Introduction to Deep Learning
Chapter 1: Introduction

Pao-Ann Hsiung

National Chung Cheng University

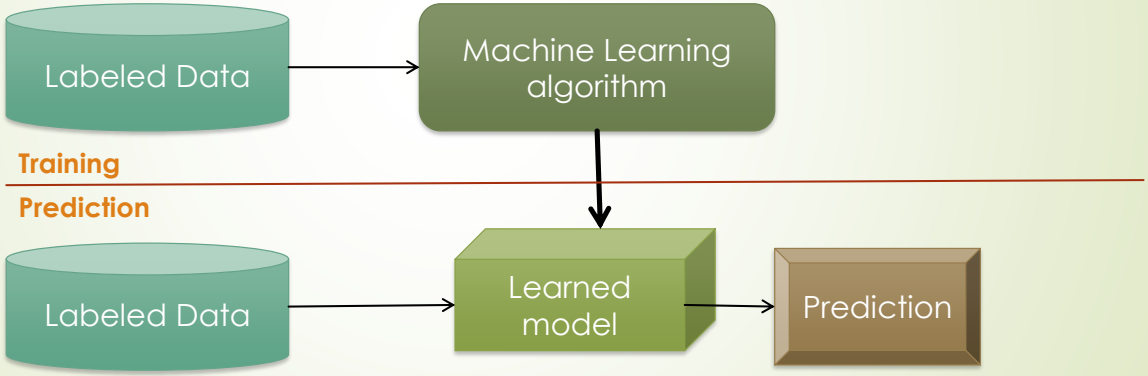
Contents

Chapter Goals -- Introduction

- To be knowledgeable about what exactly machine and deep learning is
- To be familiar with the history of deep learning,
- To be familiar with the applications of deep learning,
- To be familiar with the techniques required for deep learning
- To learn how to program in the Python programming language

What is machine learning (ML)?

- Machine learning is a field of computer science that gives computers the ability to learn without being explicitly programmed



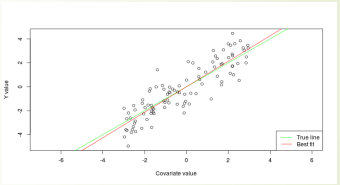
Methods that can learn from and make predictions on data

Types of Learning

Supervised: Learning with a **labeled training** set of data
Example: learn the *classification* of images based on image labels (dogs/cats, day time, numbers, etc.)

Classification

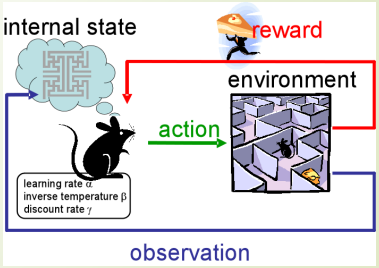
Unsupervised: Discover **patterns** in **unlabeled** data
Example: *cluster* similar documents based on text



Regression

Clustering

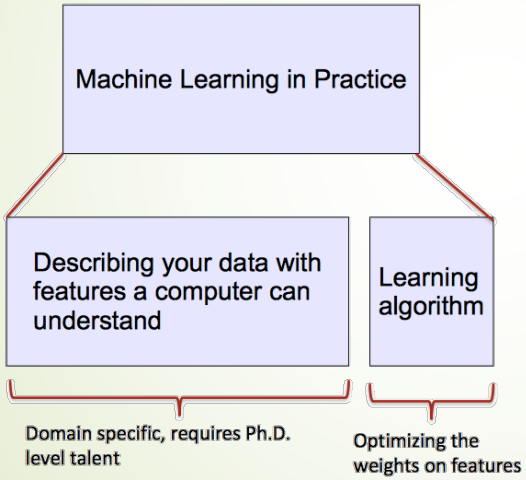
Reinforcement learning: learn to **act** based on **feedback/reward**
Example: learn to play Go, reward: *win or lose*



Sources: http://mbjoseph.github.io/2013/11/27/measure.html
https://becominghuman.ai/the-very-basics-of-reinforcement-learning-154f28a79071

Machine Learning in Practice

Most machine learning methods work well because of **human-designed representations** and **input features**
ML becomes just **optimizing weights** to best make a final prediction



Feature	NER
Current Word	✓
Previous Word	✓
Next Word	✓
Current Word Character n-gram	all
Current POS Tag	✓
Surrounding POS Tag Sequence	✓
Current Word Shape	✓
Surrounding Word Shape Sequence	✓
Presence of Word in Left Window	size 4
Presence of Word in Right Window	size 4

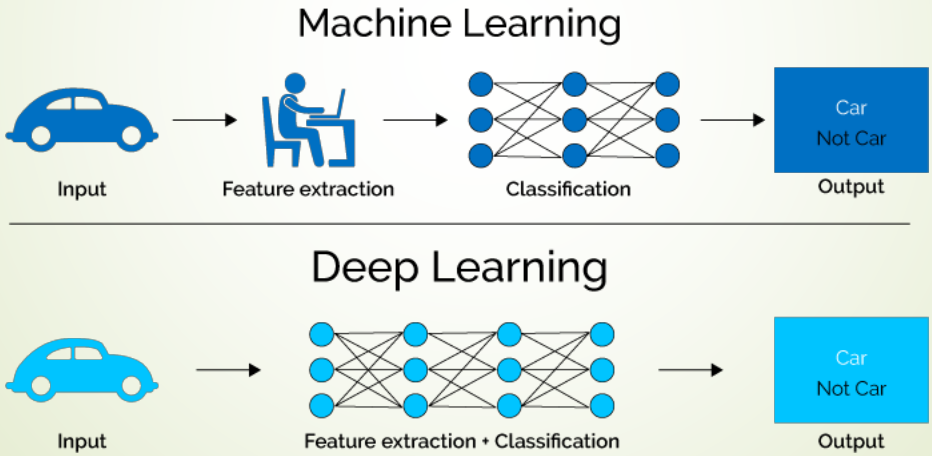
NER: Named Entity Recognition
POS: Part of Speech

Feature Hierarchy

- Hierarchy of representations with increasing levels of abstraction
- Image recognition
  - Pixel ➔ edge ➔ texton ➔ motif ➔ part ➔ object
- Text
  - Character ➔ word ➔ word group ➔ clause ➔ sentence ➔ story
- Speech
  - Sample ➔ spectral band ➔ sound ➔ … ➔ phone ➔ phoneme ➔ word

What is Deep Learning (DL) ?

- A sub-field of machine learning for learning **representations** of data.
- Exceptionally effective at **learning patterns**.
- Deep learning algorithms attempt to learn (multiple levels of) representation by using a **hierarchy of multiple layers**
- If you provide the system **tons of information**, it begins to understand it and respond in useful ways.

