

深度學習程式設計

李龍豪 (Lung-Hao Lee)

lhlee@ee.ncu.edu.tw

LMS

- <https://lms.ncu.edu.tw/>

The screenshot shows the LMS interface for the course EE3054. The top navigation bar displays the course title "EE3054-深度學習程式設計 Programming for Deep Learning-*-1082". Below the navigation bar, there are two tabs: "公開首頁" (Public Home) and "成員首頁" (Member Home), with "成員首頁" currently selected. A blue box highlights the "成員首頁" tab. On the left, a sidebar lists various tools and resources: 完整選單 (Full Menu), 主頁 (Home), 公告 (Announcements), 課程大綱 (Course Outline), 關於班級 (About Class), 教師工具 (Teacher Tools), 班級管理工具 (Class Management Tools), 應用程式 (Applications), and 系統工具 (System Tools). The "關於班級" item is highlighted with an orange box. The main content area shows course information: 課程資訊 (Course Information), 行事曆 (Calendar), 討論區 (Discussion Zone), 部落格 (Blog), 共用文件 (Shared Files), 任務 (Tasks), and 相簿 (Album). The "共用文件" item is highlighted with a blue box. On the right, the National Central University logo and name are displayed, along with the text "數位學習平台" (Digital Learning Platform) and "Learning Management System". Below the logo, it says "老師" (Teacher) and lists the course details: EE3054-* 深度學習程式設計 Programming for Deep Learning.

- This course is

Programming for Deep Learning

NOT

Learning Programming Deeply

What is Python?

<https://www.python.org/>



- Executive Summary

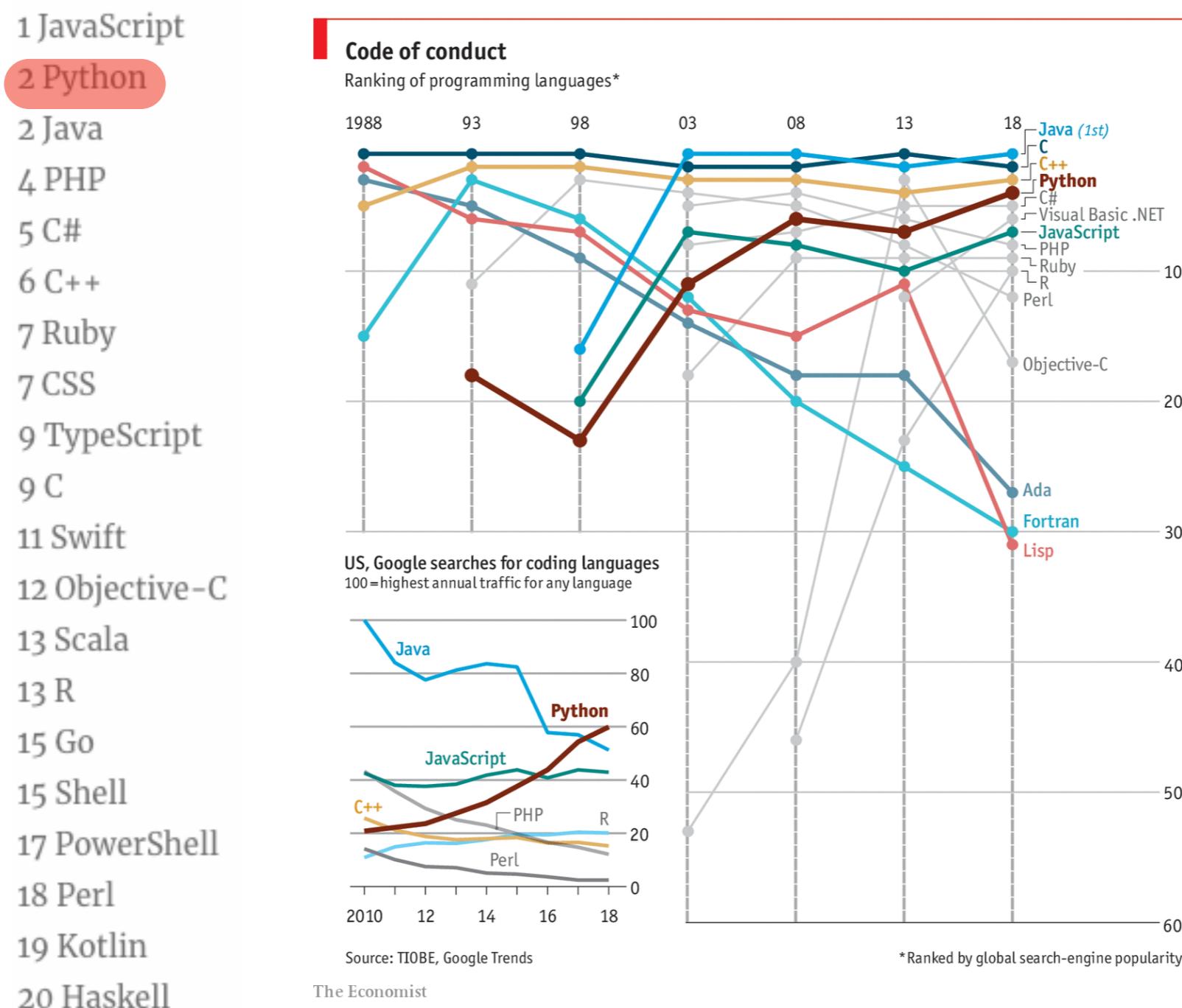
<https://www.python.org/doc/essays/blurb/>

