Machine Learning & Ethics

William Seymour | Human Centred Computing
william.seymour@cs.ox.ac.uk

Hello!

- 3rd year PhD student in Human Centred Computing
- Help teach Imperative Programming and Computer Security modules
- Sideline in competitive security (hacking/lockpicking)
- Everything you see today will be online at https://github.com/mcnutty26/uniq

UNIVERSITY OF OXFORD

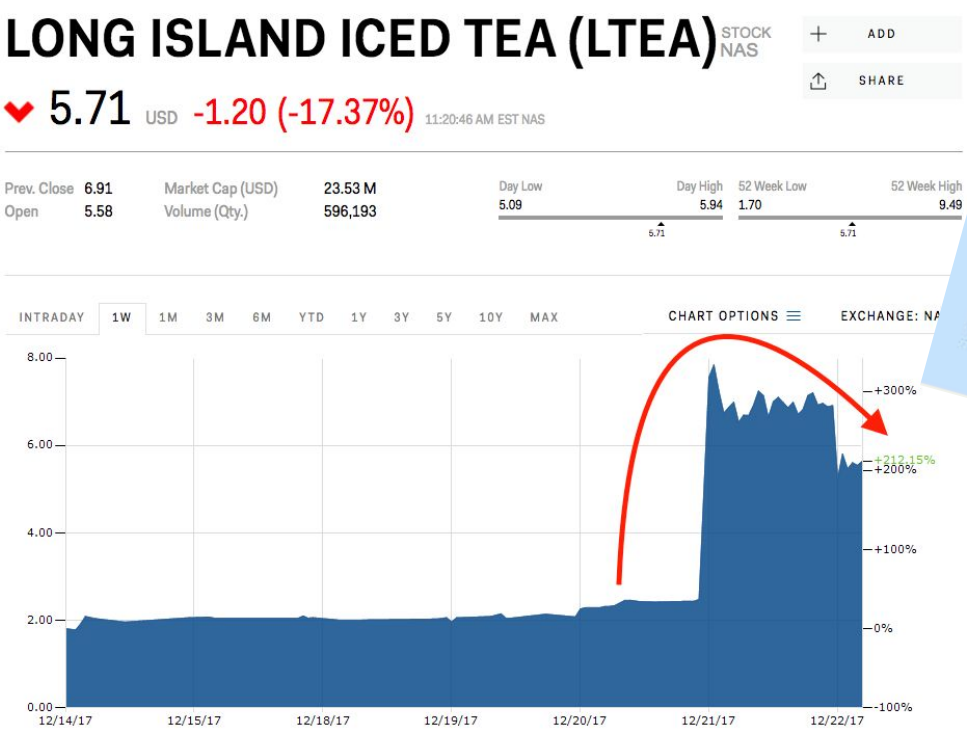
DEPARTMENT OF COMPUTER SCIENCE

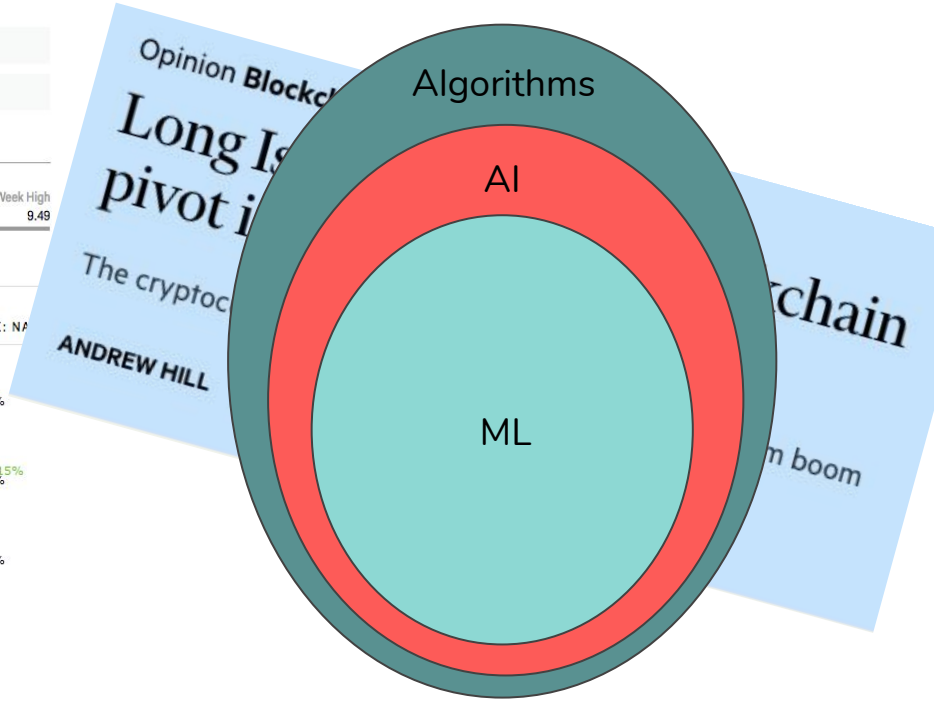
CENTRE FOR DOCTORAL TRAINING *in* CYBER SECURITY

