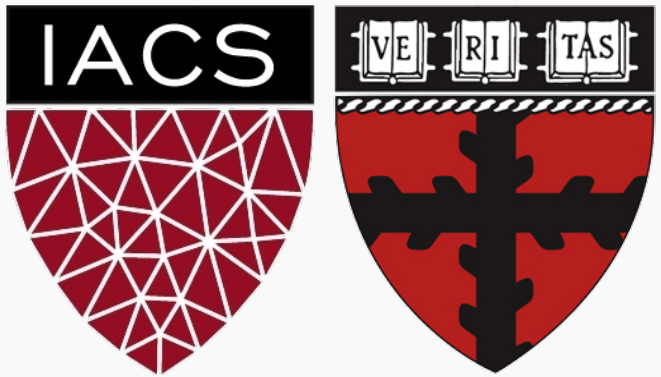
Perceptron and Multilayer Perceptron

CS109A Introduction to Data Science

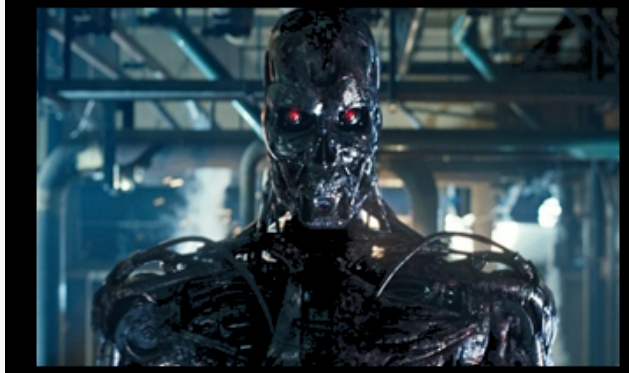
Pavlos Protopapas, Kevin Rader and Chris Tanner



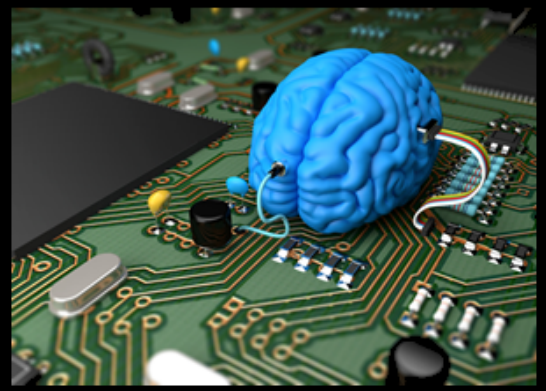
Outline

1. Introduction to Artificial Neural Networks
2. Review of basic concepts
3. Single Neuron Network ('Perceptron')
4. Multi-Layer Perceptron (MLP)