Python is an **interpreted**, **object-oriented**, **high-level** programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for rapid application development.

Python's **simple**, **easy to learn** syntax emphasizes readability and therefore reduces the cost of program maintenance.

Why Python?

- The RedMonk Programming Language Rankings: January 2020
<https://redmonk.com/sogrady/2020/02/28/language-rankings-1-20/>



The Zen of Python

- “Pythonic” idioms

```
In [1]: import this
```

The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!

```
# Python 3: Fibonacci series up to n  
>>> def fib(n):  
>>>     a, b = 0, 1  
>>>     while a < n:  
>>>         print(a, end=' ')  
>>>         a, b = b, a+b  
>>>     print()  
>>> fib(1000)  
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610  
987
```

What makes Python the best programming for AI?

<https://steelkiwi.com/blog/python-for-ai-and-machine-learning/>

- Simple and Consistent
- Extensive selection of libraries and frameworks
 - Keras, Tensorflow, and Scikit-learn for machine learning
 - Numpy for high-performance scientific computing and data analysis
 - Pandas for data manipulation
 - Matplotlib for data visualization
- Platform independence
- Great community and popularity



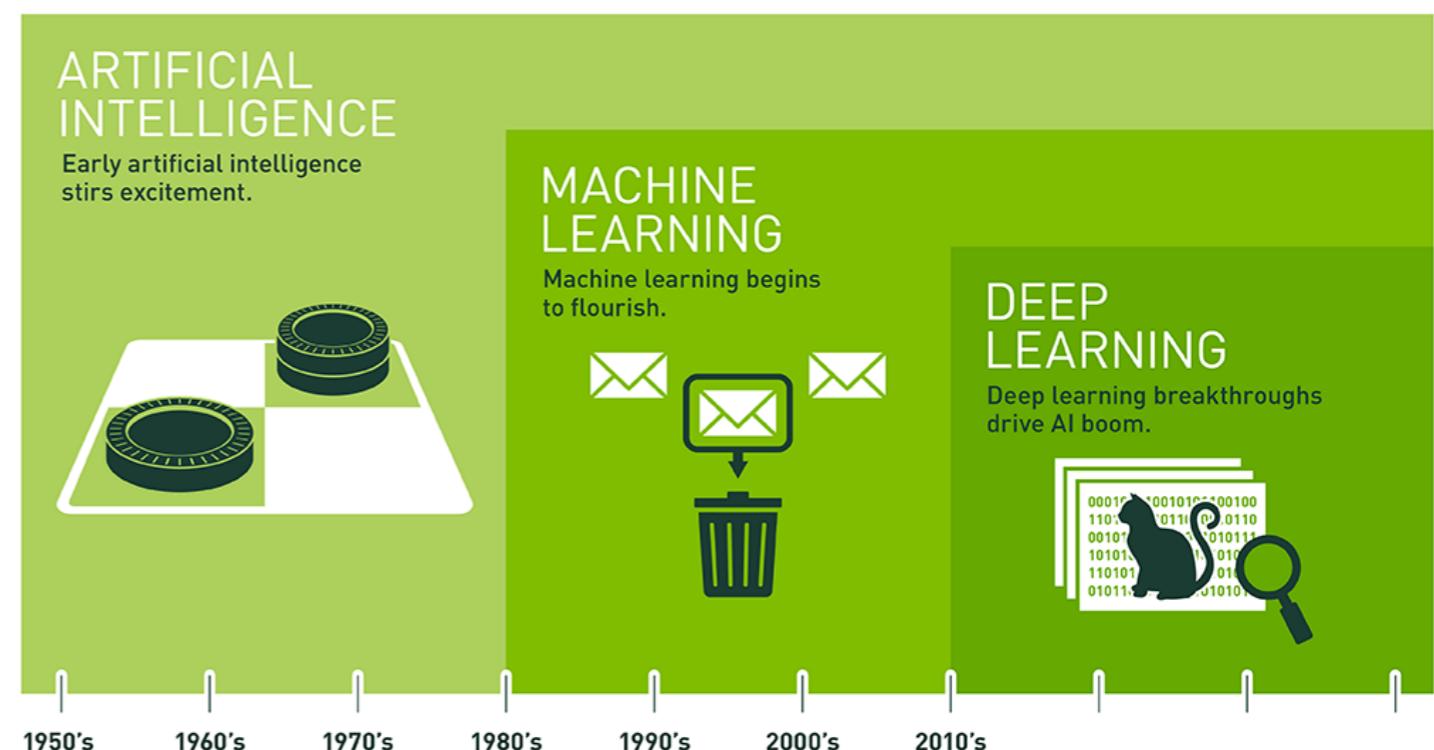
What's Deep Learning?