Overview

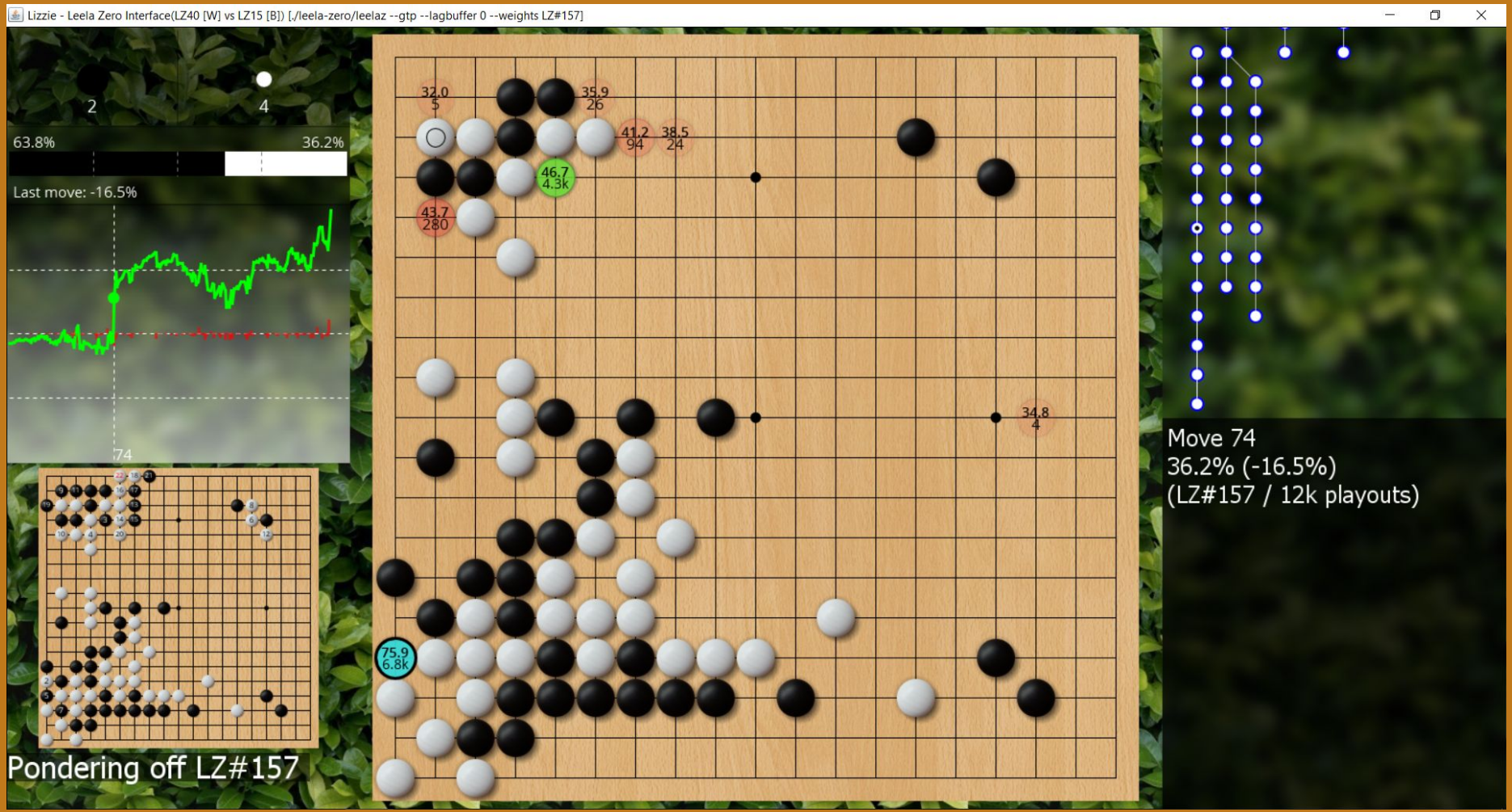
- Introduction to Machine Learning (13:30-14:10, Access Grid)
  - ML vs AI vs algorithms
  - Regression
  - Decision trees
  - Random forests
  - Neural networks
- Practical: Building a Classifier (14:20-15:00, Comlab)
  - Primer on R and RStudio
  - Training and testing classifiers
  - Leaderboard competition
- Ethics in Machine Learning (15:10-15:50, Access Grid)
- Practical : Operationalising Ethics (16:00-16:30, Comlab)

In other words, plan your startup now...





What even is “machine learning”?


Lizzie - Leela Zero Interface(LZ40 [W] vs LZ15 [B]) [./leela-zero/leelaz --gtp --lagbuffer 0 --weights LZ#157]
2
4
63.8%
36.2%
Last move: -16.5%
74
32.0
5
35.9
26
41.2
94
38.5
24
46.7
4.3k
43.7
280
34.8
4
75.9
6.8k
Move 74
36.2% (-16.5%)
(LZ#157 / 12k playouts)
Pondering off LZ#157

“Using statistical methods to have programs improve over time”