Deep Learning

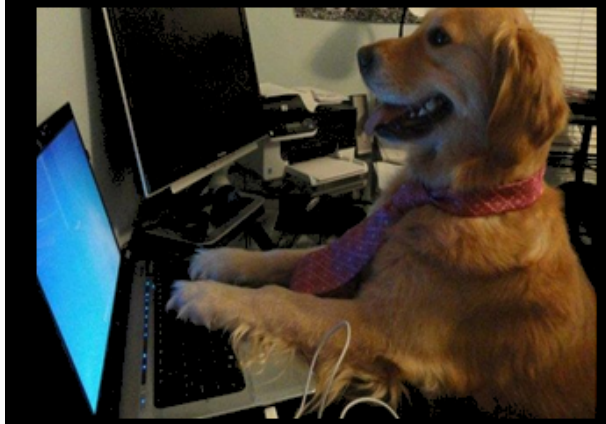


What society thinks I do



What my friends think I do

What other computer scientists think I do



What mathematicians think I do

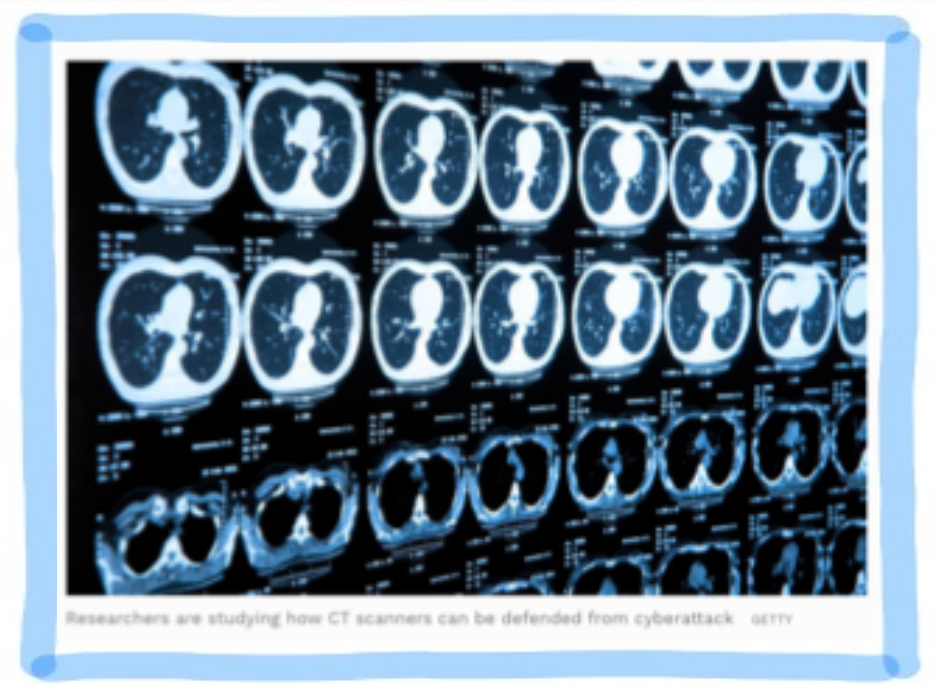
What I think I do

```
In [1]:
import keras
Using TensorFlow backend.
```

What I actually do

Today's news

Stopping Cyberattacks



Researchers are studying how CT scanners can be defended from cyberattack GETTY

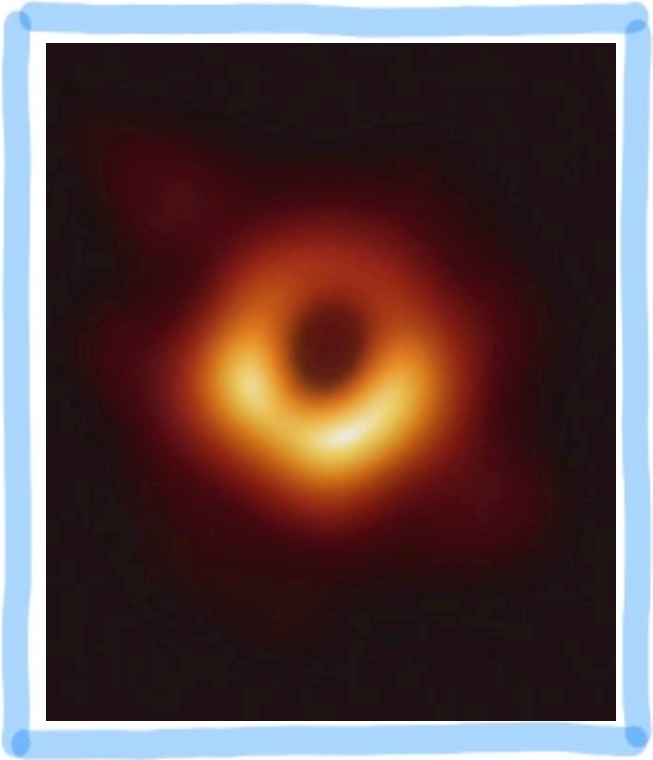
Detecting tampering with the diagnostic images, or quietly upped the radiation levels.