- From Wikipedia
https://en.wikipedia.org/wiki/Deep_learning

Deep learning is part of a broader family of **machine learning** methods based on **artificial neural networks** with representation learning.

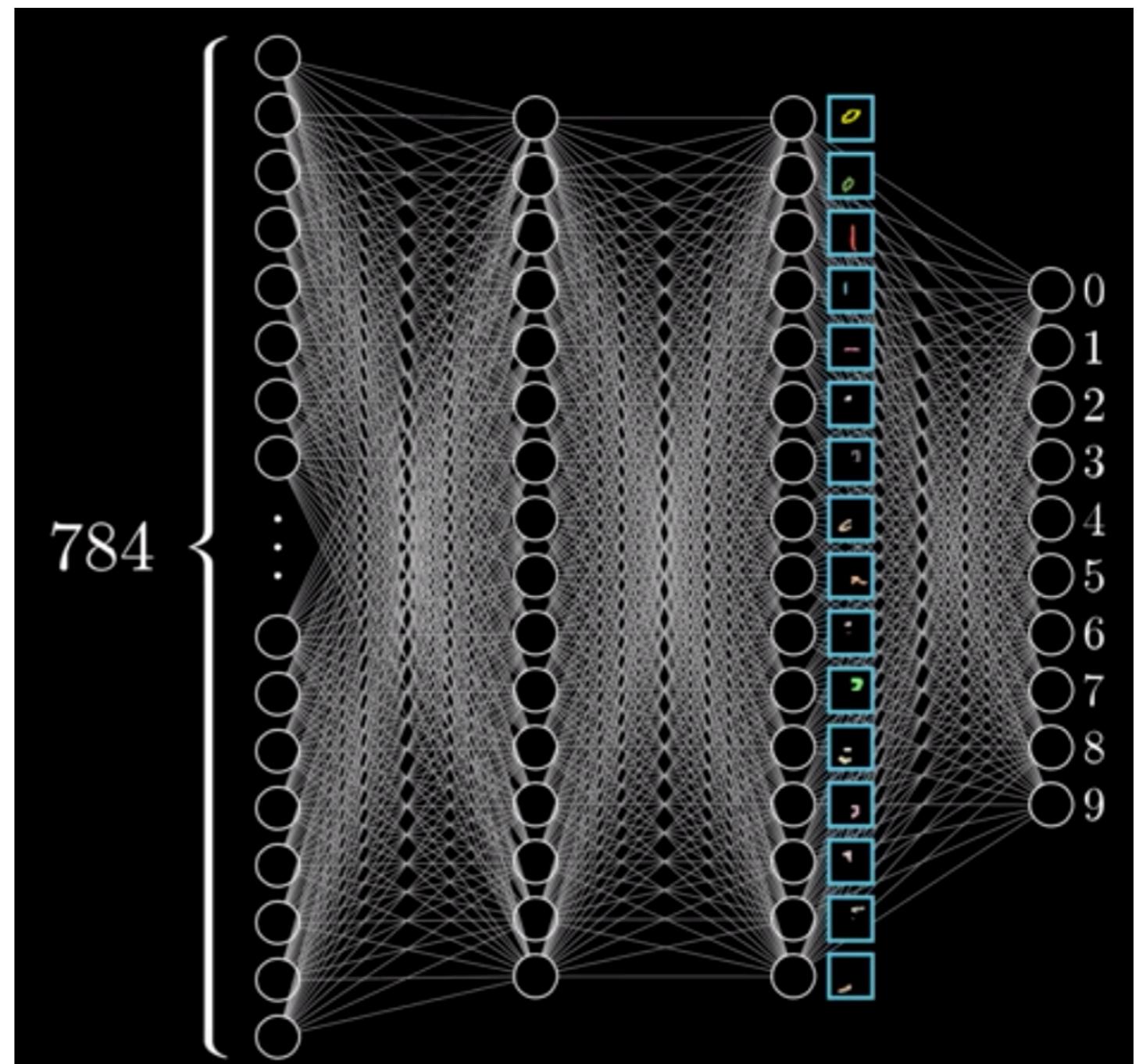
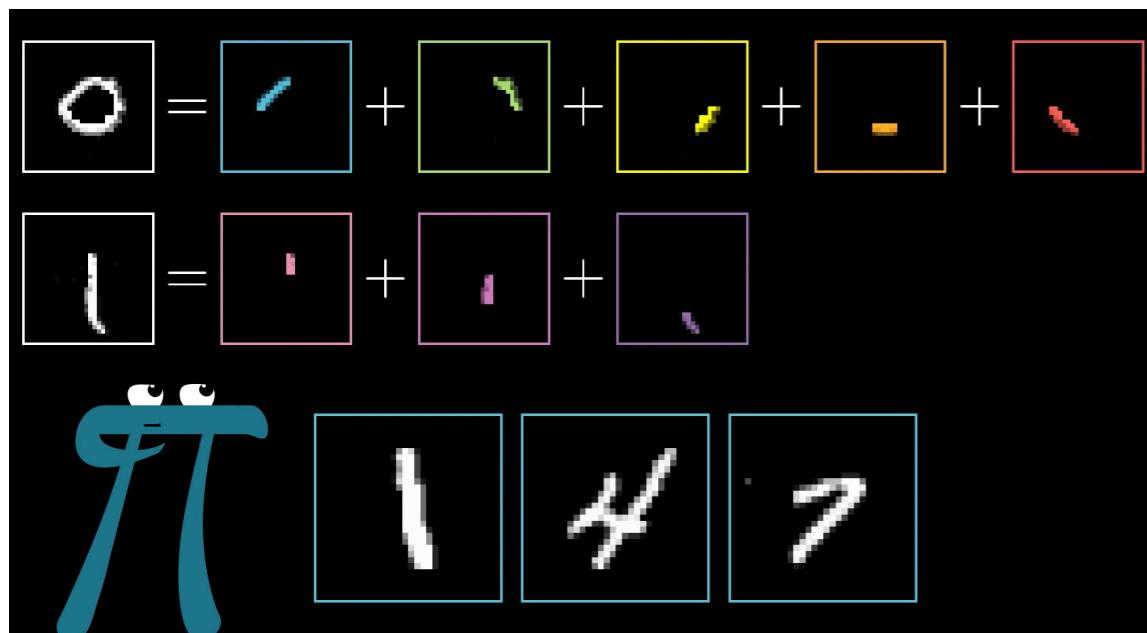
AI [off | on]