supervised

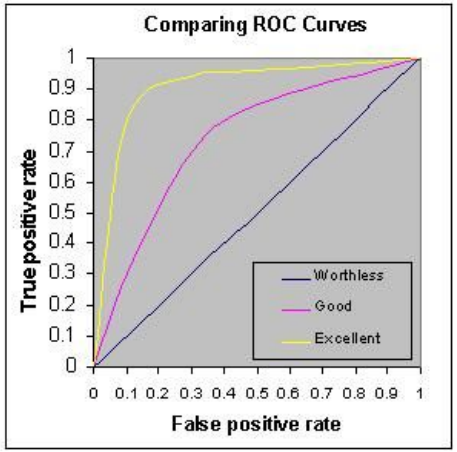
unsupervised

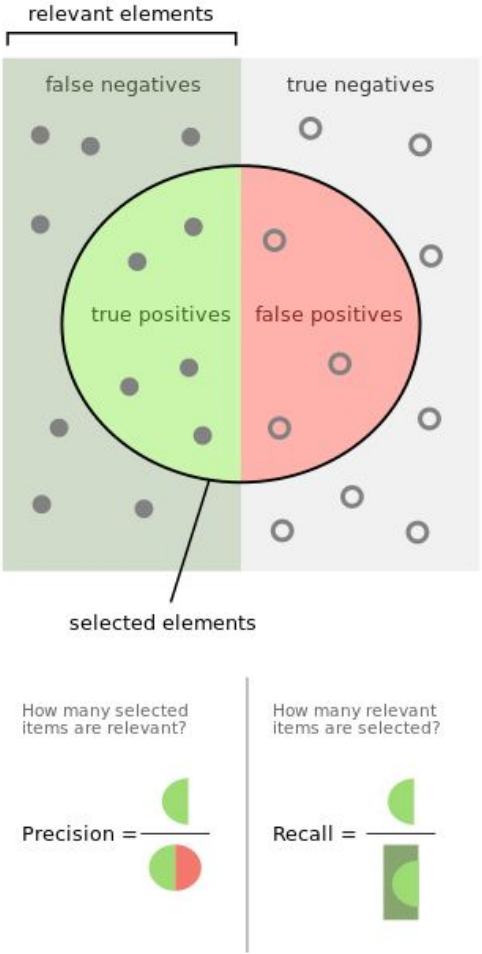
Accuracy is just a percentage, right?

Accuracy and Friends

- Precision is the probability that a (random) result predicted positive is actually positive.
- Recall is the probability that a (random) positive result is predicted positive.

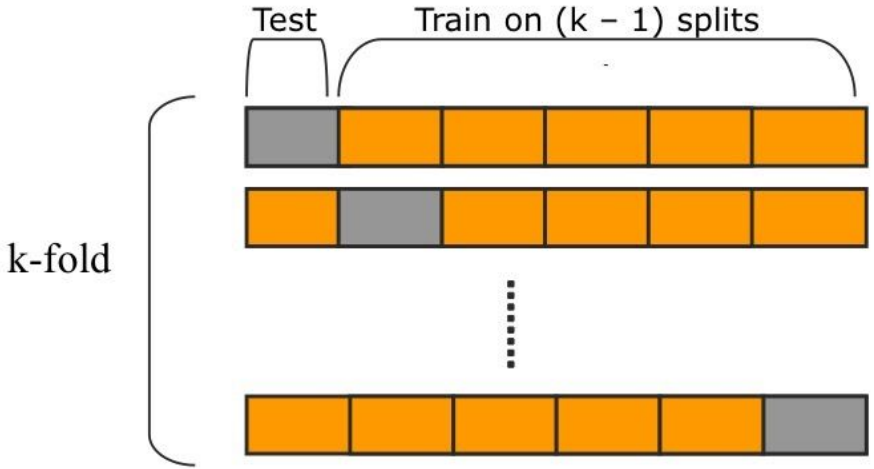
- Other options like ROC curves give us different information about a classifier



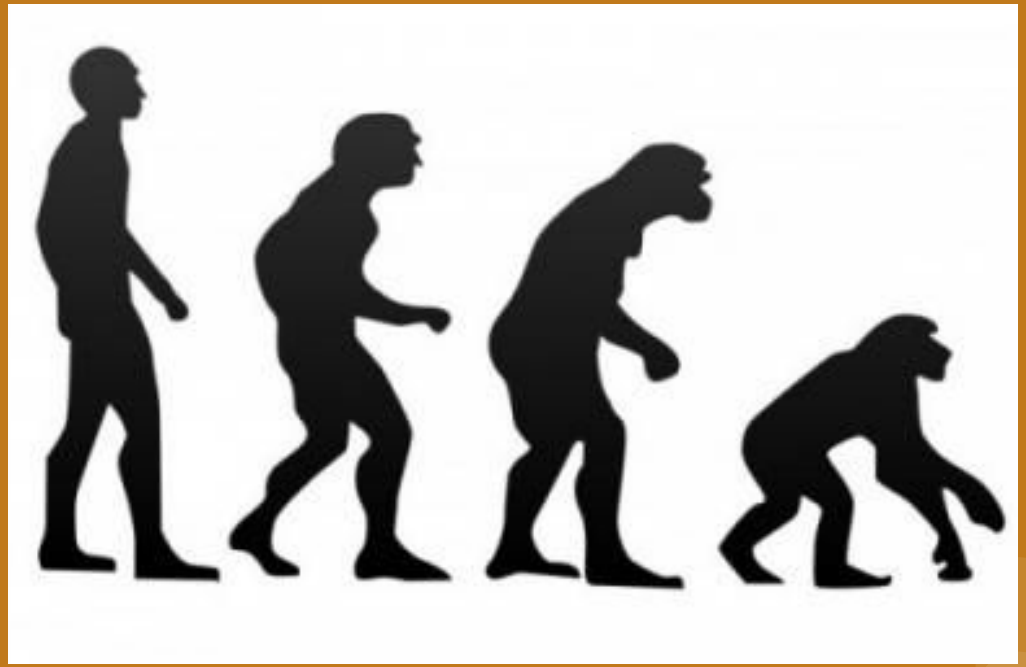


Training and Testing

- You calculate these metrics using a *training* data set
- Using the same data to test and train leads to “overfitting”
- Various (better) options like train/test split and k-fold validation



