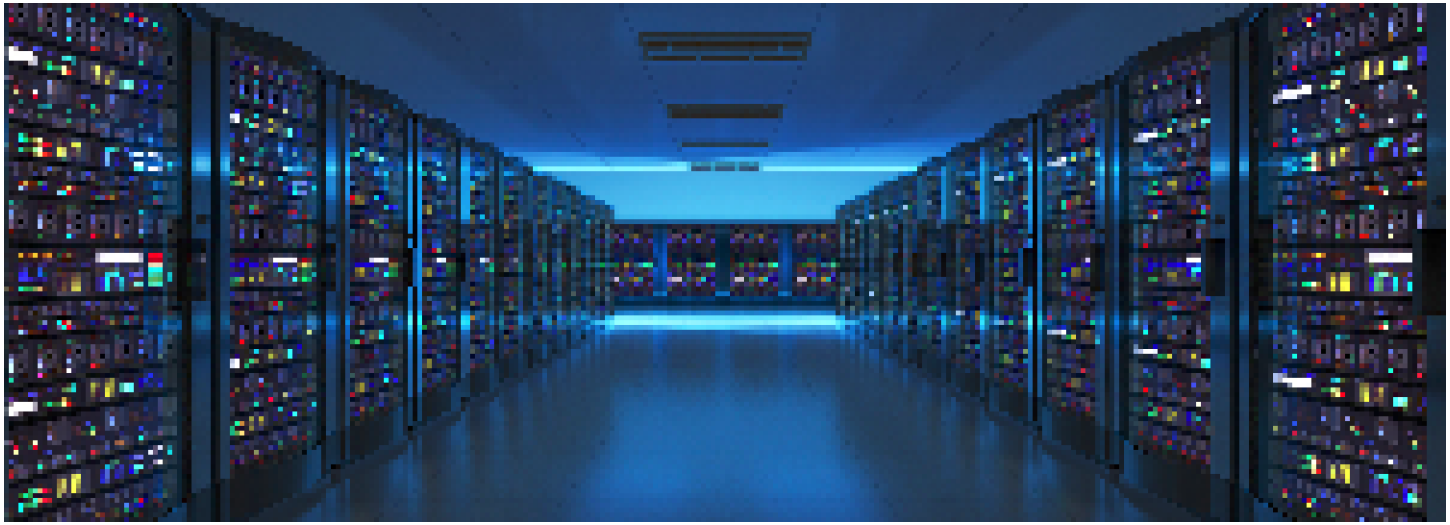
INTRODUCTION TO DATA SCIENCE



Practical Lectures

by Chris Emmery (MSc)

@_cmry • @cmry

WHAT IS DATA SCIENCE?

Data science is a "concept to unify statistics, data analysis and their related methods" in order to "understand and analyze actual phenomena" with data.

WHAT MAKES A DATA SCIENTIST?

Data scientists use their data and analytical ability to find and interpret rich data sources; manage large amounts of data (…); create visualizations to aid in understanding data; build mathematical models using the data; and present and communicate the data insights/findings.

?

RELATED FIELDS

Artificial Intelligence	Natural Language Processing	VR / Sensory
Machine Learning	Computer Vision	Medical
Data Mining	Audio Signal Processing	Intelligent Games
Information Retrieval	Cognitive Sciences	Agents (Biology)

ONE COMMONALITY: DATA-DRIVEN SCIENCE

WHAT IS DATA?

CHILD INTERPRETATION

outlook	temp.	windy	play
sunny	hot	no	no
sunny	hot	yes	no
sunny	mild	no	yes
cloudy	hot	no	yes
rainy	mild	no	yes
rainy	cold	yes	no

It's **sunny**, **mild**, and **windy**… should I play?

TO FEATURES

outlook	temperature	windy	play
1	1	0	0

or

sunny	cloudy	rainy	hot	mild	cold	windy	play
1	0	0	1	0	0	0	0

TO FEATURE VECTORS

$$\vec{v} = \langle 1, 1, 0, 0 \rangle$$

or

$$\vec{v} = \langle 1, 0, 0, 1, 0, 0, 0, 0 \rangle$$

Next lecture.