- ▶ Artificial intelligence
- ▶ Computer vision
- ▶ Machine learning & data mining
- ▶ Natural language processing
- ▶ The Web & information retrieval



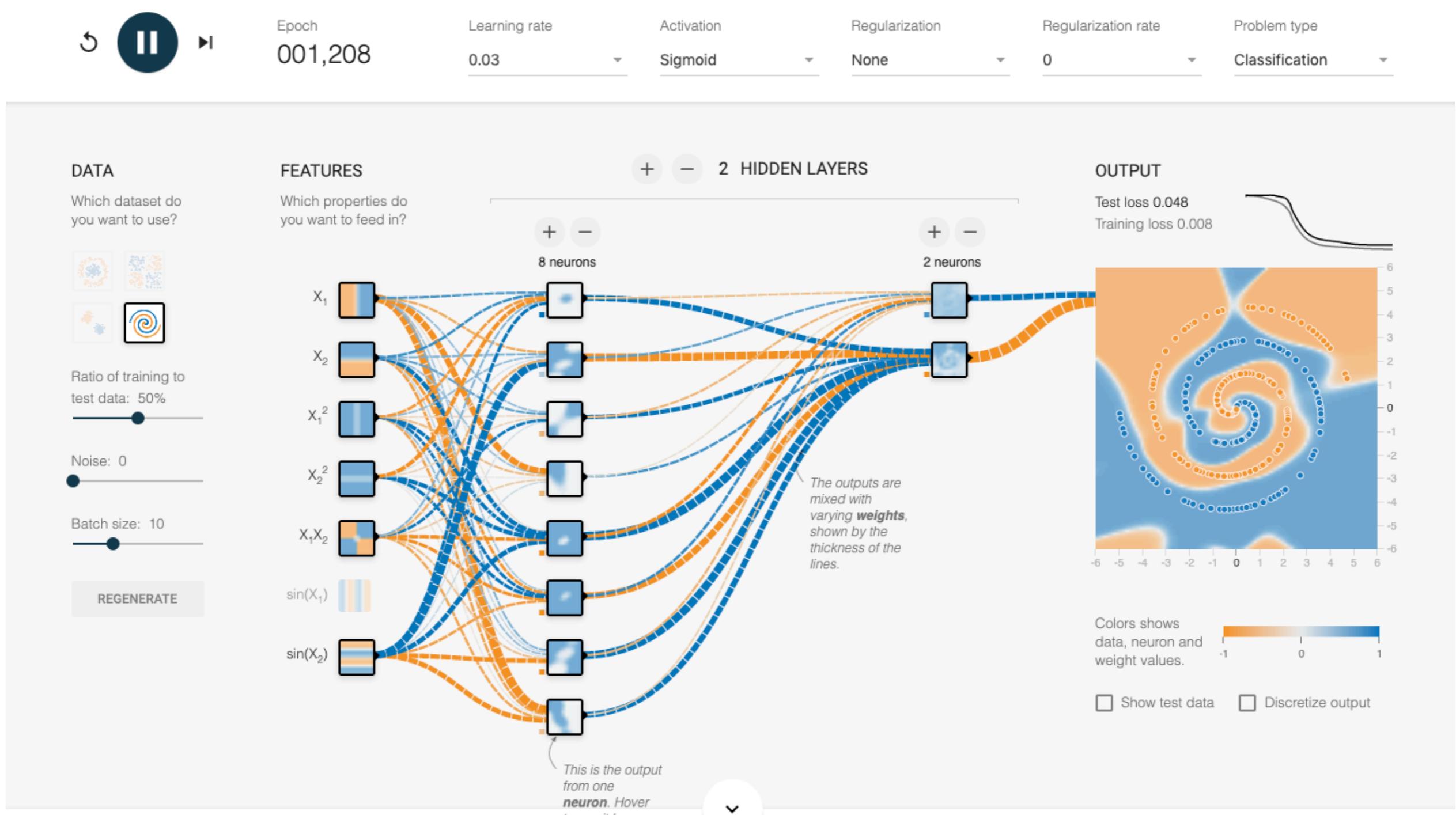
But What's a Neural Network?