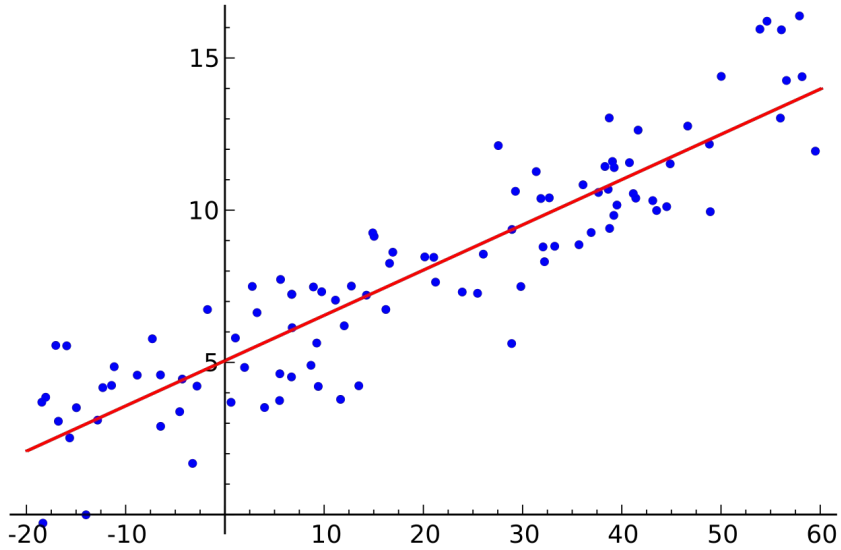
Regression



Linear Regression

- Predict the values of numeric data
- Linear regression same as in S1
- $y = \alpha + \beta x + \gamma z + \ldots$
- Trying to minimise sum of squared errors when fitting the line
- For a single independent variable:

$$\frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$



Linear Regression Example

$$\frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

- Average(x) =
- Average(y) =
- Sum of x errors =
- Sum of y errors =
- Sq. sum of x errors =
- Coefficient =
- Intercept =

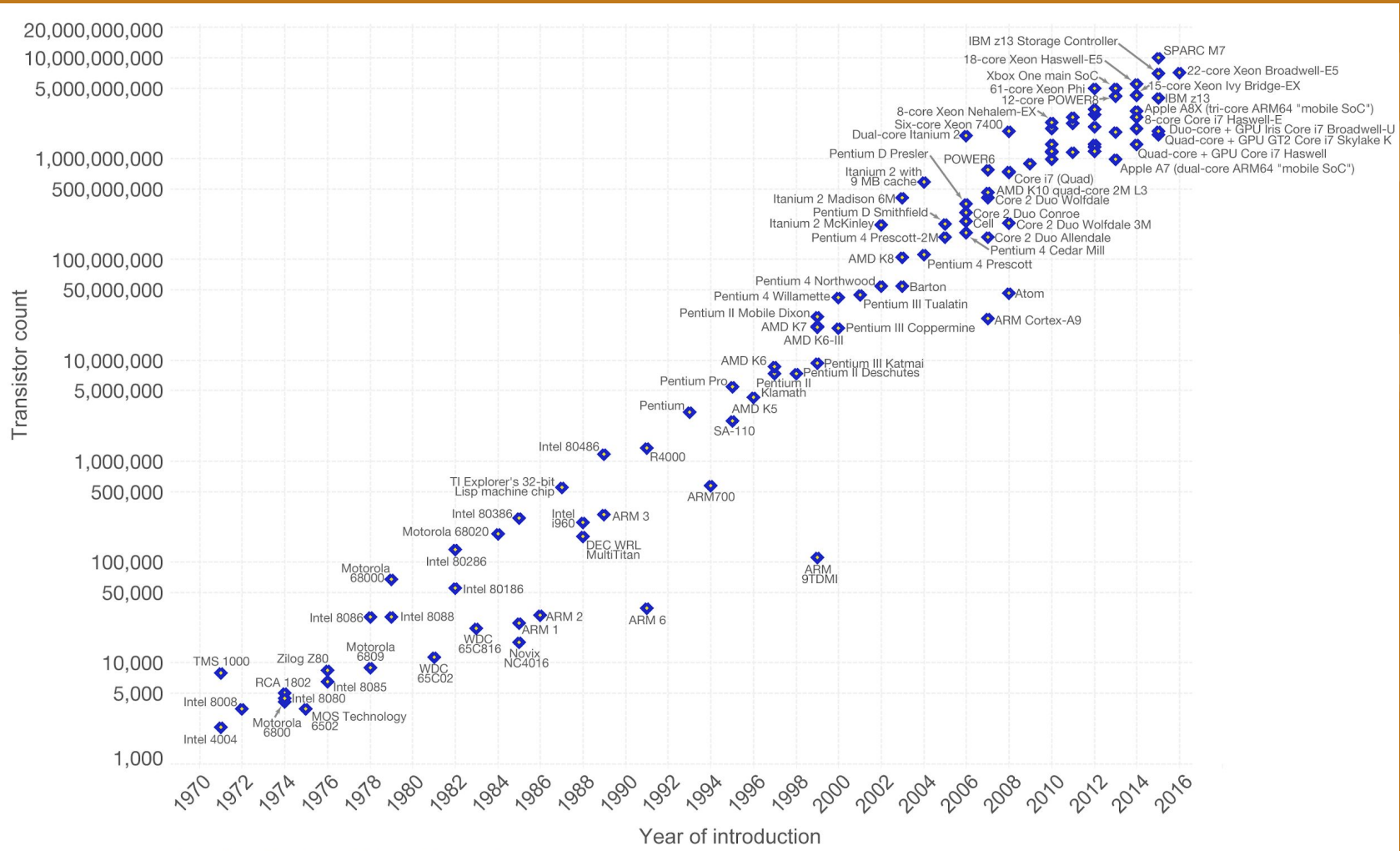
Password	Numbers	Crack Time (s)	$(x - \bar{x})$	$(y - \bar{y})$
v3BYUAHbJA	1	1.37		
PMAXimkXNs	0	0.52		
nX37FkRcLL	2	2.10		
wQ5TpMXAOT	1	1.92		
bS9Qqtjyb0	2	2.84		
sajWxrZqd9	1	1.65		
dcLhGSrGJG	0	0.45		
fajY1MLD00	3	2.99		
OQhMcTl8IA	1	1.87		
x4cu3tORHH	2	2.43		