Skin Conditions

Using Deep Learning in diagnosing skin conditions

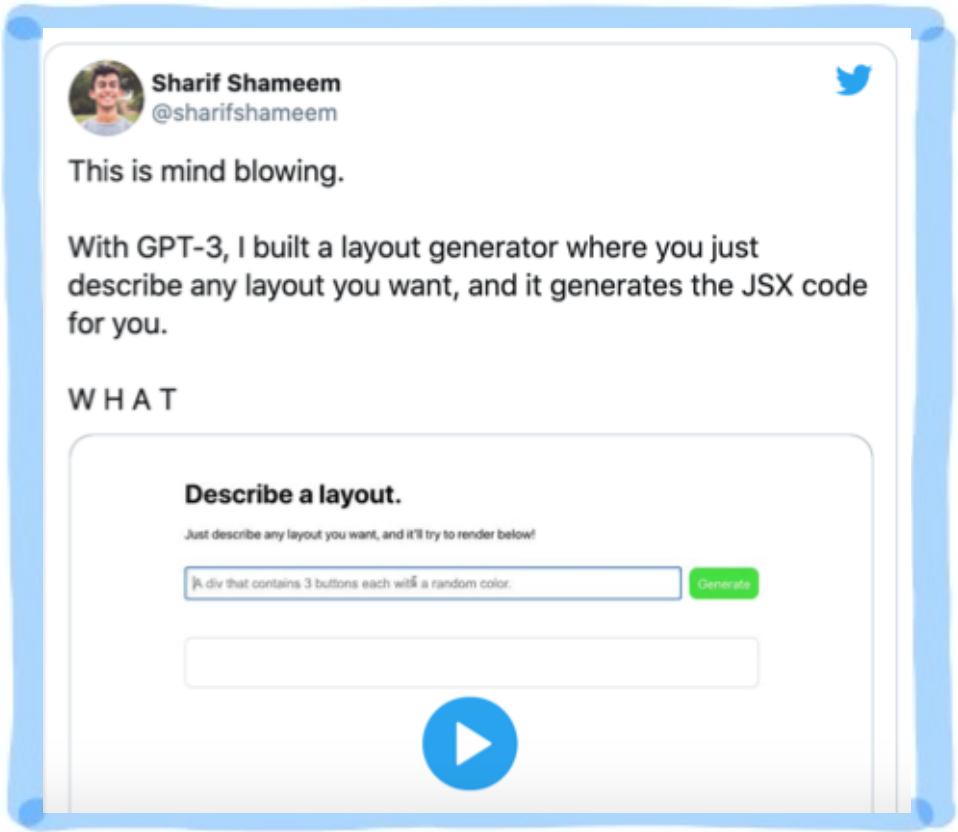
Today's news

Image generation



Katie Bouman's CHIRP produces the first-ever image of a black hole.

