Lecture 0 Software Preparation

1. (Optional) Basics of Git

Github (https://github.com/) is perhaps the world largest open source repository to store, share and collaborate on codes (now purchased by Microsoft), and is based on Git (https://en.wikipedia.org/wiki/Git)-- a version control tool. For experienced programmers, of course command line is the most popular tool, while for beginners, Github Desktop (https://desktop.github.com/), is a good choice to start with and get familiar with the concepts.

Basic concepts of git:

- **Clone**: get a local copy of the folder (repository) from the cloud
- **Fetch and Pull**: download the updated repository from cloud and merge with the current local copy
- **Commit and Push**: upload the local changes to the cloud and merge with the repository there

2. Jupyter Notebook

The minimal element of notebook file (or building blocks) are the **cells**. For each cell, it has two **types**:

- Markdown: write the text,equations ...
- Code: write and excute the python code

For both cells, you need to run the cell to evaluate (either display or run the code)

This is the markdown type: **Hello World!** *Hello World*!

This is another section (header level 3)

inline equations with latex: $a^2 + b^2 = c^2$

separate-line equation:

$$a^2 + b^2 = c^2$$

In [ ]:

```python
# this is the code type, I am the comment in Python follwed by ## signs in the code cell

print('Hello World!') # in Python, both "" and '' are for strings

# to run this cell, press control+ enter (cursor in current cell after running)
# or shift+ enter (cursor in next cell after running)
```

For each cell, it also has two **modes**:

- Command mode: The edge color is blue. To enter this mode, press `Esc` on keyboard. This mode is for changing the property or global control of the cell (for example, insert more cells, change cell type from markdown to code)

- Edit mode: The edge color is green, and there is a pencil sign in the up-right corner of the menu bar. To enter this mode, press `Enter` or (double-click). This mode is for locally edit the content of the cell.

Useful keyboard shortcuts (of course you can always click the menu bar in the top) in **command mode** (Press `Esc` to enter):

- change from markdown to code: `Y`
- change from code to markdown: `M`
- insert cell above: `A`
- insert cell below: `B`
- delete cell: type `D` consecutively for two times, or `DD`
- undo delete cell: `Z`
- move up and down: up and down in keyboard

```
In [ ]: # this cell is for you to practice
```

```
In [ ]: 5+3
```

```
In [ ]: 5-3
```

```
In [ ]: 5*3 # multiplication
```

```
In [ ]: 5/3 # float division
```

```
In [ ]: 5//3 # integer division
```

```
In [ ]: 5%3 # modulo
```

```
In [ ]: 5**3 # important ! this is exponential in python -- double star  **
```

```
In [ ]: 5^3 # this is the bitwise operator in Python -- first difference you come across with
        matlab!
```

```
In [ ]: abs(-0.8) # built-in function -- absolute value
```

```
In [4]: help(abs) # help to see the function documentations
```

```
Help on built-in function abs in module builtins:

abs(x, /)
    Return the absolute value of the argument.
```

```
In [ ]: min(3,2) # built-in function -- minimal number
```

```
In [ ]: sqrt(5) # what?
```

```
In [ ]: cos(0) # what???
```

```
In [ ]: import math # the functions are in the math module of python! will introduce in later
        lectures
        math.sqrt(5)
```

```
In [ ]: math.cos(0)
```

```
In [ ]: 'hello'+' world' # concatenation of strings
```

```
In [ ]: "hello "*3 # repeat of strings
```

Further Readings :