Linear Regression Example

$$\frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

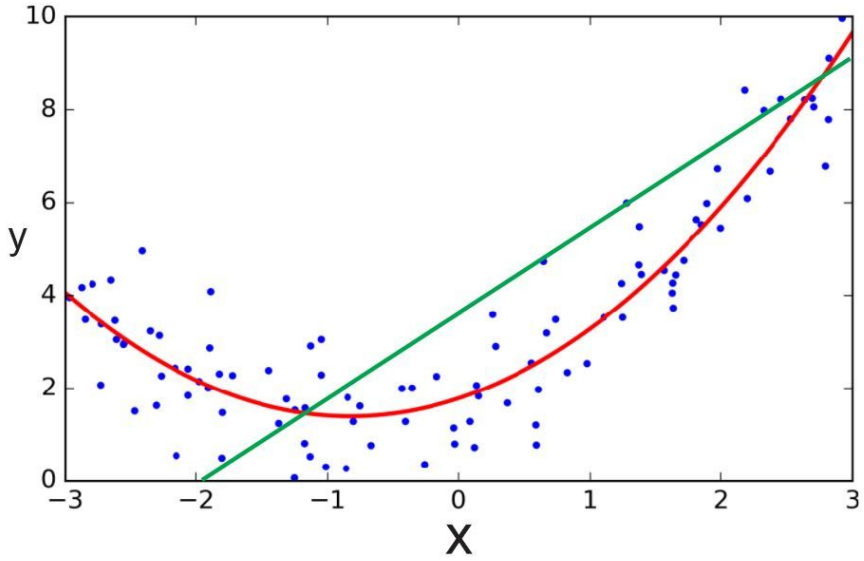
- Average(x) = 1.3
- Average(y) = 1.8
- Sum of x errors = 1
- Sum of y errors = 0.14
- Sq. sum of x errors = 1
- Coefficient = 0.14
- Intercept = 1.66

Password	Numbers	Crack Time (s)	$(x - \bar{x})$	$(y - \bar{y})$
v3BYUAHbJA	1	1.37	-0.3	-0.43
PMAXimkXNs	0	0.52	-1.3	-1.28
nX37FkRcLL	2	2.10	0.7	0.3
wQ5TpMXAOT	1	1.92	-0.3	0.12
bS9Qqtjyb0	2	2.84	0.7	1.04
sajWxrZqd9	1	1.65	-0.3	-0.15
dcLhGSrGJG	0	0.45	-0.3	-1.35
fajY1MLD00	3	2.99	1.7	1.19
OQhMcTl8IA	1	1.87	-0.3	0.07
x4cu3tORHH	2	2.43	0.7	0.63


Transistor count
20,000,000,000
10,000,000,000
5,000,000,000
1,000,000,000
500,000,000
100,000,000
50,000,000
10,000,000
5,000,000
1,000,000
500,000
100,000
50,000
10,000
5,000
1,000
Year of introduction
1970
1972
1974
1976
1978
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
2004
2006
2008
2010
2012
2014
2016
IBM z13 Storage Controller
SPARC M7
18-core Xeon Haswell-E5
22-core Xeon Broadwell-E5
Xbox One main SoC
15-core Xeon Ivy Bridge-EX
61-core Xeon Phi
12-core POWER8
IBM z13
8-core Xeon Nehalem-EX
Apple A8X (tri-core ARM64 "mobile SoC")
Six-core Xeon 7400
8-core Core i7 Haswell-E
Dual-core Itanium 2
Duo-core + GPU Iris Core i7 Broadwell-U
Quad-core + GPU GT2 Core i7 Skylake K
Pentium D Presler
POWER6
Quad-core + GPU Core i7 Haswell
Itanium 2 with 9 MB cache
Core i7 (Quad)
Apple A7 (dual-core ARM64 "mobile SoC")
AMD K10 quad-core 2M L3
Itanium 2 Madison 6M
Core 2 Duo Wolfdale
Pentium D Smithfield
Core 2 Duo Conroe
Itanium 2 McKinley
Cell
Core 2 Duo Wolfdale 3M
Pentium 4 Prescott-2M
Core 2 Duo Allendale
Pentium 4 Cedar Mill
AMD K8
Pentium 4 Prescott
Pentium 4 Northwood
Barton
Atom
Pentium 4 Willamette
Pentium III Tualatin
Pentium II Mobile Dixon
ARM Cortex-A9
AMD K7
Pentium III Coppermine
AMD K6-III
AMD K6
Pentium III Katmai
Pentium II Deschutes
Pentium Pro
Pentium II Klamath
Pentium
AMD K5
SA-110
Intel 80486
R4000
TI Explorer's 32-bit Lisp machine chip
ARM700
Intel 80386
Intel i960
ARM 3
Motorola 68020
DEC WRL MultiTitan
Intel 80286
ARM 9TDMI
Motorola 68000
Intel 80186
Intel 8086
Intel 8088
ARM 2
ARM 6
ARM 1
WDC 65C816
Motorola 6809
Novix NC4016
TMS 1000
Zilog Z80
WDC 65C02
RCA 1802
Intel 8085
Intel 8080
Intel 8008
MOS Technology 6502
Motorola 6800
Intel 4004