MEASUREMENTS

	deg	feel	precip.	wsw	uv	thunder
	22	25	13	13	9	0
units	°	°	%	km/h	index	%

OTHERS

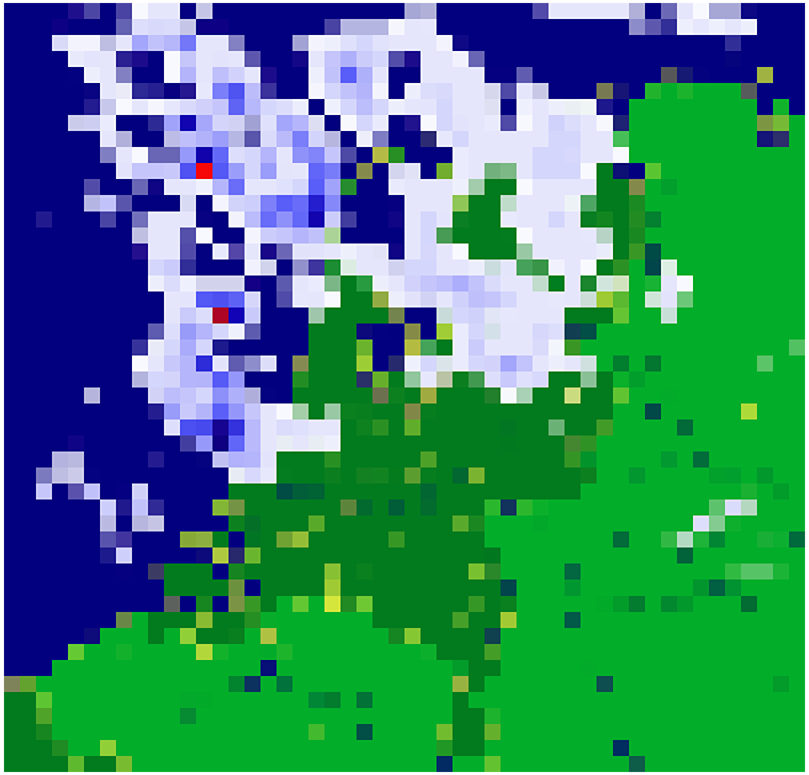


Image data / combination with other data sources.

INTERPRETING DATA

BACK TO OUR DATA

outlook	temp.	windy	play
sunny	hot	no	no
sunny	hot	yes	no
sunny	mild	no	yes
cloudy	hot	no	yes
rainy	mild	no	yes
rainy	cold	yes	no

Can think of rules it's play time?

RULES FOR PREDICTION

We want to predict our **target** `play` given the **features** we have available.

if it's windy → no play

if it's hot and no wind → no play

if it's not windy and not hot → play

FORMALLY

- We have our data: X (with features: outlook, temp., windy).
- Our data exists of smaller instances, 'some instance' is written as: x.
- If we want to specifically point at a particular instance (say our first row), we write: x_1. We can see our model as a function f, that when given any instance x, gives us a prediction $\hat{y}$.
- The application of the model to some instance in our data can be written as $f(x)$.
- Our hope is that $\hat{y}$ is the same as our target: y.

RECAP

- Features: X (outlook, temp., windy)
- Targets: Y (play)
- Some instance: x
- Some target: y
- First column: x_1 (sunny, hot, no)
- First target: y_1 (no)
- Model: if it's not windy and not hot → play (f)
- Predictions by $f(x)$: $\hat{y}$
- Prediction for $f(x_1)$: $\hat{y}_1$ (no)

PREDICTIVE MODEL (OR ALGORITHM)

what makes an algorithm?

```
def play_predictor(data):
    if data['windy'] == 'no' and data['temp'] != 'hot':
        return 'play'
    else:
        return 'no play'
```

It's **sunny**, **mild**, and **windy**... should I play?