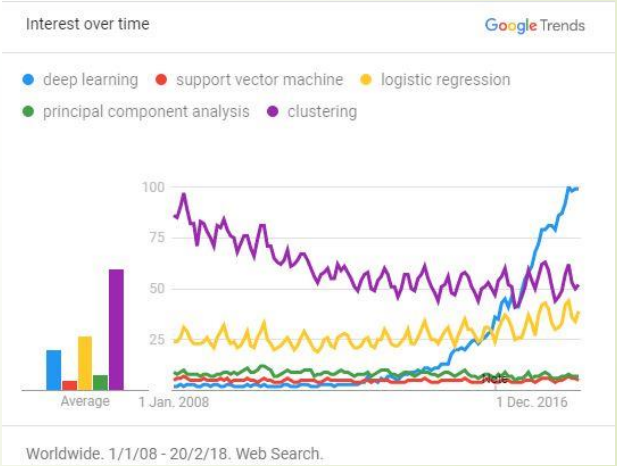


https://www.xenonstack.com/blog/static/public/uploads/media/machine-learning-vs-deep-learning.png

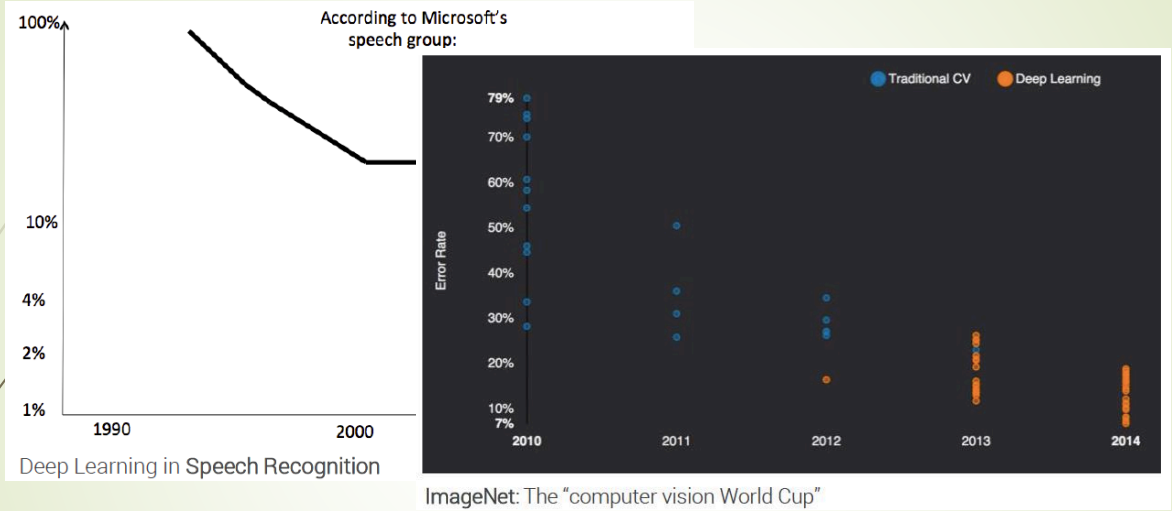
Why is DL useful?

- Manually designed features are often **over-specified**, **incomplete** and take a **long time to design** and validate
- Learned Features are **easy to adapt**, **fast** to learn
- Deep learning provides a very **flexible**, (almost?) **universal**, learnable framework for representing world, visual and linguistic information.
- Can learn in both unsupervised and supervised ways
- Effective **end-to-end** joint system learning
- Utilize large amounts of training data

Around 2010, DL started to outperform other ML techniques, first in speech and vision, then in Natural Language Processing (NLP)



State of the art in …



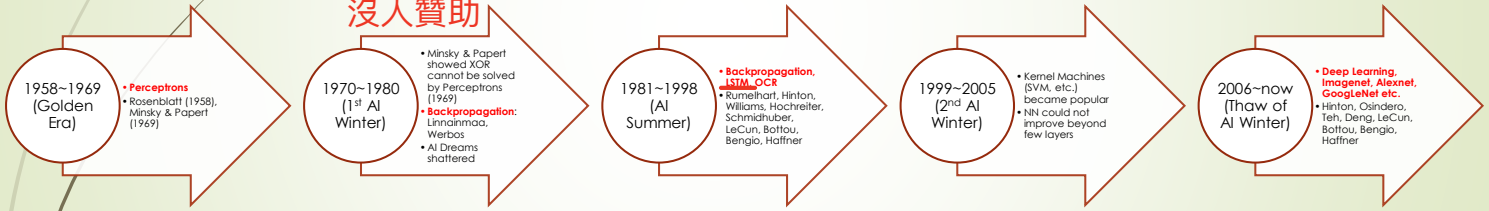
Several big improvements in recent years in NLP

- ✓ Machine Translation
- ✓ Sentiment Analysis
- ✓ Dialogue Agents
- ✓ Question Answering
- ✓ Text Classification …

Leverage different levels of representation

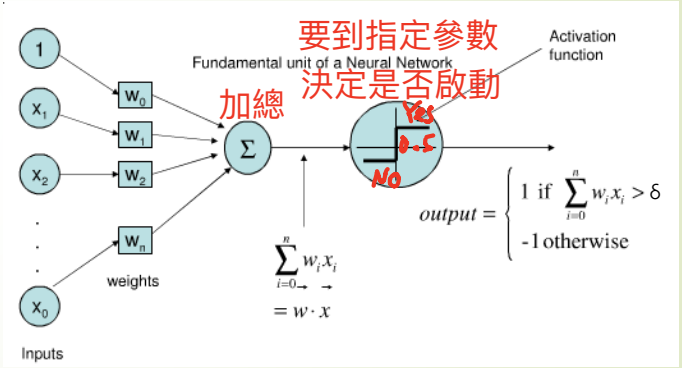
- words & characters
- syntax & semantics

History of neural networks & deep learning



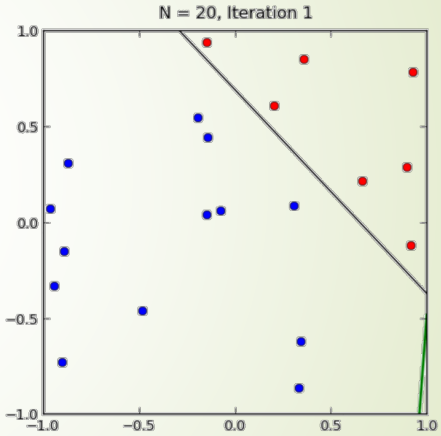
1958~1969: Birth of AI and Golden Era -- Perceptrons 單層神經

- Rosenblatt proposed a machine for binary classifications
- Main idea
  - One weight w_i per input x_i
  - Multiply weights with respective inputs and add bias w_0
  - If result is larger than threshold δ, return 1, otherwise 0.



Training a perceptron

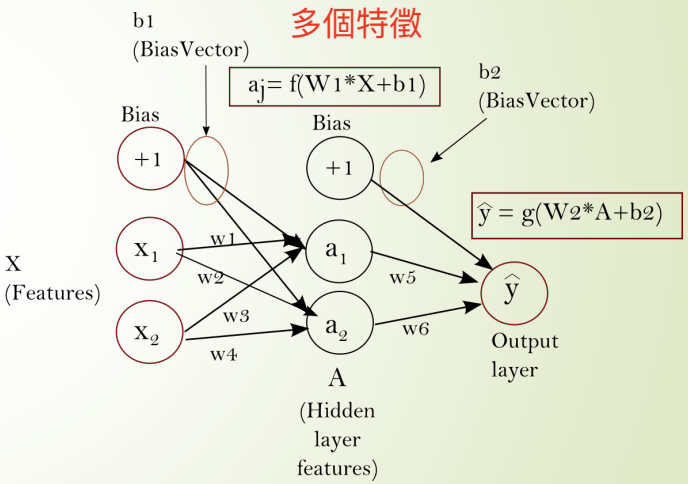
- **Learning Algorithm** by Rosenblatt
  - Initialize weights randomly
  - Take one sample x_i and predict y_i
  - For erroneous predictions, update weights
    錯誤要更改權重
    - If the output was $\hat{y} = 0$ and $y_i = 1$, increase weights
    - If the output was $\hat{y} = 1$ and $y_i = 0$, decrease weights
  - Repeat until no errors are made



Video (18'54"): https://youtu.be/OVHc-7GYRo4

Multi-Layer Perceptron

- One perceptron = one decision
- Question: What about multiple decisions?
  - Eg. Digit classification
- Answer: Neural Network (NN) or Multi-Layer Perceptron (MLP)
  - Stack multiple perceptrons (neurons) into a single layer
  - Connect two or more layers by feeding output of one layer as input to the next layer