Polynomial and Logistic Regression

- Commonly polynomial & logistic
- Apply principles of linear regression to higher order relationships
- Logistic regression deals with categoric (non-numeric) dependent variables




J
Sponge


K
Mammal


L
Arthropod

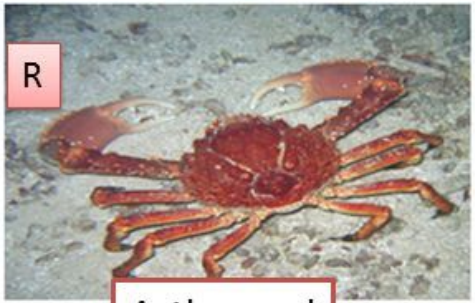

M
Fish


N
Reptile


O
Bird


P
Mollusks


Q
Amphibian

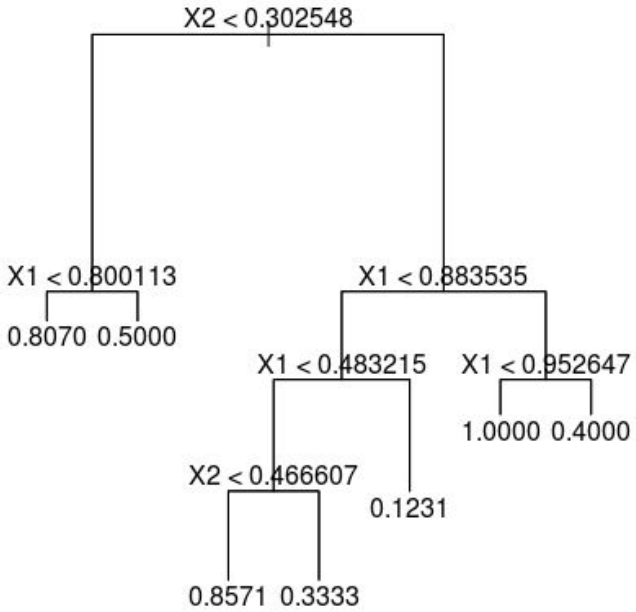

R
Arthropod

Trees and Forests

Decision Trees

- Match categorical data to classes
- Create an n-ary tree
- Trying to maximise the amount by which we partition the data
- For a binary variable:

$$1 - \sum_{t=0}^{t=1} P_t^2$$



CART for Decision Trees

Split on writing hand?

- $p(A \mid L)^2 = (1/3)^2 = 0.11$
- $P(\neg A \mid L)^2 = (2/3)^2 = 0.44$
- $p(A \mid R)^2 = (3/7)^2 = 0.18$
- $p(\neg A \mid R)^2 = (4/7)^2 = 0.33$

$$\text{Total L} = 1 - 0.44 - 0.11 = 0.44$$

$$\text{Total R} = 1 - 0.18 - 0.33 = 0.49$$

$$\text{Gini Index} = (3(0.45)+7(0.49))/10$$

$$= 0.48$$

Writing Hand	Multilingual	Ambidextrous
Right	No	No
Left	No	No
Right	Yes	Yes
Right	No	Yes
Right	Yes	Yes
Left	No	No
Left	Yes	Yes
Right	No	No
Right	Yes	No
Right	No	No

CART for Decision Trees

Split on multilingual?

- $p(A \mid Y)^2 = (3/4)^2 = 0.56$
- $P(\neg A \mid Y)^2 = (1/4)^2 = 0.06$
- $p(A \mid N)^2 = (1/6)^2 = 0.03$
- $p(\neg A \mid N)^2 = (5/6)^2 = 0.69$

$$\text{Total L} = 1 - 0.56 - 0.06 = 0.38$$

$$\text{Total R} = 1 - 0.03 - 0.69 = 0.28$$

$$\text{Gini Index} = (4(0.38)+6(0.28))/10$$

$$= 0.32$$

Writing Hand	Multilingual	Ambidextrous
Right	No	No
Left	No	No
Right	Yes	Yes
Right	No	Yes
Right	Yes	Yes
Left	No	No
Left	Yes	Yes
Right	No	No
Right	Yes	No
Right	No	No

CART for Decision Trees

Gini index for writing hand was **0.48**

Gini index for multilingual was **0.32**

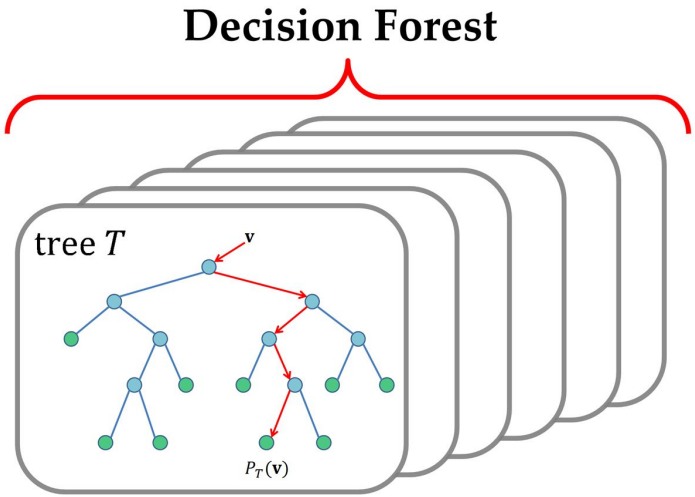
Therefore we should split by writing hand

Note: the highest possible Gini index for binary variables is

$$1 - (0.5)^2 - (0.5)^2 = 0.5$$

Ok, having one tree is cute.
What if we had a whole *forest* of trees?