Computer Code Generation



The Potential of Data Science

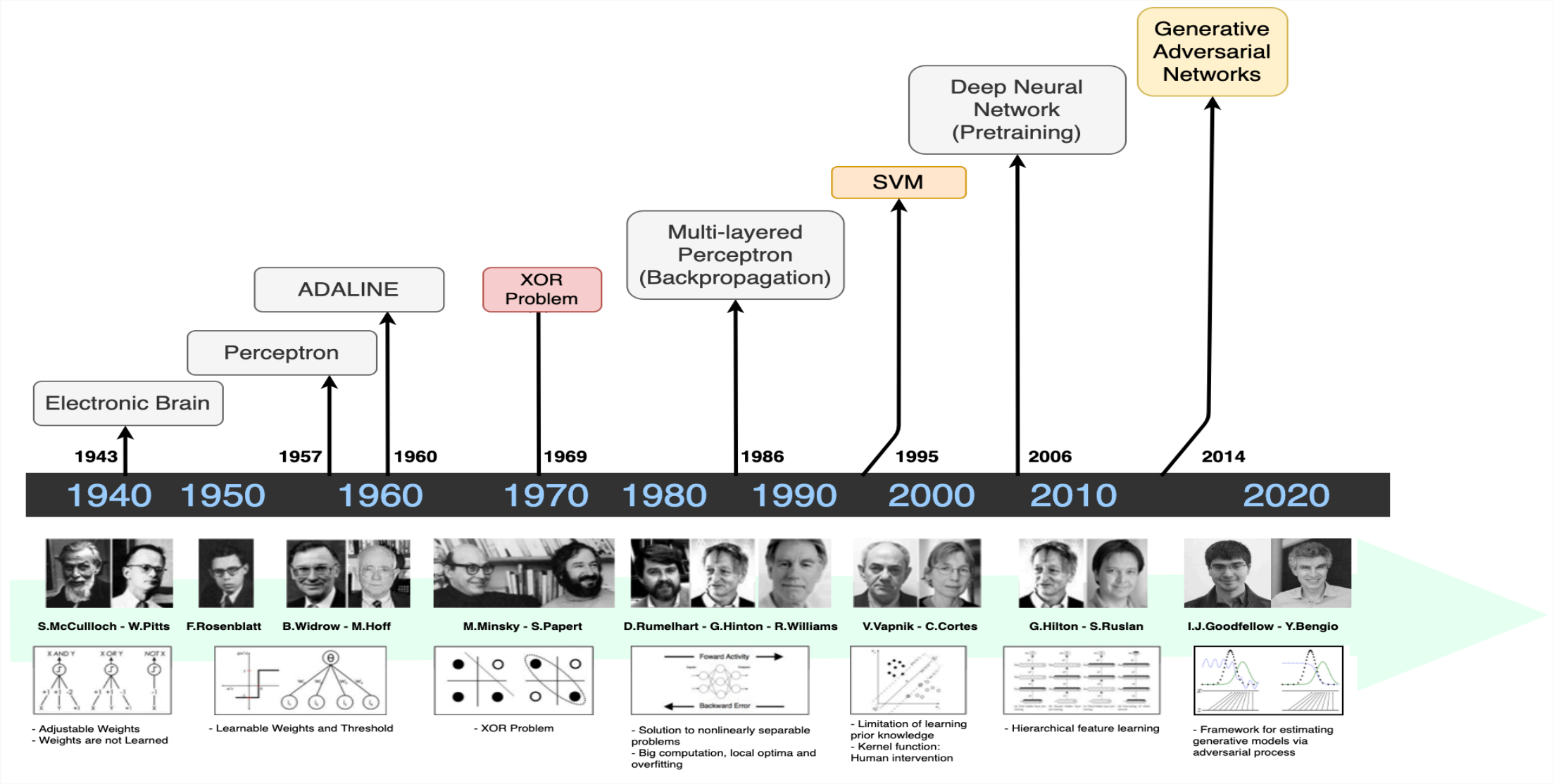
Gender Bias

Some DS models for evaluate job applications show bias in favor of male candidate

Racial Bias

Risk models used in US courts have shown to be biased against non-white defendants

Historical Trends



Historical Trends

Disease prediction

Google's new AI can predict heart disease by simply scanning your eyes

Share on F Share on +

IMAGE: BEN BRAIN/DIGITAL CAMERA MAGAZINE VIA GETTY IMAGES

BY MONICA CHIN

FEB 2018

The secret to identifying certain health conditions may be hidden in our eyes.

Researchers from Google and its health-tech subsidiary Verily announced on Monday that they have successfully created algorithms to predict whether someone has high blood pressure or is at risk of a heart attack or stroke simply by scanning a person's eyes, the *Washington Post* reports.

SEE ALSO: This fork helps you stay healthy

Google's researchers trained the algorithm with images of scanned retinas from more than 280,000 patients. By reviewing this massive database, Google's algorithm trained itself to recognize the patterns that designated people as at-risk.

This algorithm's success is a sign of exciting developments in healthcare on the horizon. As Google fine-tunes the technology, it could one day