1970~1980: 1st AI Winter

- XOR cannot be solved by Perceptron (Minsky)
- Perceptron training method cannot be applied to Neural Networks
- Funding slushed, Neural Networks were damned
- AI WINTER!!!
- Dreams shattered!
- Some significant results
  - Backpropagation: training method for NN (1970, 1974)

1981~1998: AI Summer

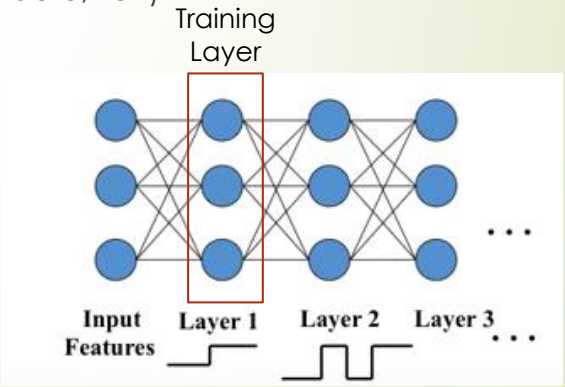
- XOR can be **solved** using Neural Networks (Multi-Layer Perceptron)
- **Backpropagation** (1986)
- Complicated NN architectures allowed
  - Temporal behavior, language, sequences could be learnt using "Recurrent Long-Short Term Memory (LSTM) Networks" in **1997** 連續性資料
  - Digit recognition in cheques (OCR) solved using Convolutional Neural Network (CNN) in **1998** 物件偵測

1999~2005: 2nd AI Winter

- Kernel Machines (e.g. Support Vector Machines (SVM), etc.) became popular 分類方法
  - Achieved similar accuracies
  - Included much fewer heuristics 探索的
  - Nice proofs on generalization 一般化
- Neural networks could not improve beyond a few layers
  - Lack of processing power (No GPUs)
  - Lack of data (No big, annotated datasets)
  - Overfitting (Models could not generalize) 過度訓練
  - Vanishing gradients (0.1*0.1*0.1*….*0.1 = 0.000000000001, too small for learning)
- AI community turned away from Neural Networks

2006~now: Thaw of AI Winter

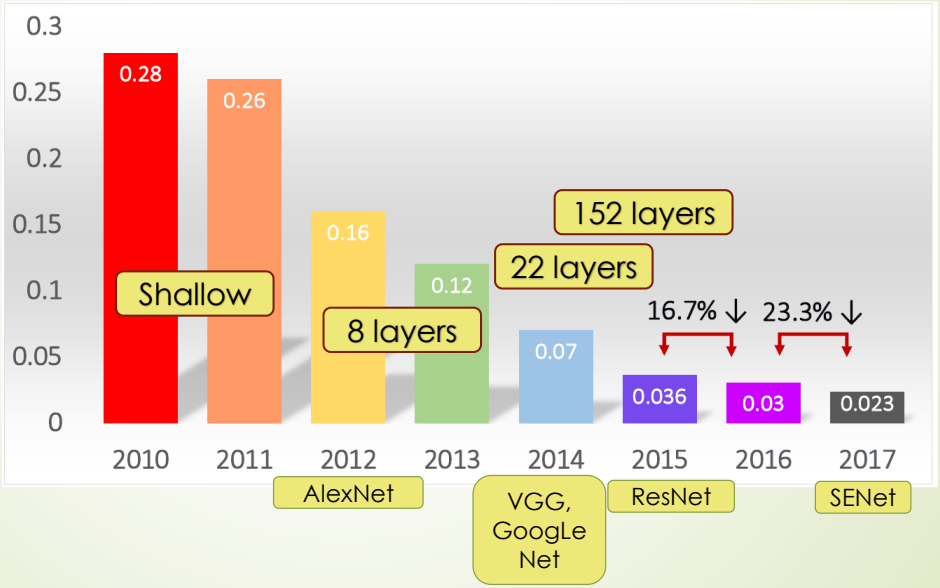
- Questions: 訓練層
  - Are 1-2 hidden layers the best NN can do?
  - Or, is it the learning algorithm not really mature?
- Deep Learning (2006, Hinton, Osindero, Teh)
- Layer-by-layer training
  - Per-layer trained parameters initialize further training using contrastive divergence
    對比的 分離



Deep Learning is here …

- **ImageNet** dataset (Deng et al, 2009)
  - Collected images for each term of Wordnet (100,000 classes)
  - Tree of concepts organized hierarchically
    - "Ambulance", "Dalmatian dog", "Egyptian cat", …
  - **Imagenet Large Scale Visual Recognition Challenge** (ILSVRC)
    - 1 million images
    - 1,000 classes
    - Top-5 and top-1 error measured
      - Errors reduced drastically in the past 8 years (2010~2017): 28.2% → 2.3%

ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

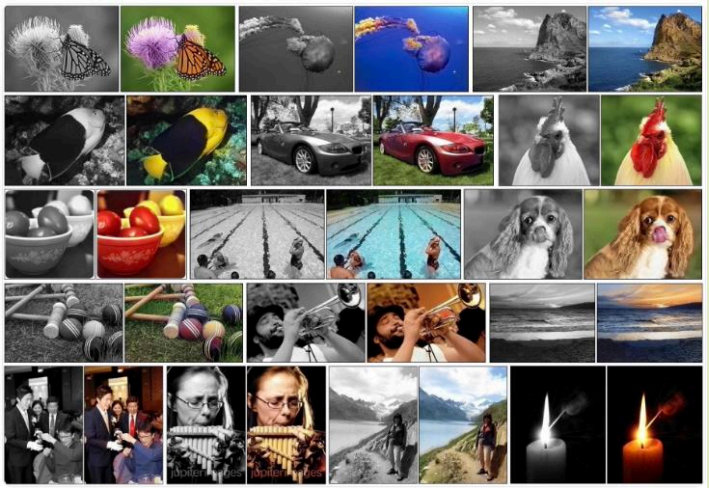


Some FUN now …

- The Neural Network ZOO
  - Graphical notations for all kinds of neural networks
  - http://www.asimovinstitute.org/neural-network-zoo/
- A Neural Network Playground
  - An online interactive way to play with different network architectures
  - http://playground.tensorflow.org
- 8 Inspirational Applications of Deep Learning
  - Very interesting applications of deep learning
  - http://machinelearningmastery.com/inspirational-applications-deep-learning/

(1/8) Automatic Colorization of B&W Images

- Large Convolutional Neural Networks (CNN) 物件辨識
- Website
  - http://richzhang.github.io/colorization/
- Video (5 s)
  - http://whattogive.com/videoColourization/



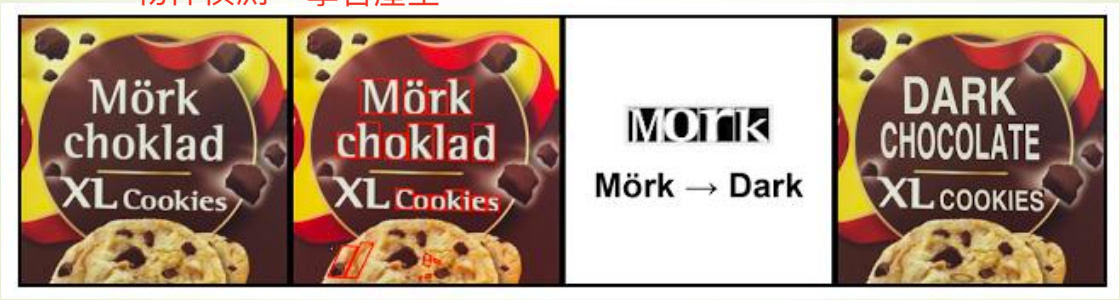
(2/8) Automatically Adding Sounds

- Two types of NN 時序性資料
  - Large CNN for images
  - Large Long Short-Term Memory (LSTM) Recurrent Neural Networks (RNN) for sound
- News
  - http://news.mit.edu/2016/artificial-intelligence-produces-realistic-sounds-0613
- Visually Indicated Sounds (MIT)
  - http://vis.csail.mit.edu/
- Video (2.54 s)
  - https://youtu.be/0FW99AQmMc8

(3/8) Automatic Machine Translation

- Automatic Translation of Text
  - Large Long Short-Term Memory (LSTM) Recurrent Neural Networks
- Automatic Translation of Images
  - CNN + LSTM RNN

物件偵測　學習產生



How Google Translate squeezes deep learning onto a phone!