Realistic?

HOW DO WE KNOW IF OUR MODEL PERFORMS WELL?

- **Correct** evaluation is incredibly important in Data Mining.
- We came up with some rules, but how do we know they generalize; if the rules we learned apply with the same success rate to data where we **don't** know what the **target** is.

LET'S EVALUATE OUR CURRENT MODEL

if it's not windy and not hot → play

outlook	temp.	windy	play
sunny	hot	no	no
sunny	hot	yes	no
sunny	mild	no	yes
cloudy	hot	no	yes
rainy	mild	no	yes
rainy	cold	yes	no

RESULTS

- We got 5/6 correct! 😊
    - The model has 83.3% **accuracy**.
- Did we cover all conditions?
- What if we are presented with new conditions?
- Rules are probably too strict.
- Other than the **training** data we determined our rules by, we also need **test** data; unseen by us, to evaluate.

TEST

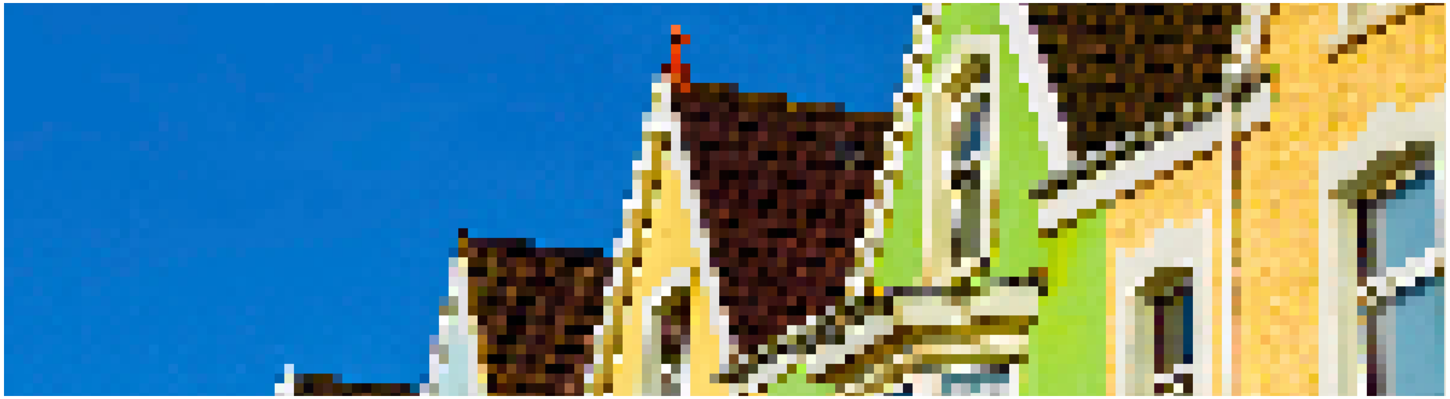
if it's not windy and not hot → play

outlook	temp.	windy	play
cloudy	hot	yes	?
rainy	mild	no	?

Actual labels turned out to be: 1 - yes, 2 - no.

Accuracy: 0% - but should we update our rules?

REALISTIC USE CASE



PREDICTING HOUSING PRICES

- Would you be able to determine the price of a house? → Expert knowledge.
- **Many observations** required to gain experience.
- Can you come up with a few features to predict the price of a house?

HOW TO EVALUATE?

- Previously we had a clear **binary** prediction. Either yes, or no.
- Say we had more classes, we would still be predicting a **nominal target** (order does not matter).
- What about a **numeric target** like housing prices?
- We can't say: we got ... out of ... correct, and therefore use **accuracy**.
- We are more likely interested in how far our prediction was off from the actual value: this is <text **error**.