Random Forests

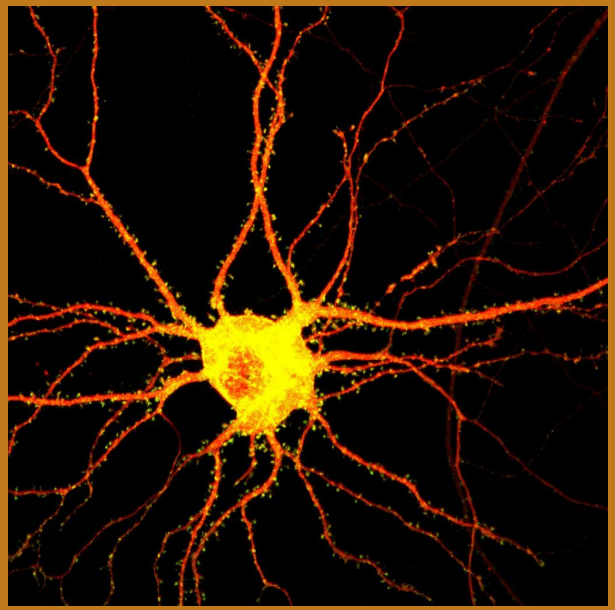


- Ensemble approach to machine learning
- Generate n trees and output the modal class
- Normally uses a technique called ‘bagging’
- Generate new training sets by sampling with replacement
- Helps with the overfitting problem

But, wait…

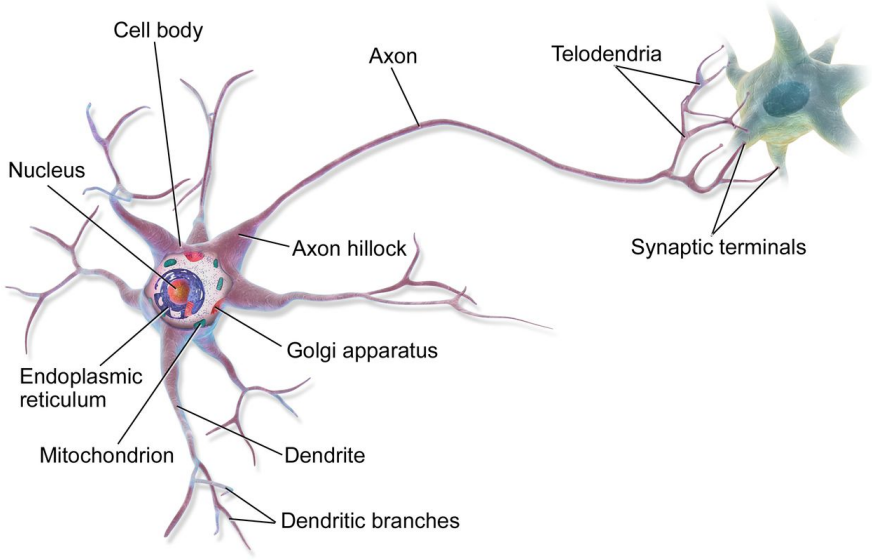
None of this is
DEEP LEARNING

Neural Networks

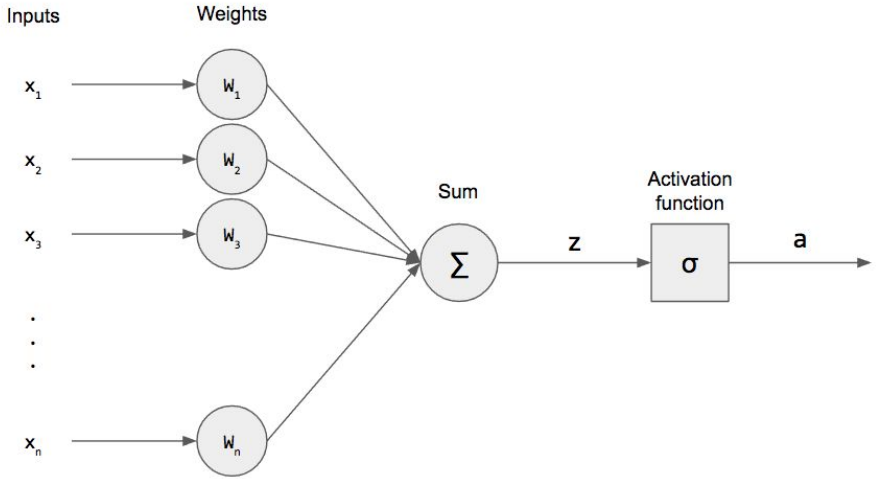


Neuroscience 101

- Basic unit of the brain is the neuron
- ~ 86bn neurons in the brain
- Modern i7 core has 2bn transistors
- About 100tn synapses
- (UK National Debt is ~£1.8tn)
- Has inputs and outputs
- “All or nothing” activation