<https://www.youtube.com/watch?v=aircArUvnKk>

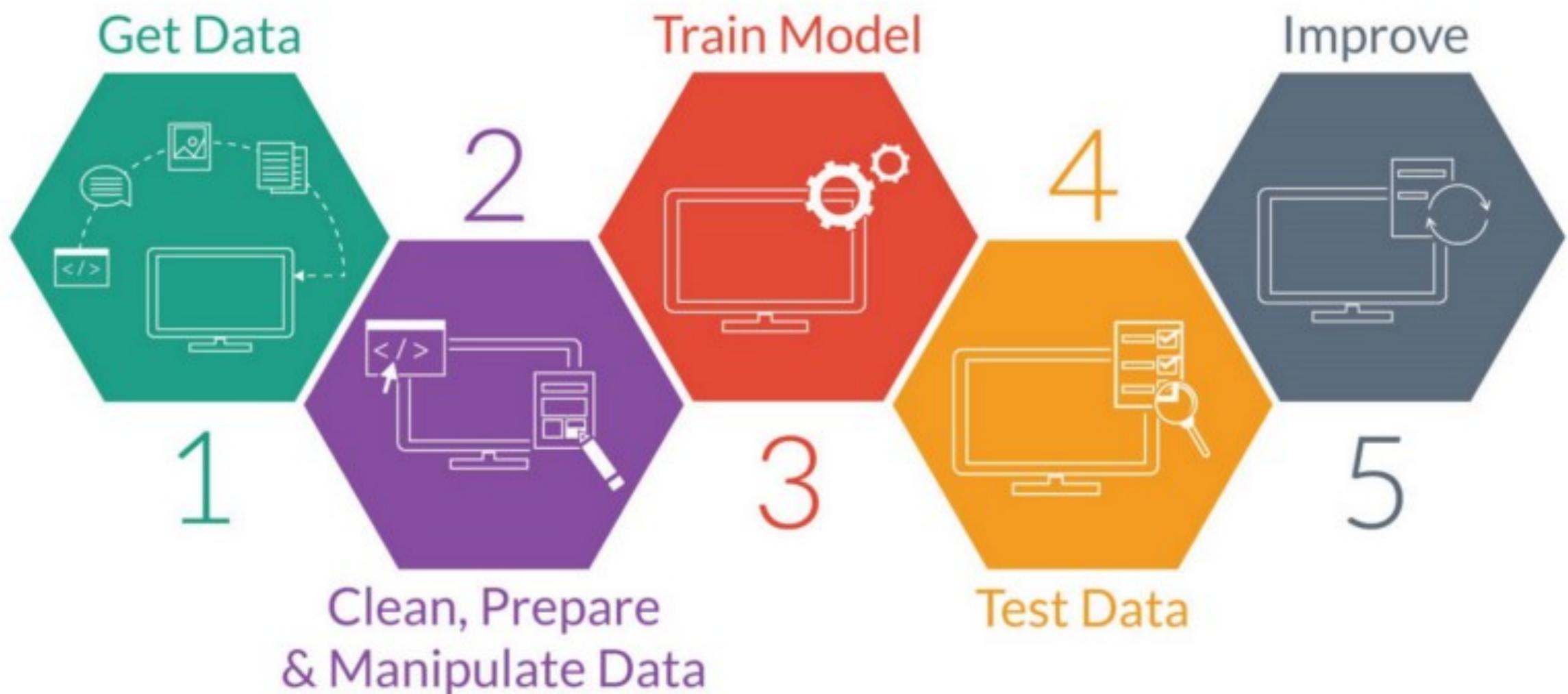


A Neural Network Playground - TensorFlow

<https://playground.tensorflow.org/>

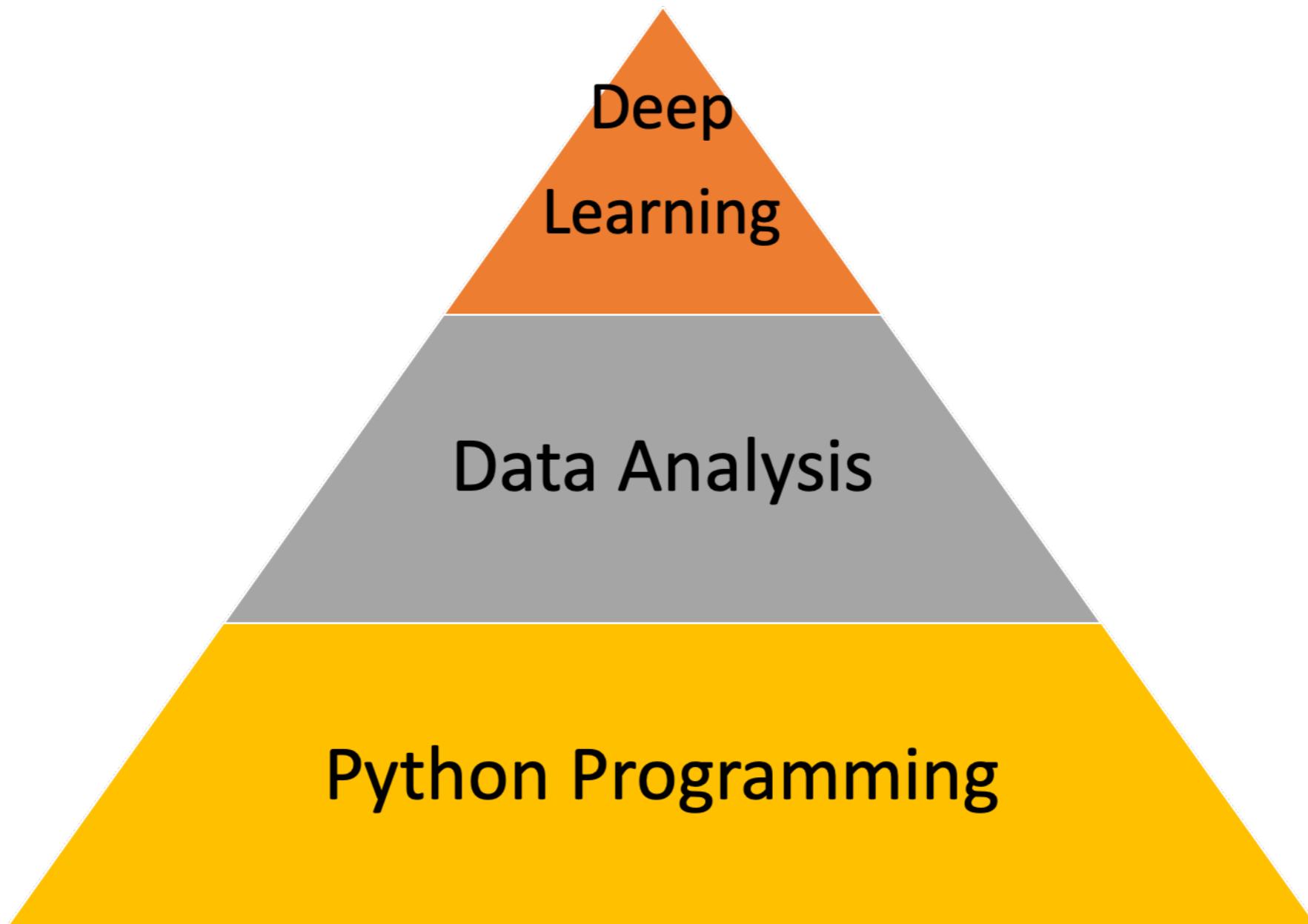


Machine Learning Workflow



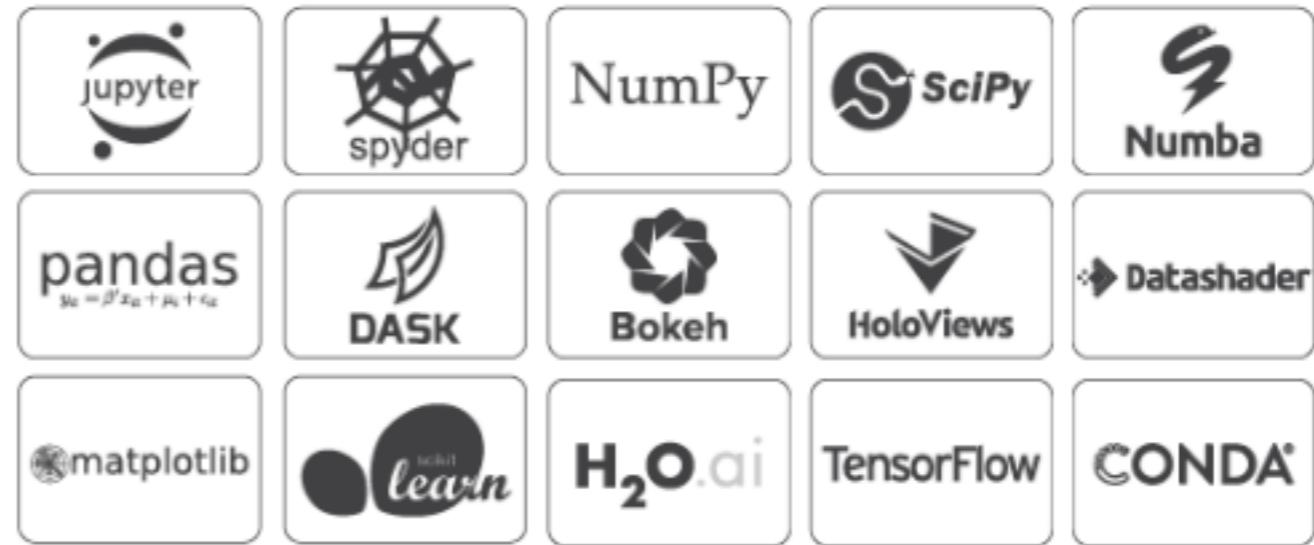
<https://www.freecodecamp.org/news/every-single-machine-learning-course-on-the-internet-ranked-by-your-reviews-3c4a7b8026c0/>

Course Content



Integrated Development Environment

- Anaconda: <https://www.anaconda.com/>