TYPES OF PREDICTION

- classes → **classification**
- values → **regression**

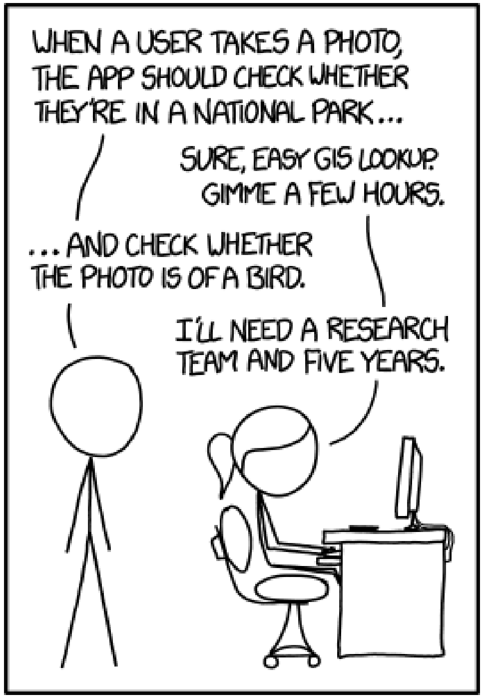
COMPLEX INFORMATION

- How would location affect price?
- How would pollution affect price?
- How about good location but high pollution?
- Do you know how much of either would affect the price?
- Would one be able to easily craft a successful ruleset?

LEARNING TO PREDICT

- Hand-made rules are not flexible.
- Given more instances / **observations**, rules will become more complex, thus requiring better (more complex) rules.
- Too much data becomes impossible to manually analyse.
- If done automatically, little expert knowledge is required; **mostly** data.
- Models can give information regarding underlying patterns and feature importances.
    - If many rules mention location as a first condition to look at, that must be an important feature.

MACHINE LEARNING (PAST)



IN CS, IT CAN BE HARD TO EXPLAIN THE DIFFERENCE BETWEEN THE EASY AND THE VIRTUALLY IMPOSSIBLE.

MACHINE LEARNING (NOW V1)

DeepMind's new AI system can learn based on its past experiences ...

This New Atari-Playing AI Wants to Dethrone DeepMind | WIRED

Google's DeepMind makes AI program that can learn like a human ...

AI is one step closer to mastering StarCraft - The Verge

Google's DeepMind A.I. has learned to play a game called ant soccer ...

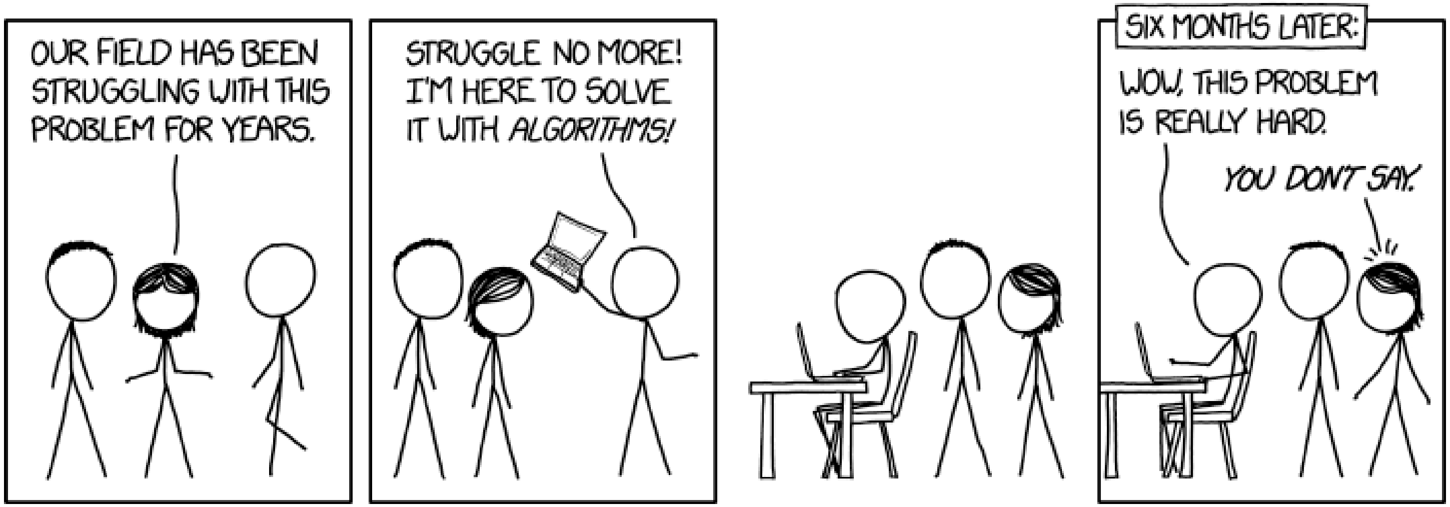
DeepMind is using games to test AI aggression and cooperation