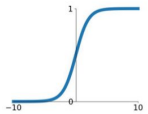
Enter the Perceptron



- Each input is weighted
- Output can vary (typically -1 to 1)
- Common activation functions include:

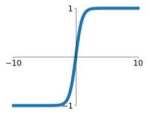
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



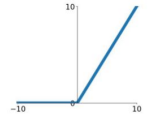
tanh

$\tanh(x)$



ReLU

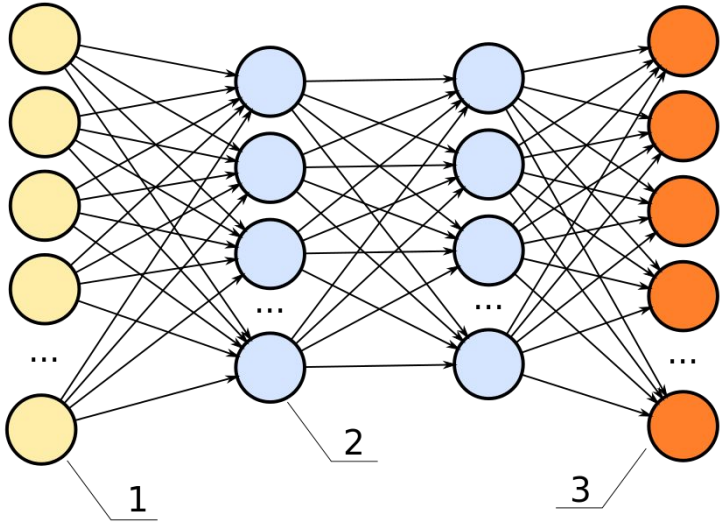
$\max(0, x)$



Ok, having one perceptron is cute.
What if we had a whole *brain* of perceptrons?

Bigger = better?

1. Input layer
2. Hidden layer(s)
3. Output layer



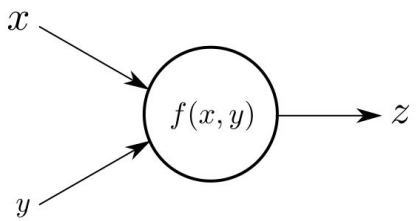
How do we figure out the weights?

Backpropagation

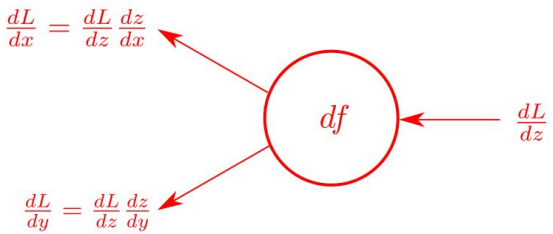
$$Error = \sum(target - output)^2$$

- Calculate the error from the output
- Gradient descent on weights
- How does each weight affect the output?
- Differentiate the output wrt. weights

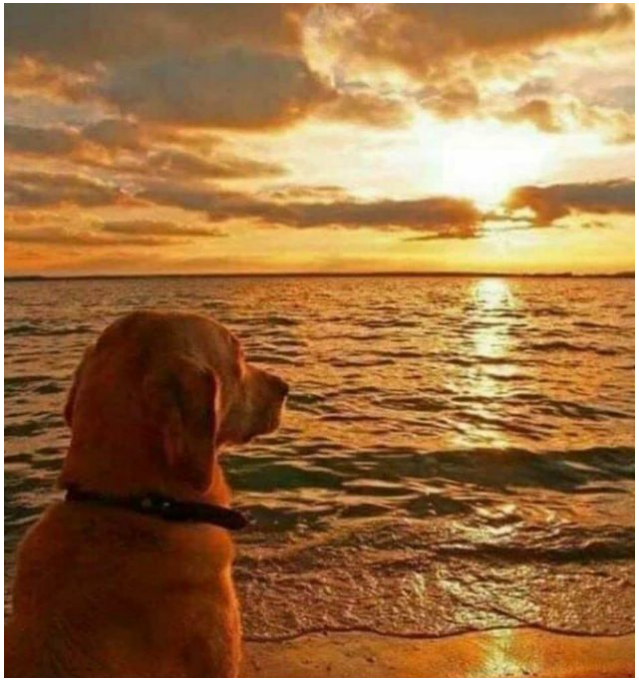
Forwardpass



Backwardpass



“Deep” learning



- TLDR; deep = more layers
- NNs have a depth (CAP)
- CAP = 2 theoretically universal
- CAP > 10 “very deep” learning
- (╯°□°)╯ ⌒ ┻━┻

Lab session: Introduction to R

A disclaimer (and apology) about using R...

Loading Packages and Importing Data

library(“caret”)

- Load the caret package

ds = read.csv(“filename.csv”)

- Load the data set in the file filename.csv

Creating Training and Test Data Sets

```
index = createDataPartition(ds$feature, p=0.75, list=FALSE)

training = ds[index,]

test = ds[-index,]
```

- Split the dataset ds into new datasets containing 75% and 25% of the data
- Make sure that each set has roughly equal proportions of possible ds$feature values

Creating and Using a Model

model = train(training[,columns], training$target), method = 'type')

predictions = predict(object=model, test[,columns])

- Train a model of a certain type on the columns and target feature of the training data set

Challenge (for after the break)

Build a Predictive Policing Classifier

Real data from ~7,000 arrests

Predict who will reoffend within 2 years