- Demonstration

(4/8) Object Classification & Detection in Photographs

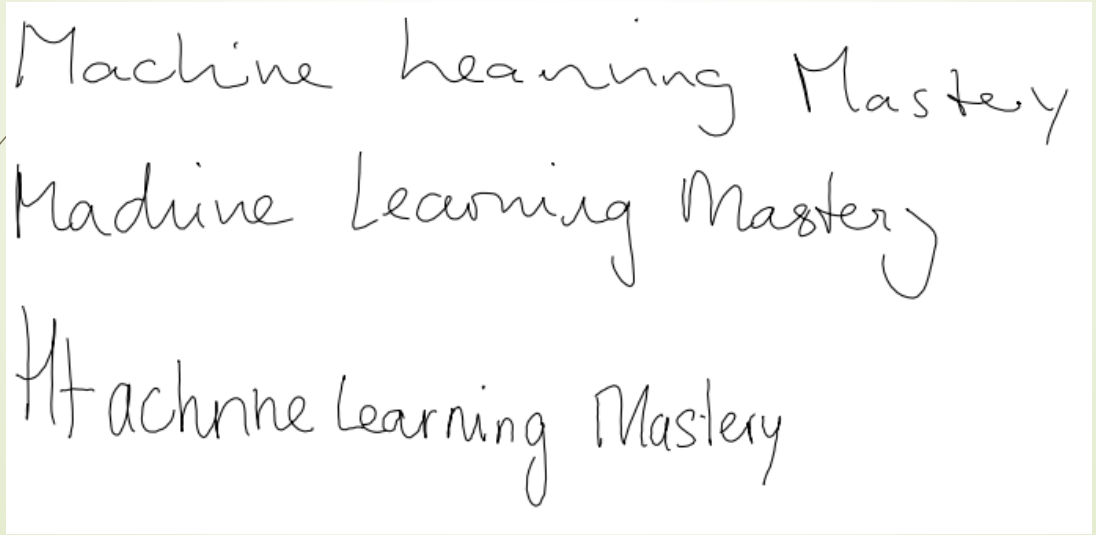
- Large deep CNN
- Paper on ImageNet Classification
  - http://www.cs.toronto.edu/~fritz/absps/imagenet.pdf

ConvNetJS: CIFAR-10 Demo

- ConvNetJS: CIFAR-10 Demo
  - http://cs.stanford.edu/people/karpathy/convnetjs/demo/cifar10.html
- Clarifai: 10,000 images
  - https://www.clarifai.com/

(5/8) Automatic Handwriting Generation

- Usage: used with forensic analysis



Automatic Handwriting Generation DEMO

- http://www.cs.toronto.edu/~graves/handwriting.html

(6/8) Automatic Text Generation

- Large RNN
- Code on Github
  - https://github.com/karpathy/char-rnn
- Paul Graham generator
- Shakespeare
- Wikipedia
- Algebraic Geometry (LaTeX)
- Linux source code
- Generating Baby Names

```
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.

Second Lord:
They would be ruled after this chamber, and
my fair nues begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.

Clown:
Come, sir, I will make did behold your worship.

VIOLA:
I'll drink it.
```

Automatic Music Synthesis

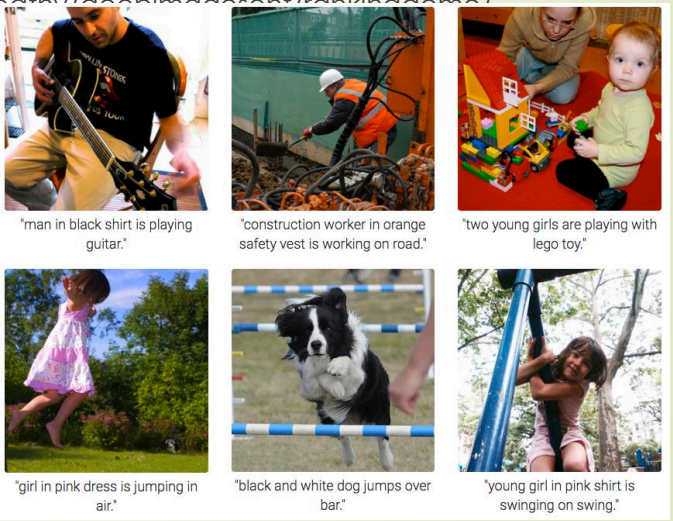
- Using large RNN
- Authors comments:
  - This track was made using a RNN. Fed 500 mb guitar tabs in ASCII. It writes the tabs out in ASCII, I imported into GuitarPro, recorded the output, imported that into FL Studio, added some filters and a drum loop and got this. The notes and rhythms themselves are totally unedited.
- Music (5:20):
  - https://soundcloud.com/optometrist-prime/recurrence-music-written-by-a-recurrent-neural-network

(7/8) Automatic Image Caption Generation

- Large CNN → Object Detection
- Large LSTM RNN → Caption Text Generation
- Deep Visual-Semantic Alignments for Generating Image Descriptions
  - http://cs.stanford.edu/people/karpathy/deepimagesent/

Automatic Image Captioning

- Demo
  - http://cs.stanford.edu/people/karpathy/deepimagesent/rankingdemo/



(8/8) Automatic Game Playing

- Vision
- Decision Making
- Etc.

Game Playing Demo

- https://youtu.be/TmPfTpjtdgg

AlphaGo

- AlphaGo by Google Deepmind Team
  - https://deepmind.com/research/alphago
- News:
  - AlphaGo program played against **Lee Sedol (18 times world champion)** in a tournament that took a week to play, five matches between March 9-15, 2016 at the Four Seasons Hotel in Seoul, South Korea.
  - It became the first program to beat a professional Go player, and in doing so showed an advancement of artificial intelligence that took the world by surprise.
  - Press Conference
    - https://youtu.be/yCALyQRN3hw?t=20728

AlphaGo in the news!

Google DeepMind

AlphaGo

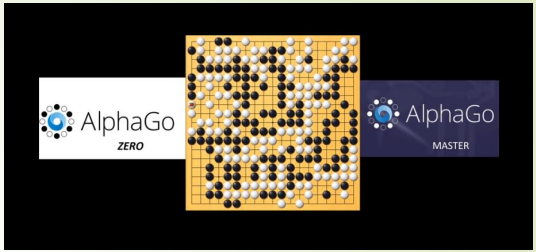
4:1

5:0

Alpha Go

- Used general purpose algorithms
  - Not a set of handcraft rules
  - Very different from Deep Blue (expert system)
- Modular system combining planning and pattern recognition
- Think like human
- Reinforcement and deep learning

AlphaGo Zero



- Alpha Go
  - Learn from large amounts of HUMAN PLAYS
  - Defeated the World Champion Lee Sedol from Korea
- AlphaGo Zero
  - Learning by itself (No human data)!
  - Defeated all previous versions of Alpha Go
  - Used less computation that previous versions of Alpha Go
  - Defeated the version of Alpha Go that defeated the World Champion (result: 100:1)
- About: https://www.youtube.com/watch?v=tXIM99xPQC8
- Nature Paper: https://www.nature.com/articles/nature24270
- AlphaGo Zero vs. AlphaGo Master Game1: https://youtu.be/-Wh4CfsWDyM
- AlphaGo Zero vs. AlphaGo Master Game2: https://youtu.be/xOVwmCOX7S4

Alpha Zero

- Not only Go!
  - Also Chess and Shogi
- Self-play using 5,000 first-generation TPUs to generate games, 64 second-generation TPUs to train the NN, all in parallel.
- Differences between AlphaGo Zero
  - Alpha Zero has hard-coded rules for setting search hyperparameters
  - NN is updated continually
  - Did not take advantage of symmetry (AlphaGo Zero used symmetry in Go)
  - Chess can end in a draw, AlphaZero takes draws into account.