Windows | macOS | Linux

Anaconda 2019.10 for Windows Installer

Python 3.7 version

[Download](#)

64-Bit Graphical Installer (462 MB)
32-Bit Graphical Installer (410 MB)

Python 2.7 version

[Download](#)

64-Bit Graphical Installer (413 MB)
32-Bit Graphical Installer (356 MB)



Google Colaboratory

<https://colab.research.google.com/>

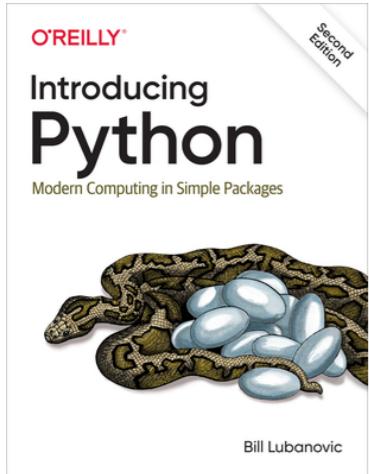
Google Colab(oratory)

- For you Jupyter Notebook fans, this is an even better option!
- 1) Hosted by Google
 - a) Extremely fast network speed
 - 2) Access to GPU and TPU (Tensor Processing Unit)
 - a) Not something you can buy for your PC!
 - b) TF code works the same on all devices
 - 3) Stored in Google Drive (the “cloud”)
 - a) You’ll never lose it, and it’s easy to share
 - 4) Many libraries for deep learning / machine learning / data science
 - a) More than I assumed there would be! (Theano, PyTorch)

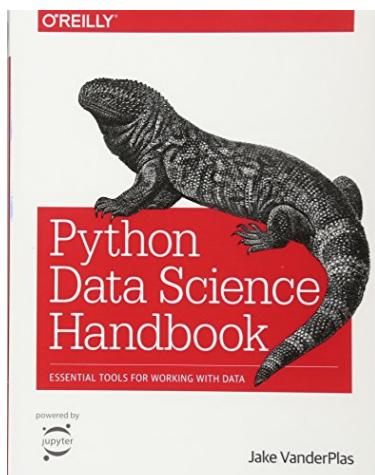
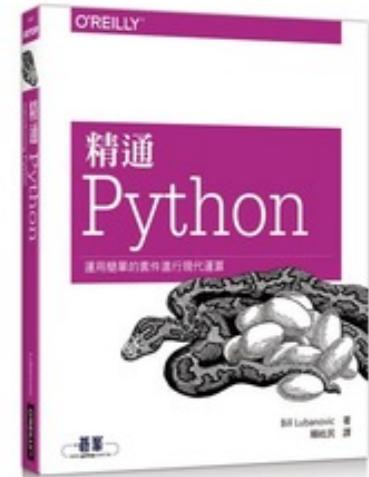