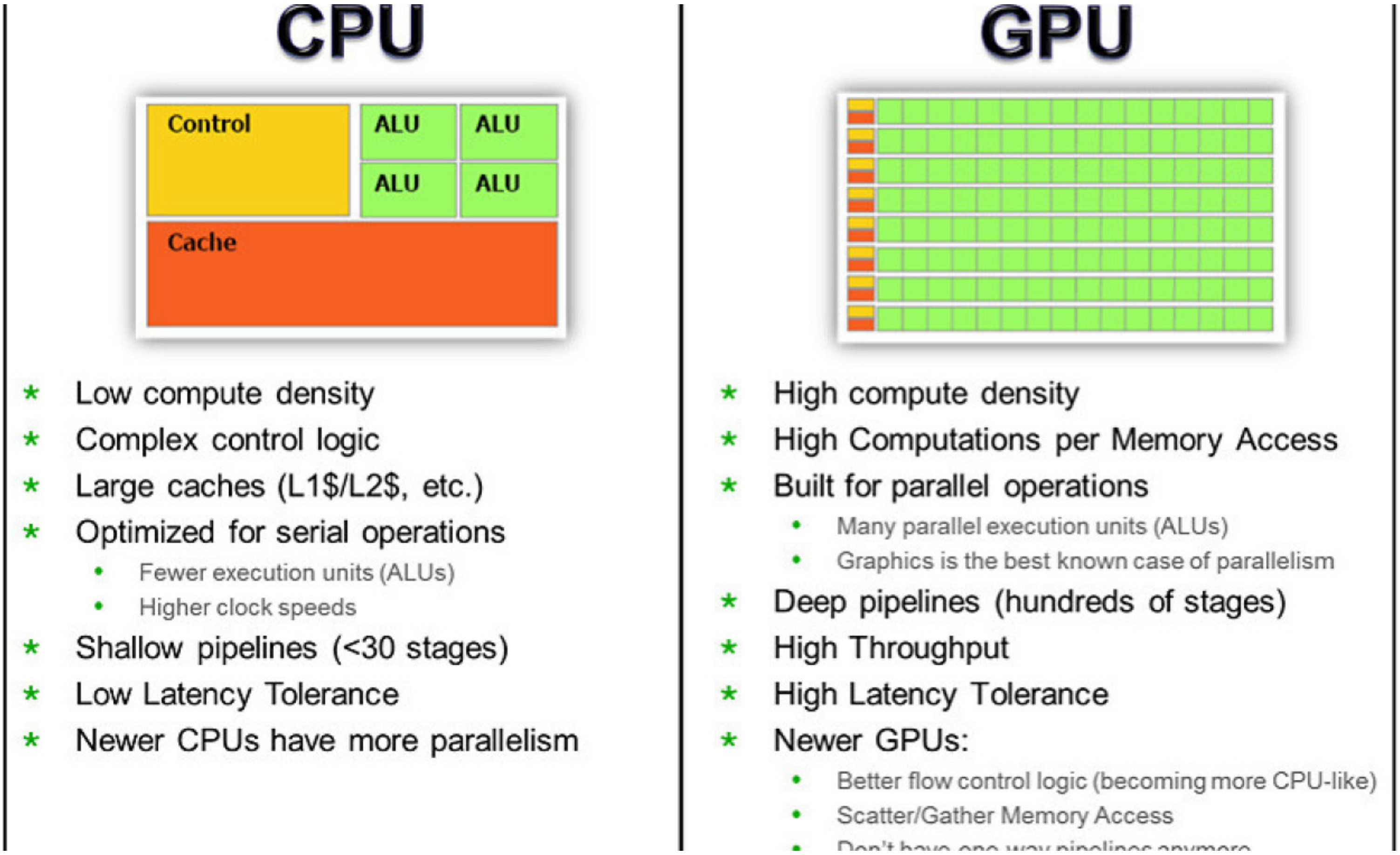
- Jupyter Notebook Shorcuts (https://towardsdatascience.com/jypyter-notebook-shortcuts-bf0101a98330)
- Markdown syntax (https://guides.github.com/features/mastering-markdown/)
- Latex syntax (https://www.math.ubc.ca/~pwalls/math-python/jupyter/latex/)

3. Basics of CPU and GPU

- CPU (https://en.wikipedia.org/wiki/Central_processing_unit): Central Processing Unit, good at complex operations, small number of processor cores. *Famous Manufacturer*: Intel and AMD

- GPU (https://en.wikipedia.org/wiki/Graphics_processing_unit): Graphics Processing Unit, good at simple operations, huge amount of processor cores. *Famous Manufacturer*: Intel (integrated cards), Nvidia and AMD (discrete cards). In machine learning application, the most popular cards are from Nvidia, partly because of its support for CUDA (https://en.wikipedia.org/wiki/CUDA) (programming interface to write codes with GPU).

Intuitive Understanding: CPU is like one college student who can do calculus, while GPU is like 100 elementary school students who only know basic arithmetics.



CPU

- Low compute density
- Complex control logic
- Large caches (L1$/L2$, etc.)
- Optimized for serial operations
  - Fewer execution units (ALUs)
  - Higher clock speeds
- Shallow pipelines (<30 stages)
- Low Latency Tolerance
- Newer CPUs have more parallelism

GPU

- High compute density
- High Computations per Memory Access
- Built for parallel operations
  - Many parallel execution units (ALUs)
  - Graphics is the best known case of parallelism
- Deep pipelines (hundreds of stages)
- High Throughput
- High Latency Tolerance
- Newer GPUs:
  - Better flow control logic (becoming more CPU-like)
  - Scatter/Gather Memory Access
  - Don't have one way pipelines anymore

It happens that GPU is very suitable for machine learning (especailly deep learning) tasks (easily speed up calculation by 10-100 times than CPU), and the fast development of GPU is one of the key contributor of machine learning popularity. See more introductions here (https://www.analyticsvidhya.com/blog/2017/05/gpus-necessary-for-deep-learning/).

Nowadays, beyond CPU and GPU, AI ASIC (https://en.wikipedia.org/wiki/Application-specific_integrated_circuit) (Application-specific integrated circuit) is becoming more and more popular. One of the most well-known example to general public is Google's TPU (https://en.wikipedia.org/wiki/Tensor_Processing_Unit) (Tensor Processing Unit), which is used in Alpha GO. Until today, it is still a "secret weapon" of Google, and you can use TPU from the google's cloud service, which can be 10 times more expensive than GPU (https://cloud.google.com/tpu/pricing). (Google Colab provides some out-dated TPU for free trial).

4. (Optional) Google Colab (https://colab.research.google.com/notebooks/intro.ipynb#scrollTo=lSrWNr3M and Kaggle Notebook (https://www.kaggle.com/code)

Both provide the free CPU and GPU resources and the notebook enviroment in the cloud.