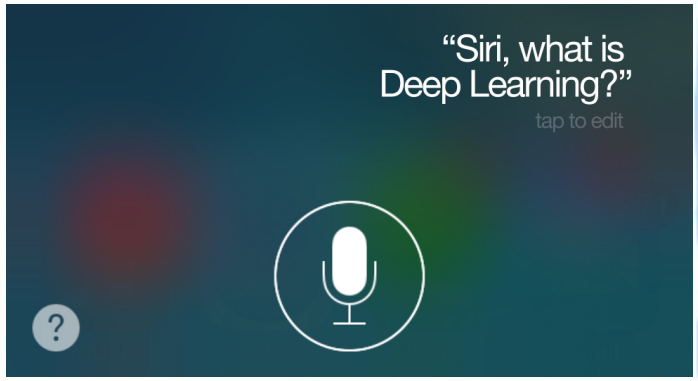
Game strategy

DeepMind

AlphaZero AI beats champion chess program after teaching itself in four hours

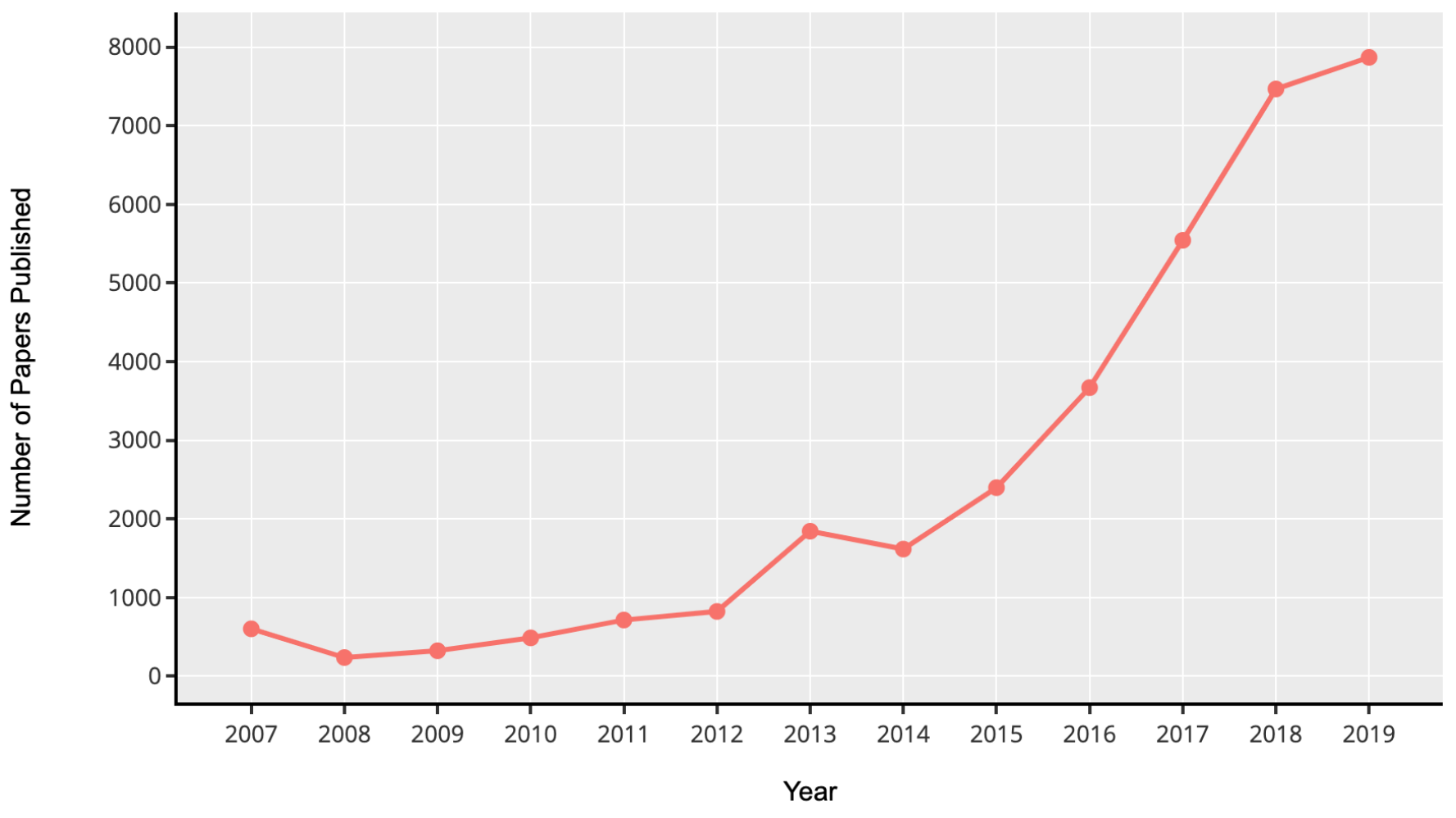
Google's artificial intelligence sibling DeepMind repurposes Go-playing AI to conquer chess and shogi without aid of human knowledge

Natural Language Processing



Historical Trends

ArXiv papers on Machine Learning and Artificial Intelligence: 2007-2019



Outline

1. Introduction to Artificial Neural Networks
2. **Review of basic concepts**
3. Single Neuron Network ('Perceptron')
4. Multi-Layer Perceptron (MLP)