<https://www.youtube.com/watch?v=inN8seMm7UI>

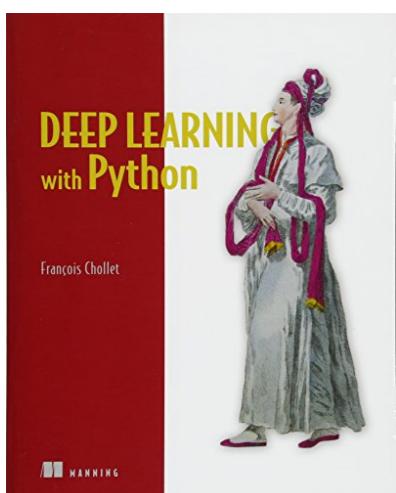
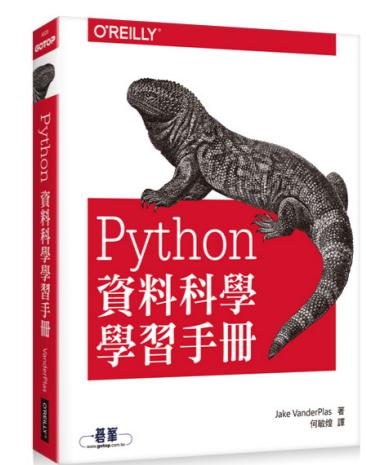
Reference Books



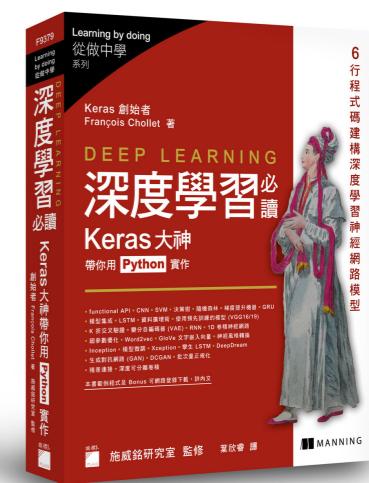
- Bill Lubanovic (2019). *Introducing Python: Modern Computing in Simple Packages*. Second Edition. O'Reilly, 500 pages



- Jake VanderPlas (2016). *Python Data Science Handbook: Essential Tools for Working with Data*. O'Reilly, 548 pages



- Francois Chollet (2017). *Deep Learning with Python*. Manning, 384 pages



Grading

- **Homework 30%**
 - About 10 homework, each 10 points
 - Hard deadline. Any delay will not be accepted.

- **Exam 40%**
 - Computer test at E1-223
 - Open Book
 - Without Internet



- **Project 30%**
 - At most three students per group
 - Total three data, each 6 points
 - Judgment by performance ranking in the leaderboard
 - Project report, occupied 12 points

Schedule (1/2)

- Week #1: 3/5 (四) Course Overview、Jupyter Notebook
- Week #2: 3/12 (四) Number, String, List, Tuple, Dictionary, Set
- Week #3: 3/19 (四) Choose with if, Loop with while and for, Function
- Week #4: 3/26 (四) Model, Package, Program, File
- Week #5: 4/2 (四) 兒童節補假
- Week #6: 4/9 (四) Numpy
- Week #7: 4/16 (四) Pandas
- Week #8: 4/23 (四) Matplotlib (期中考週)
- Week #9: 4/30 (四) Scikit-Learn

Schedule (2/2)

- Week #10: 5/7 (四) Keras, Tensorflow
- Week #11: 5/14 (四) MLP, CNN
- Week #12: 5/21 (四) RNN, LSTM
- Week #13: 5/28 (四) **Computer Test**
- Week #14: 6/4 (四) **Final Project Presentation #1**
- Week #15: 6/11 (四) **Final Project Presentation #2**
- Week #16: 6/18 (四) **Final Project Presentation #3 (期末考週)**
- Week #17: 6/25 (四) 端午節彈性放假
- Week #18: 7/2 (四) **Final Project Report Due (彈性補充教學)**

Teaching Assistants

- TA Hours (Other time slots by appointments)
 - **Tue. 10:00 - 12:00**
高浩銓 (lockingkao@g.ncu.edu.tw)
鄭少鈞 (jim860530@gmail.com)
 - **Thu. 15:00 - 17:00**
陳柏翰 (elphc34@gmail.com)
陳昌浩 (haoareyou1025@gmail.com)
- Location at **E6-B417** (工程五館4F)

Summary

- 如何學好程式設計？錯誤中學習！不間斷的練習
- 你願意花多少時間寫程式呢？一分耕耘一分收穫