Techniques required for DL

- Python Programming Language
- Deep Learning Frameworks
- Calculus

Python programming language

- Why Python?
- More Information on Python
- Installing Python
- Jupyter Notebooks
- Examples of running Python
- Running Python in Jupyter Notebooks

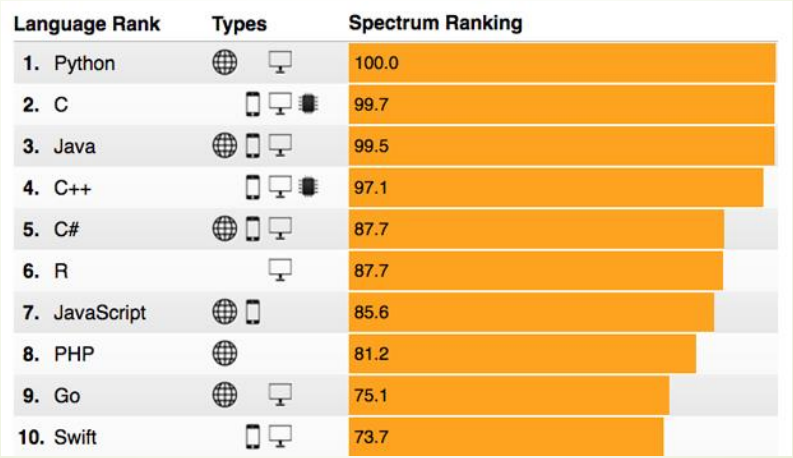
Python programming language

Guido van Rossum

- Conceived in late 1980's
- Implementation started in December 1989
- Creator: Guido van Rossum at CWI in the Netherlands (2005-2012 with Google, now with Dropbox)
- As successor to ABC programming language, with exception handling and for Amoeba OS
- Python named after BBC TV show Monty Python's Flying Circus

Why Python?

- **2017 Top Programming Languages** by IEEE Spectrum

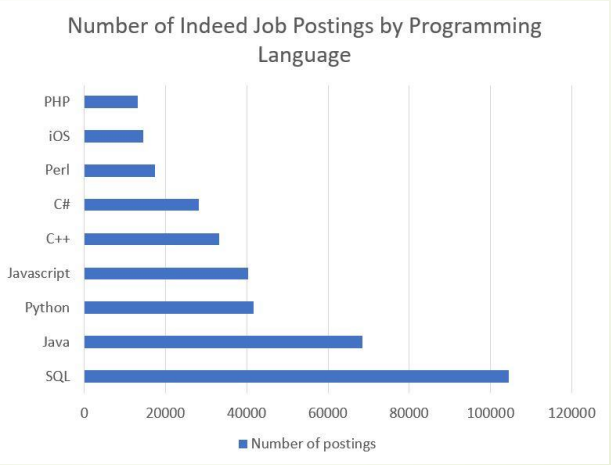


Language Rank	Types	Spectrum Ranking
1. Python		100.0
2. C		99.7
3. Java		99.5
4. C++		97.1
5. C#		87.7
6. R		87.7
7. JavaScript		85.6
8. PHP		81.2
9. Go		75.1
10. Swift		73.7

Source: https://spectrum.ieee.org/computing/software/the-2017-top-programming-languages

Why Python?

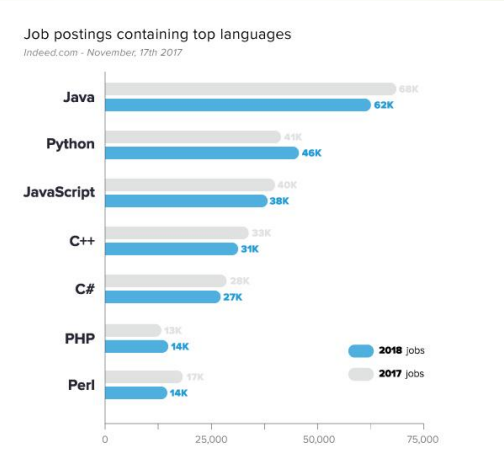
- **2017 Top Programming Languages by Number of Job Postings** by Coding Dojo Blog



http://www.codingdojo.com/blog/9-most-in-demand-programming-languages-of-2017/

Why Python?

- The **7 Most-in-Demand Programming Languages of 2018** by Coding Dojo Blog



Source: http://www.codingdojo.com/blog/7-most-in-demand-programming-languages-of-2018/

Why Python?

- **Most Pull Requests on GitHub** in 2017

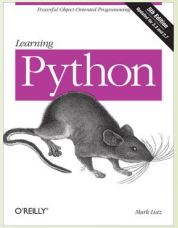
Source: https://stackify.com/popular-programming-languages-2018/

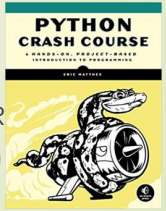
Why Python?

- **It is Popular**: 1 million Python users
- **It has NICE** features: Object-oriented, open-source, portable
- **It is POWERFUL:** dynamic typing, built-in types and tools, libraries, 3rd party utilities (Numeric, Numpy, SciPy, etc.), automatic memory management
- **It has W I D E applications**
  - System/GUI programming, Internet Scripting, Component Integration, Database, Gaming, Images, XML, Robot, etc.
  - Google: uses Python in web search system (Python's creator employed)
  - Intel, Cisco, HP, Seagate, Qualcomm, IBM: use Python for hardware testing
  - YouTube: video sharing services written in Python mostly

More Information on Python

- http://python.org/
  - Documentation, tutorials, beginners guide, core distribution, etc.
- Books
  - https://wiki.python.org/moin/PythonBooks
  - http://learning-python.com/about-lp5e.html
    - Learning Python by Mark Lutz, O'Reilly
- Online Python Popular Code Recipes
  - http://code.activestate.com/recipes/langs/python/ (author site)
  - http://www.dsf.unica.it/~fiore/LearningPython.pdf (e-book)
- Python Crash Course (e-book)
  - http://ap-n.us/books/Programming/Python%20Crash%20Course.pdf
- Videos
  - https://code.tutsplus.com/courses/introduction-to-python (Learn from SCR
  - https://www.tutorialspoint.com/python_online_training/index.asp (Course)