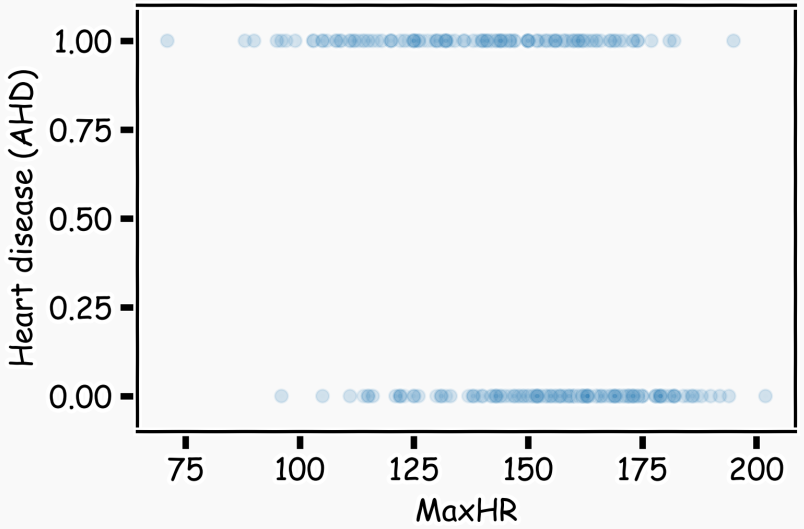
Classification example: Heart Data

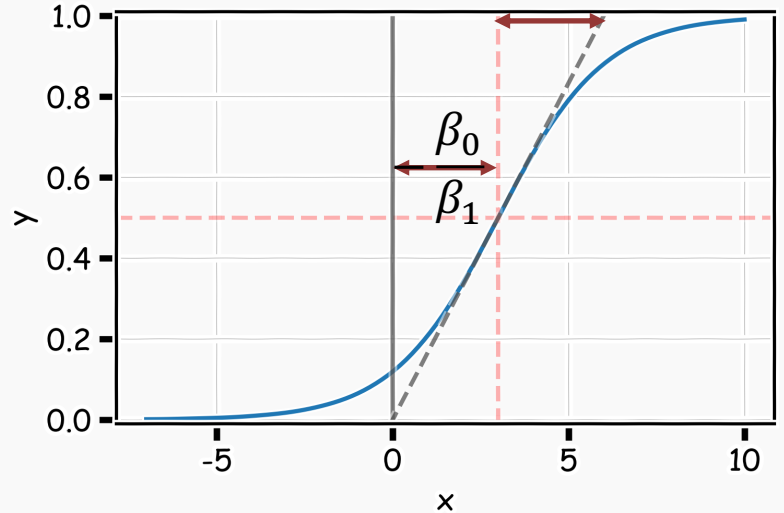
response variable ***Y*** is Yes/No

Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak	Slope	Ca	Thal	AHD
63	1	typical	145	233	1	2	150	0	2.3	3	0.0	fixed	No
67	1	asymptomatic	160	286	0	2	108	1	1.5	2	3.0	normal	Yes
67	1	asymptomatic	120	229	0	2	129	1	2.6	2	2.0	reversable	Yes
37	1	nonanginal	130	250	0	0	187	0	3.5	3	0.0	normal	No
41	0	nontypical	130	204	0	2	172	0	1.4	1	0.0	normal	No

Heart Data: logistic estimation

We'd like to predict whether or not a person has a heart disease. And we'd like to make this prediction, for now, just based on the MaxHR.





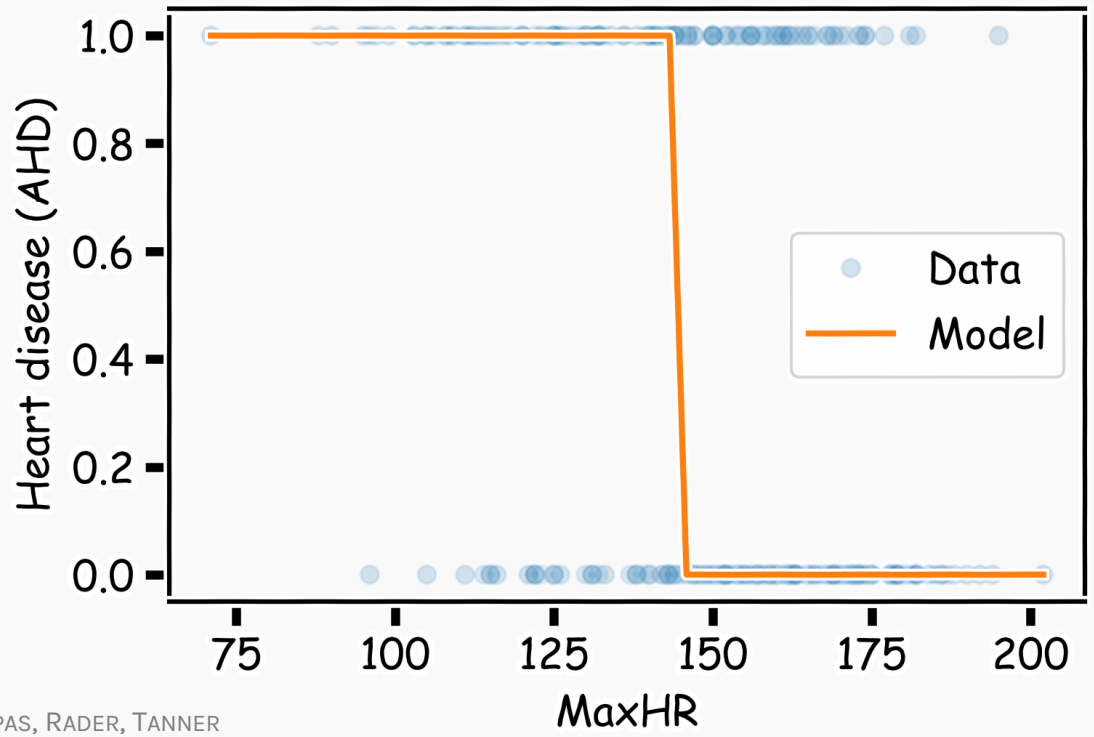
The logistic regression model uses a function, called the ***logistic*** function, to model $P(y = 1)$:

$$P(Y = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$

Logistic Regression

Find the coefficients that minimize the loss function

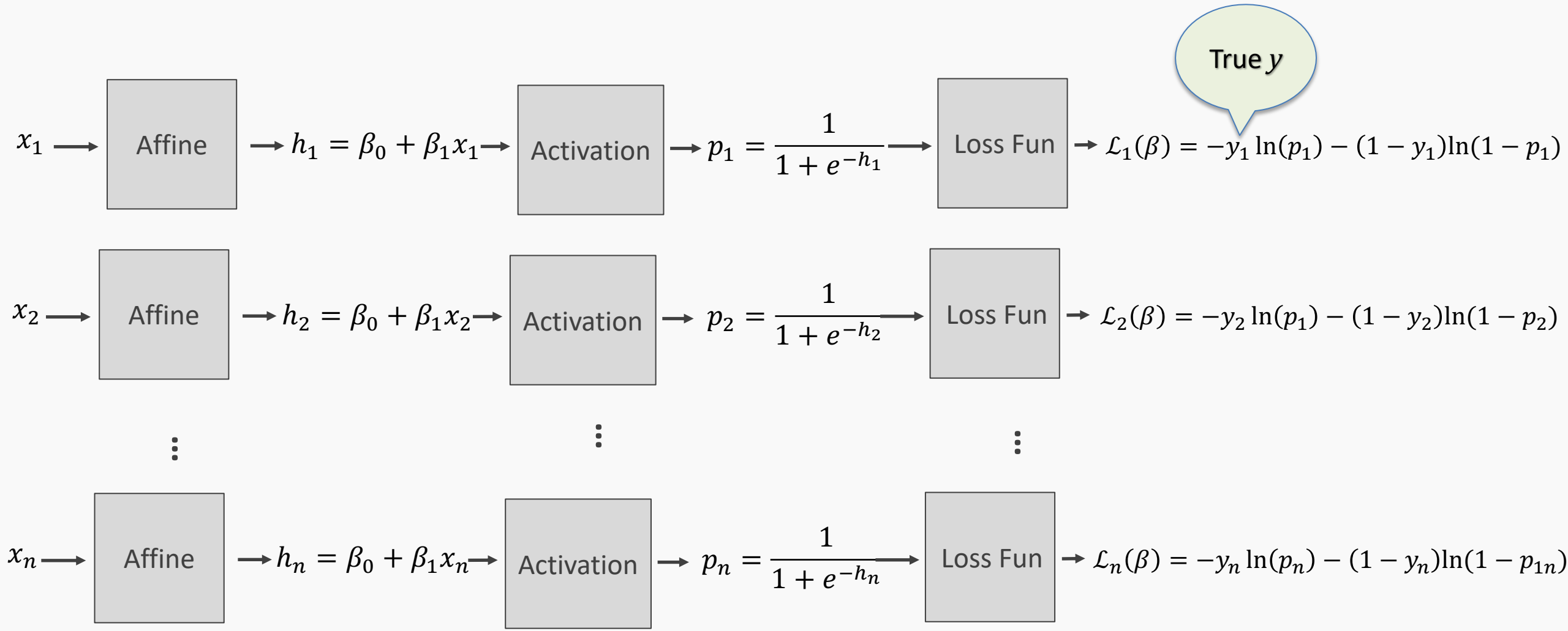
$$\mathcal{L}(\beta_0, \beta_1) = -\sum_i [y_i \log p_i + (1 - y_i) \log(1 - p_i)]$$



Outline

1. Introduction to Artificial Neural Networks
2. Review of basic concepts
3. **Single Neuron Network ('Perceptron')**
4. Multi-Layer Perceptron (MLP)

Logistic Regression Revisited



$$\mathcal{L}(\beta) = \sum_i^n \mathcal{L}_i(\beta)$$

Build our first ANN

n_p: number of predictors
n_o: number of observations

matrix: $n_o \times n_p$

vector: $n_o \times 1$

vector: $n_p \times 1$

scalar

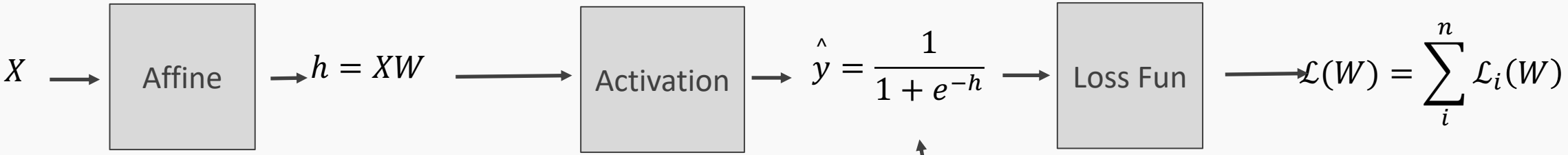
$$X \longrightarrow \text{Affine} \longrightarrow h = \beta_0 + X\beta_1 \longrightarrow \text{Activation} \longrightarrow p = \frac{1}{1+e^{-h}} \longrightarrow \text{Loss Fun} \longrightarrow \mathcal{L}(\beta) = \sum_i^n \mathcal{L}_i(\beta)$$

$$X \longrightarrow \text{Affine} \longrightarrow h = XW + b \longrightarrow \text{Activation} \longrightarrow \hat{y} = \frac{1}{1+e^{-h}} \longrightarrow \text{Loss Fun} \longrightarrow \mathcal{L}(\beta) = \sum_i^n \mathcal{L}_i(\beta)$$

