol{b}_1)\big) + \boldsymbol{b}_2$
  - Total loss = data loss(softmax) + regularization loss
  - The IPython Notebook **two_layer_net.ipynb** will walk you through the implementation of a two-layer neural network classifier.

2-layer Neural Network - Practice

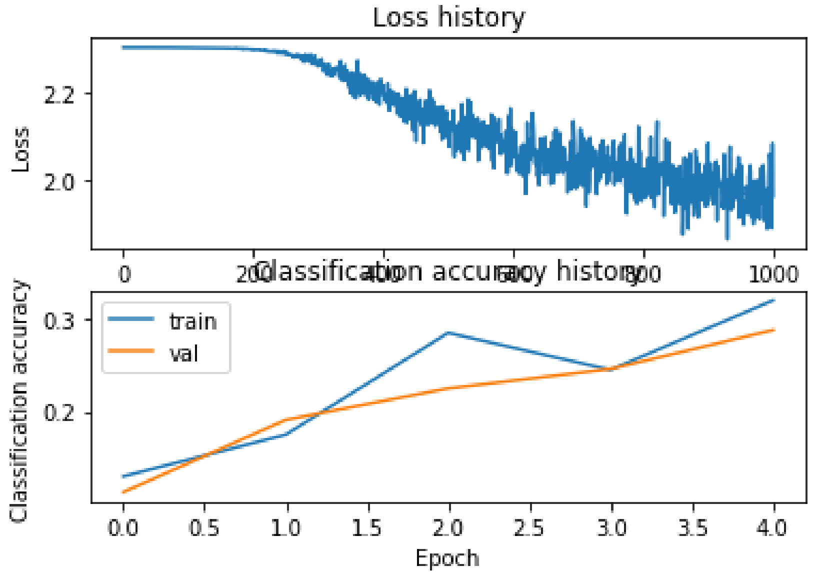
- Requirements
  - Need to install some python libraries to run **two_layer_net.ipynb**
  - Run the following command on prompt (cmd)
    1. *cd (path of assignment folder)*
       - e.g. *cd Downloads/Assignment 11*
    2. *pip install –r requirements.txt*
       - *pip install A* : install the python library “A”
       - *pip install –r A.txt* : install python libraries listed in text file “A”
  - CIFAR-10 Dataset
    - http://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz
    - Unzip above file to (Assignment folder)/datasets
      - (Assignment folder)/datasets/cifar-10-batches-py

2-layer Neural Network - Practice

- 2-layer Neural Network
  - Fill the codes following the instruction in markdown cells
    - **two_layer_net.ipynb, classifier/neural_net.py**
  - There are "***#START OF YOUR CODE***" / "***#END OF YOUR CODE***" tags denoting the start and end of code sections you should fill out.



```
iteration 0 / 1000: loss 2.302954
iteration 100 / 1000: loss 2.302551
iteration 200 / 1000: loss 2.297649
iteration 300 / 1000: loss 2.259604
iteration 400 / 1000: loss 2.204187
iteration 500 / 1000: loss 2.118602
iteration 600 / 1000: loss 2.051566
iteration 700 / 1000: loss 1.988489
iteration 800 / 1000: loss 2.006616
iteration 900 / 1000: loss 1.951511
Validation accuracy:  0.287
```


Loss history
Loss
Classification accuracy history
Classification accuracy
train
val
Epoch