Installing Python

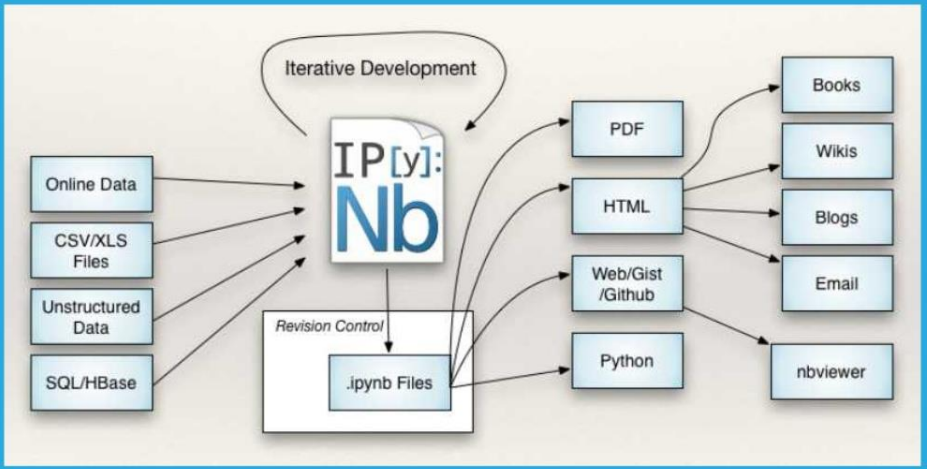
- https://www.python.org/downloads/
- Mainly TWO versions:
  - Python2 (currently 2.7.14)
  - Python3 (currently 3.6.4)
- Platforms supported: Windows, Linux/UNIX, Mac OS X, Others
- Beginners' guide
  - https://wiki.python.org/moin/BeginnersGuide/Programmers
- Which IDE to use for Python?
  - https://docs.google.com/spreadsheets/d/1I3x94P55qoxqYbq5GosWQ7IonZ4vR-4ZyCaImiVmCSk/pubhtml (there are lots of IDE, among which IDLE has a GUI)
  - In this course, you can use any as long as you can submit your work assignments
  - For labs, we will use Jupyter Notebooks

Jupyter Notebooks

- An open-source web application for **interactive** data science and scientific computing across **40 programming languages including Python, R, Julia and Scala …**
- Combines LIVE code and NARRATIVE text, equations, images, video, and visualizations.
- Can be shared on GitHub, Dropbox, and Jupyter Notebook Viewer
- Leverage big-data tools such as Apache Spark, from Python, R, and Scala. Explore data with pandas, scikit-learn, ggplot2, dplyr, etc.

Jupyter Notebooks

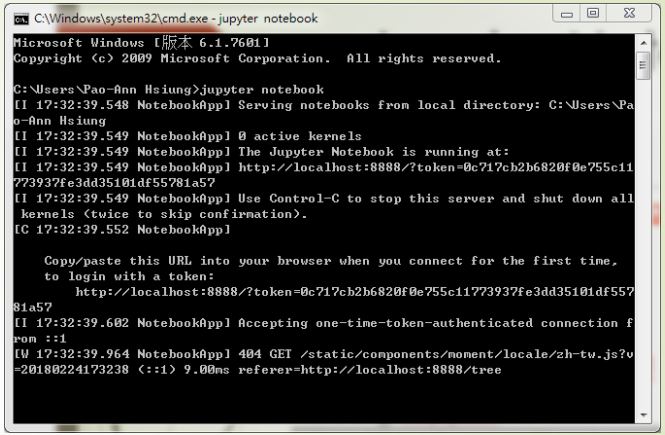
Notebook Workflows

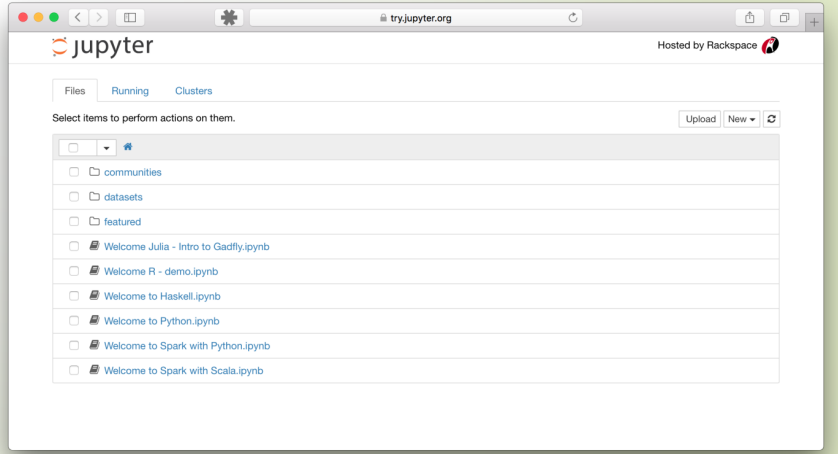


Credit: Joshua Barrat (via F. Perez)

Jupyter Notebooks

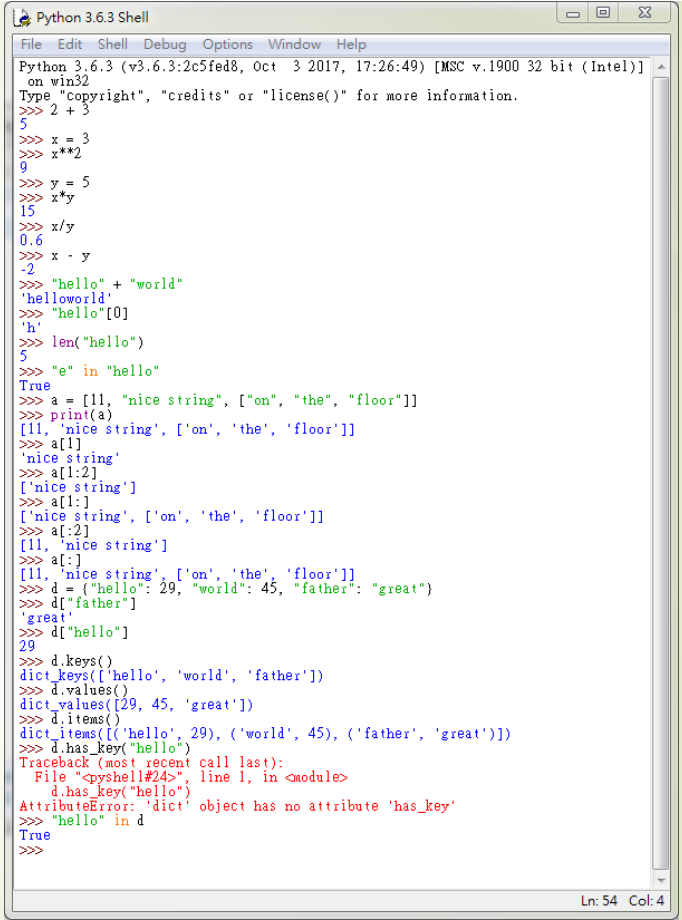
- Main page: http://jupyter.org
- Try it in your browser: https://try.jupyter.org/
- Install Jupyter: http://jupyter.org/install.html
- Running Jupyter Notebook: https://jupyter.readthedocs.io/en/latest/running.html#running





Examples of running Python

- Data types
  - Numbers, Strings, Lists, Dictionaries
- List items and slicing
- Dictionary keys and values