Google's DeepMind Taught Its A.I. How To Categorize Objects After ...

IS THAT ALL?

- Intuitions.
- Domain expertise.
- Get to know your data.

MACHINE LEARNING (NOW V2)



EXTRA MATERIAL

Quick discussion of:

- PC hardware and relation to data and algorithms.
- Programming languages and their relation to above.

> *This overview is **very** limited, but all you need to know.*

THIS IS NOT COMPUTER SCIENCE, WHY DO I NEED TO KNOW THIS?

- Algorithm choices often depend on hardware limitations.
- Some model families specifically deal with shortage of computation power.
- Different data types often relate to storage and processing.
- Certain terms are widespread throughout this course.

PC HARDWARE

HARD DRIVE I

HARD DRIVE II

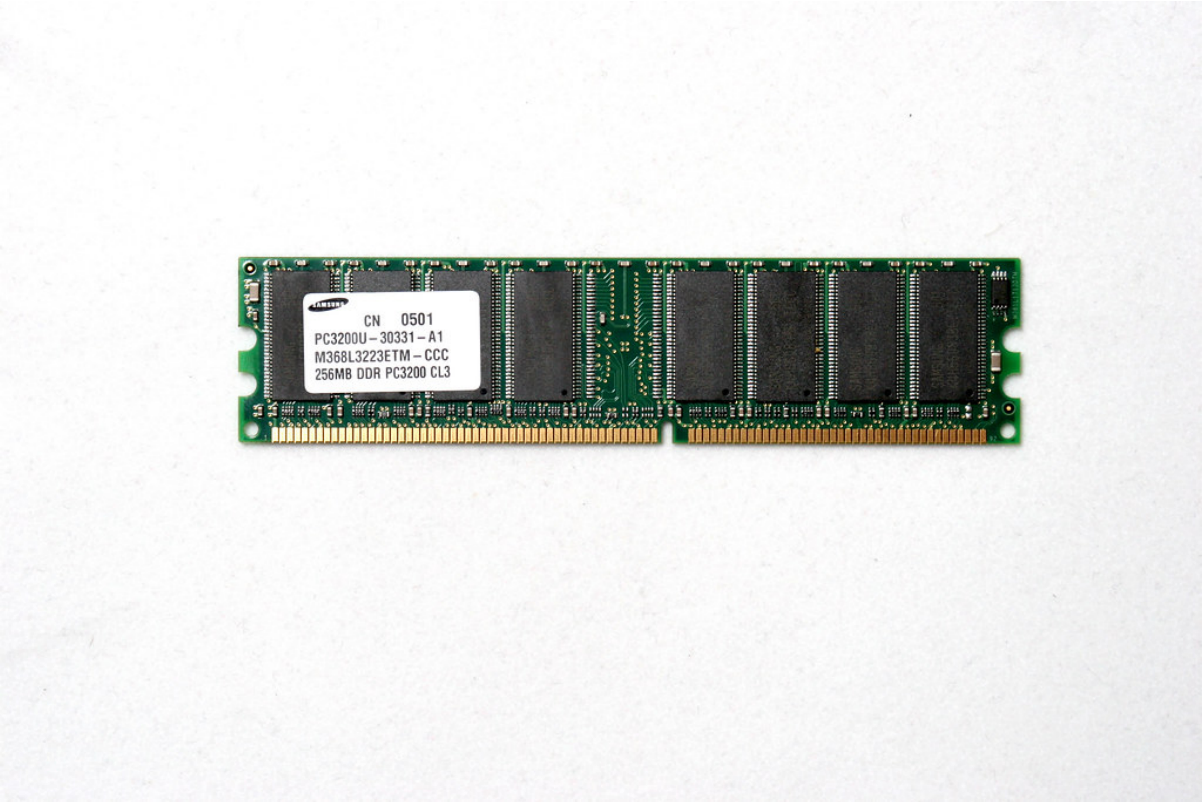
DRIVES

hdd-ssd

DRIVES (HDD / SSD)

- Stores your files.
- HDD are larger (store more data, 1-5T) but slower (in reading / writing), and fragile.
- SSDs are smaller (up to 1T), faster, more robust, but expensive.
- Most modern laptops come with an SSD.
- For computation, algorithms / models read a particular set of data from your disks into **memory**.

MEMORY

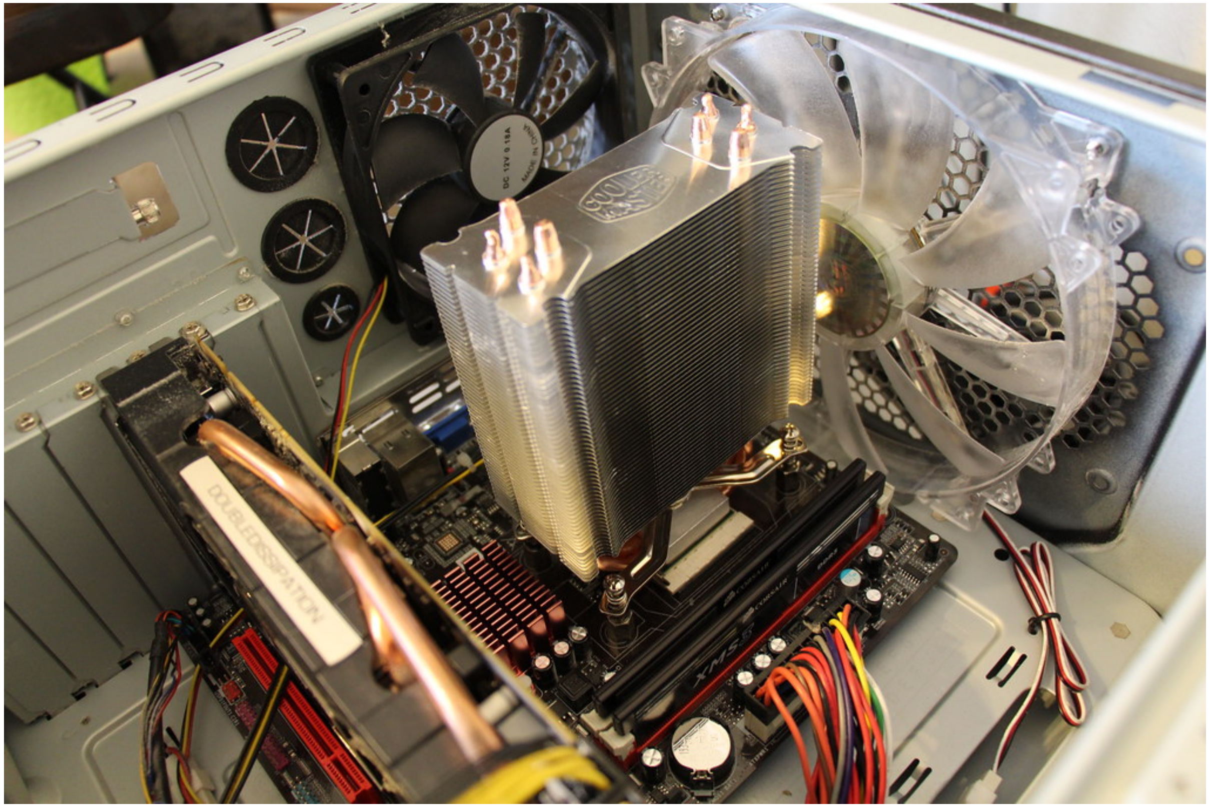

CN 0501
PC3200U-30331-A1
M368L3223ETM-CCC
256MB DDR PC3200 CL3

MEMORY (RAM)

- Very fast reading / writing, but even more limited in space (commonly 8-16G, up to 256G), very expensive.
- Algorithms can quickly access and manipulate data that is in memory.
- If memory limit is exceeded, computers usually freeze / processes slow down.
- Computations done on data in memory are commonly handled by the **CPU**.

CPU I

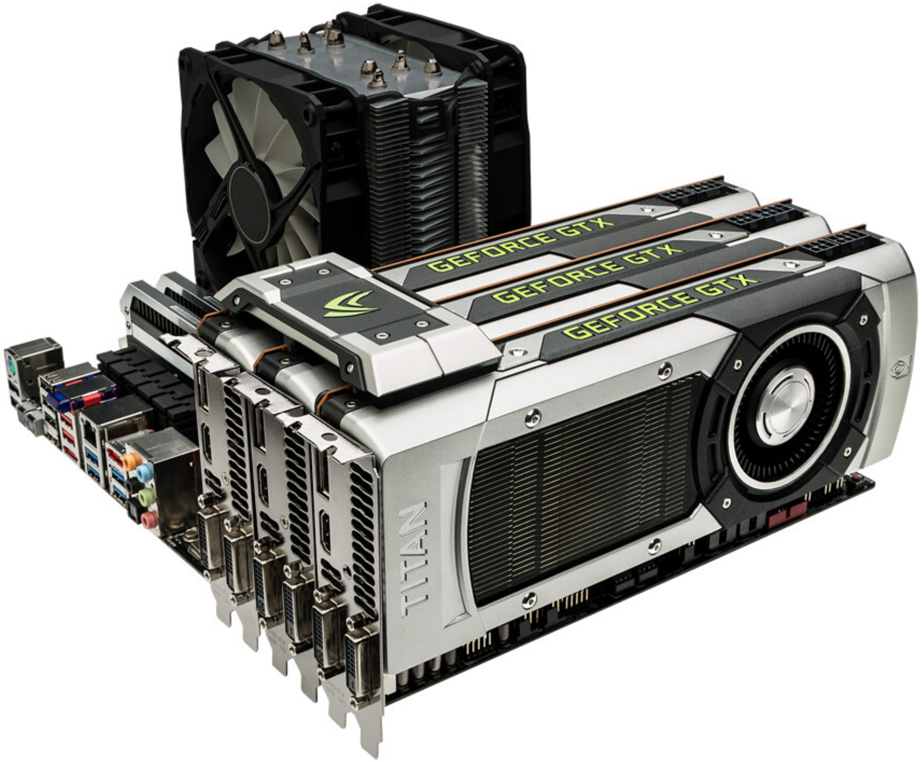
CPU II



PROCESSOR (CPU)

- Does computation part of a computer.
- Can have multiple computation cores (duo core, quad core, etc.) to run operations in parallel (i.e. simultaneously), which speeds up processes.
- The more expensive the CPU, the faster it does similar computations. The more cores, the faster it runs parallel computations.

GPU



GRAPHICS CARD (GPU)

- Some computations can be done on a GPU rather than the CPU.
- Commonly used for processing images or other visual content. Popular for video games.
- For ordinary systems, GPU is usually embedded in the CPU.
- GPU's are very fast at 'matrix operations', and have therefore been popularized for Deep Learning research (explained in future lectures).
- Has its own RAM (and therefore limitations).

PROGRAMMING LANGUAGES