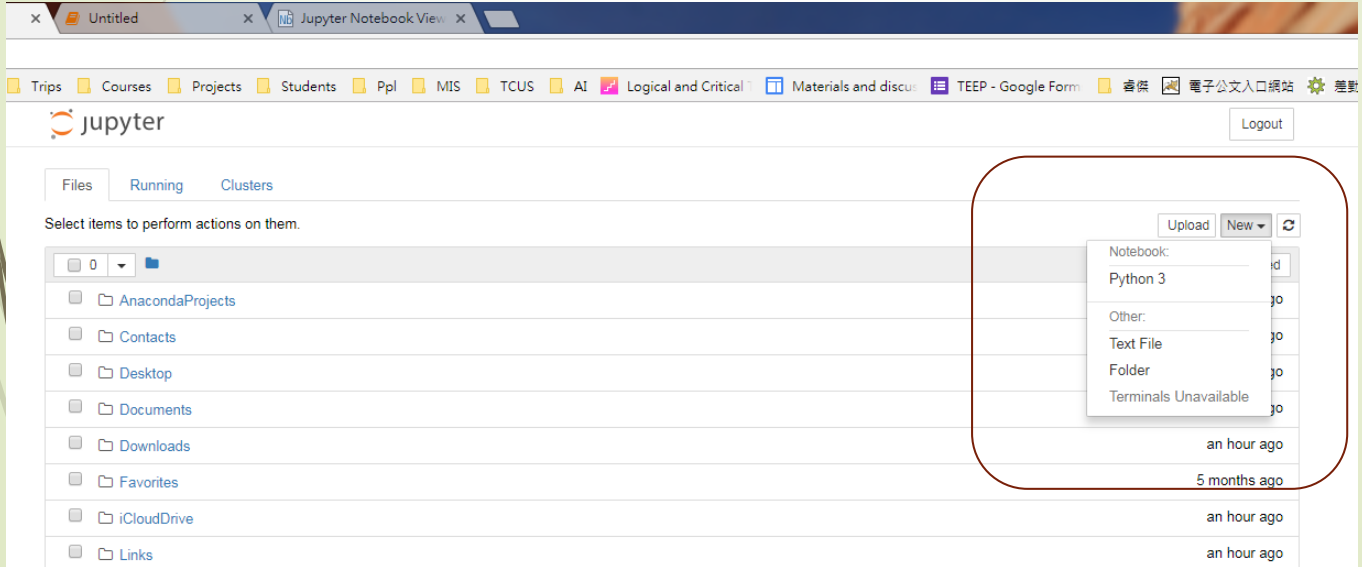


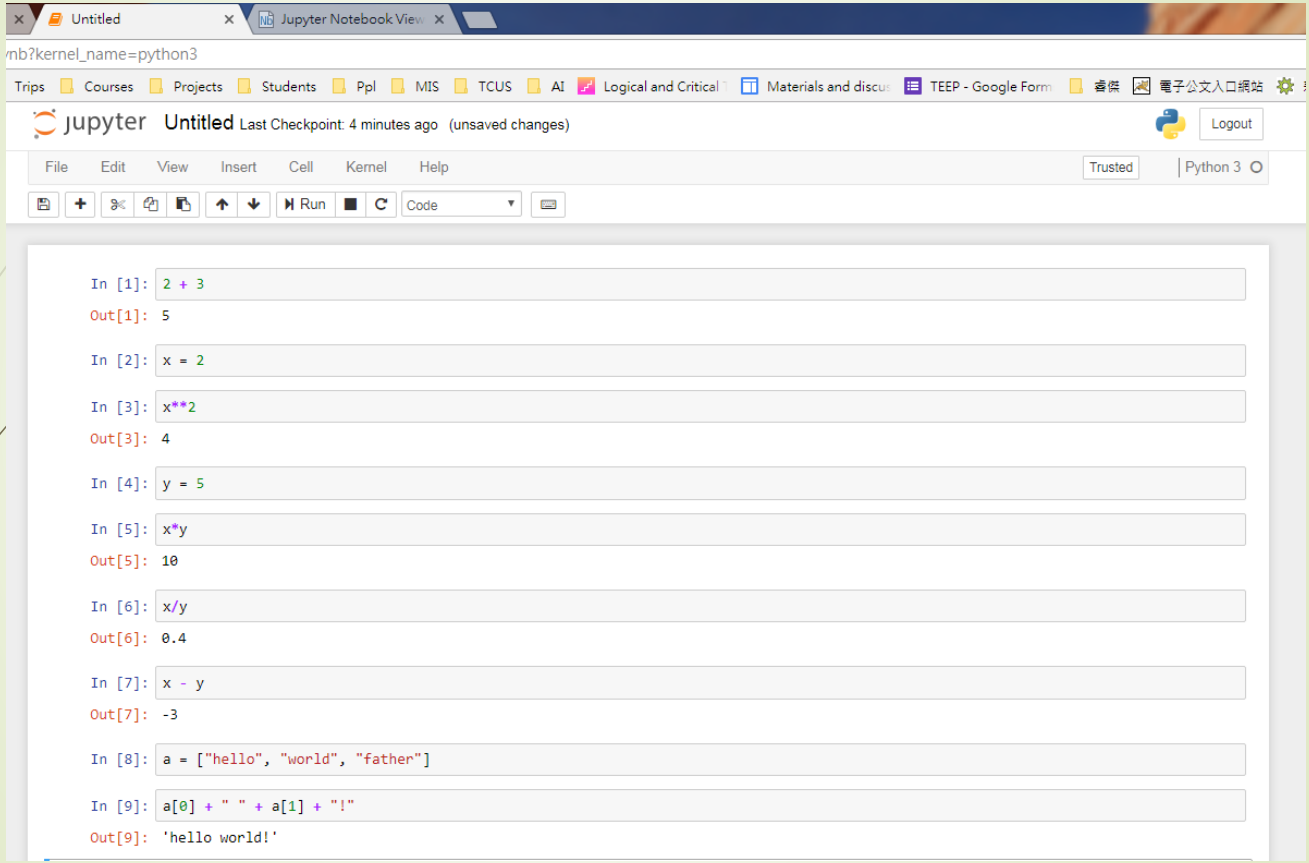
Running Python in Jupyter Notebooks

- A notebook is made up of a number of **cells**
  - Each cell can contain Python code, text, images, etc.
- You can **execute** a cell by clicking on it and pressing **Shift-Enter.**
  - The output of the cell will be displayed below.
- By convention, cells are to be run from top to bottom.
  - Failing to execute some cells or executing cells out of order can result in errors.
- More questions? Check out the following Jupyter Notebook Tutorial:
  - https://www.datacamp.com/community/tutorials/tutorial-jupyter-notebook

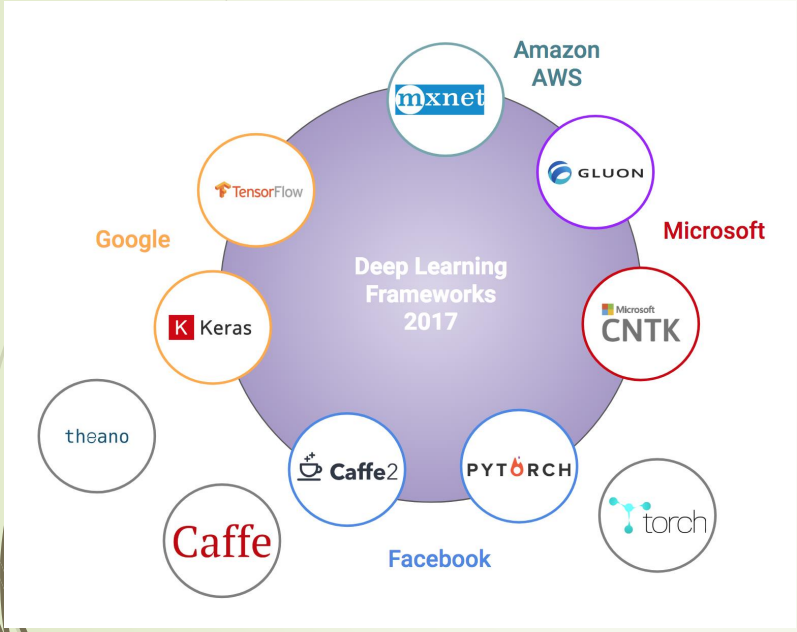
Examples of running Python in Notebook

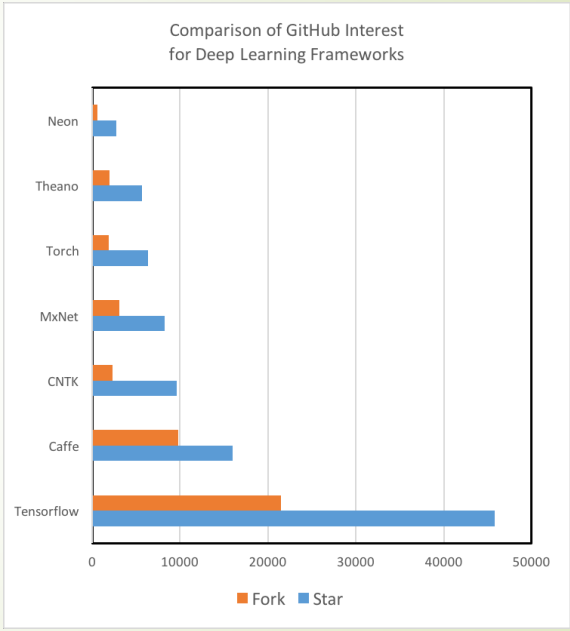
- Several Examples
  - https://github.com/jakevdp/PythonDataScienceHandbook/tree/master/notebooks




Untitled
Jupyter Notebook Viewer
nb?kernel_name=python3
Trips
Courses
Projects
Students
Ppl
MIS
TCUS
AI
Logical and Critical
Materials and discus
TEEP - Google Form
jupyter
Untitled Last Checkpoint: 4 minutes ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Help
Trusted
Python 3
Run
Code
In [1]: 2 + 3
Out[1]: 5
In [2]: x = 2
In [3]: x**2
Out[3]: 4
In [4]: y = 5
In [5]: x*y
Out[5]: 10
In [6]: x/y
Out[6]: 0.4
In [7]: x - y
Out[7]: -3
In [8]: a = ["hello", "world", "father"]
In [9]: a[0] + " " + a[1] + "!"
Out[9]: 'hello world!'

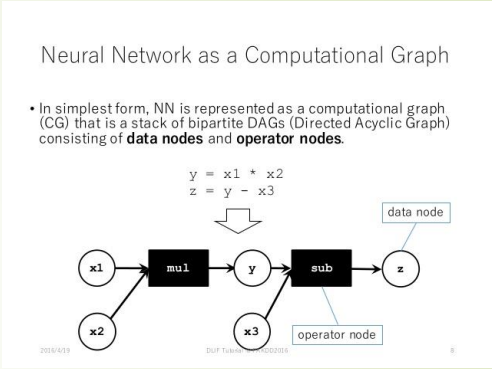
Deep learning frameworks





Deep learning frameworks

- Easily build big computational graphs
- Easily compute gradients in computational graphs
- Run it all efficiently on GPU



https://en.wikipedia.org/wiki/Comparison_of_deep_learning_software

http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture8.pdf

https://towardsdatascience.com/a-survey-of-deep-learning-frameworks-43b88b11af34

Comparisons in modeling capabilities

https://www.kdnuggets.com/2017/03/getting-started-deep-learning.html

	Languages	Tutorials and training materials	CNN modeling capability	RNN modeling capability	Architecture: easy-to-use and modular front end	Speed	Multiple GPU support	Keras compatible
Theano	Python, C++	++	++	++	+	++	+	+
Tensor-Flow	Python	+++	+++	++	+++	++	++	+
Torch	Lua, Python (new)	+	+++	++	++	+++	++	
Caffe	C++	+	++		+	+	+	
MXNet	R, Python, Julia, Scala	++	++	+	++	++	+++	
Neon	Python	+	++	+	+	++	+	
CNTK	C++	+	+	+++	+	++	+	

Comparisons in code

Numpy

```
import numpy as np
np.random.seed(0)

N, D = 3, 4

x = np.random.randn(N, D)
y = np.random.randn(N, D)
z = np.random.randn(N, D)

a = x * y
b = a + z
c = np.sum(b)

grad_c = 1.0
grad_b = grad_c * np.ones((N, D))
grad_a = grad_b.copy()
grad_z = grad_b.copy()
grad_x = grad_a * y
grad_y = grad_a * x
```

TensorFlow

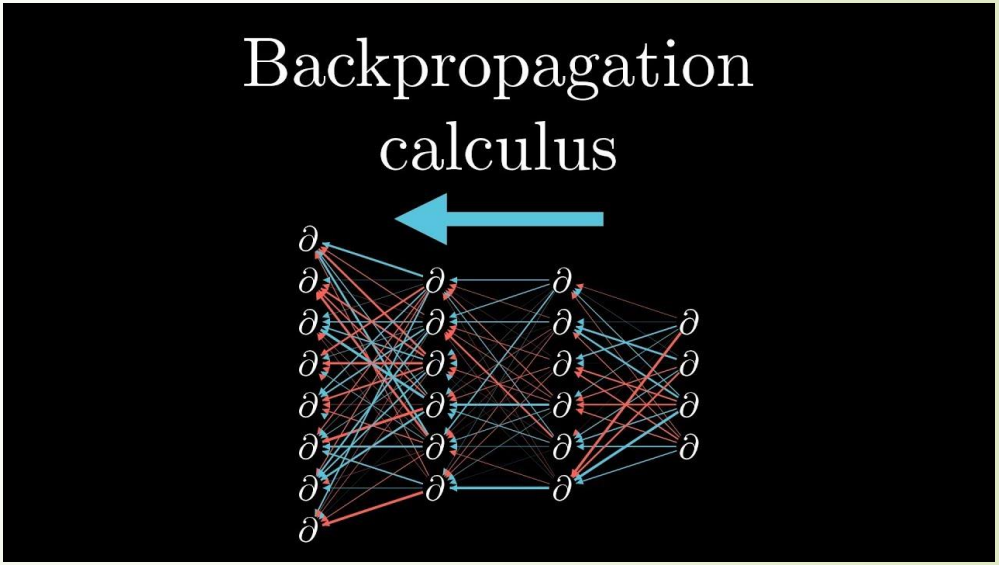
```
import numpy as np
np.random.seed(0)
import tensorflow as tf

N, D = 3, 4

with tf.device('/gpu:0'):
    x = tf.placeholder(tf.float32)
    y = tf.placeholder(tf.float32)
    z = tf.placeholder(tf.float32)

    a = x * y
    b = a + z
    c = tf.reduce_sum(b)

grad_x, grad_y, grad_z = tf.gradients(c, [x, y, z])

with tf.Session() as sess:
    values = {
        x: np.random.randn(N, D),
        y: np.random.randn(N, D),
        z: np.random.randn(N, D),
    }
    out = sess.run([c, grad_x, grad_y, grad_z],
                   feed_dict=values)
    c_val, grad_x_val, grad_y_val, grad_z_val = out
```

PyTorch

```
import torch
from torch.autograd import Variable

N, D = 3, 4

x = Variable(torch.randn(N, D).cuda(),
             requires_grad=True)
y = Variable(torch.randn(N, D).cuda(),
             requires_grad=True)
z = Variable(torch.randn(N, D).cuda(),
             requires_grad=True)

a = x * y
b = a + z
c = torch.sum(b)

c.backward()

print(x.grad.data)
print(y.grad.data)
print(z.grad.data)
```

Calculus

- Partial derivatives
- Chain rule



https://www.youtube.com/watch?v=tIeHLnjs5U8

What we learnt?

- Bonus points for correct and accurate answers to the following questions:
  - What is machine learning?
  - What is the difference between machine and deep learning?
  - Can CNN and RNN be combined into one neural network architecture? If yes, for what kind of applications?
  - Given the following Python code:

```
>>> def f4(x,a=1):
...     return a*x**2
...
>>>
```

  - What is the value of **f4(2)**?
  - What is the value of **f4(2, 2)**?

What to do now?

- Install Python
- Install Jupyter
- Play with Jupyter Notebooks and Python
- Next class on March 6 (Tuesday) 13:15 will be in the **COMPUTER ROOM R341 in the INNOVATION BUILDING**!!!
  - We will have a lab in the computer room for only 75 minutes
  - The second class 14:45~16:00 will still be conducted in room EA204 here!