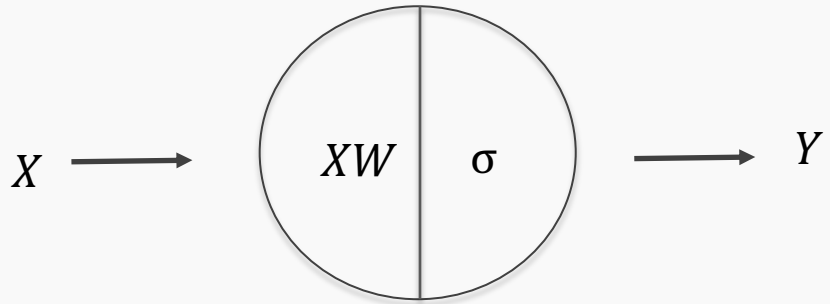
$$X \longrightarrow \text{Affine} \longrightarrow h = XW \longrightarrow \text{Activation} \longrightarrow \hat{y} = \frac{1}{1+e^{-h}} \longrightarrow \text{Loss Fun} \longrightarrow \mathcal{L}(\beta) = \sum_i^n \mathcal{L}_i(\beta)$$

$$X = \begin{bmatrix} 1 & X_{11} & \dots & X_{1p} \\ 1 & \vdots & \dots & \vdots \\ 1 & X_{o1} & \dots & X_{op} \end{bmatrix} \quad \mathrm{W} = \begin{bmatrix} b \\ W_1 \\ \vdots \\ W_p \end{bmatrix}$$

Build our first ANN

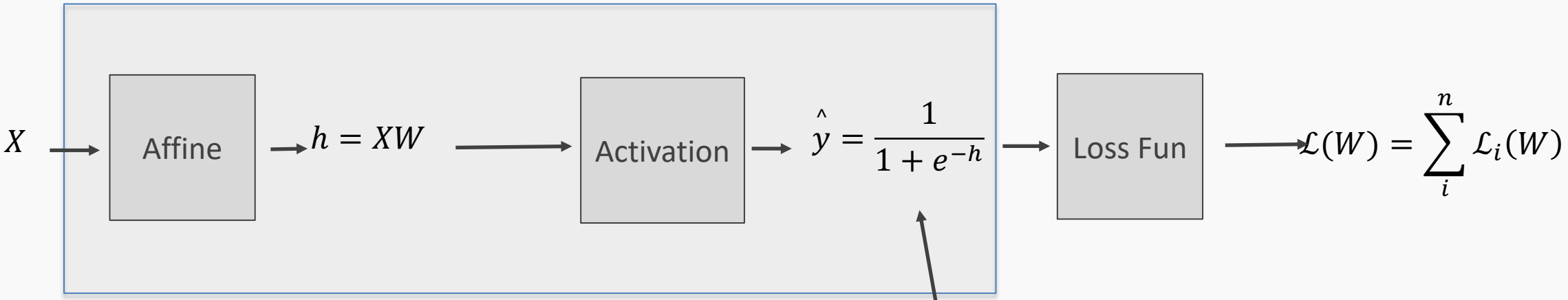


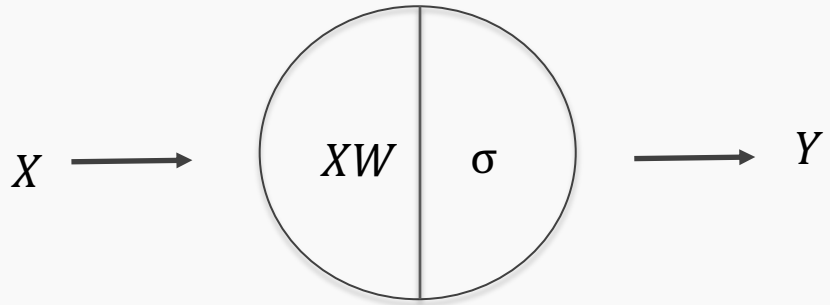
"Sigmoid activation" $\boldsymbol{\sigma}$



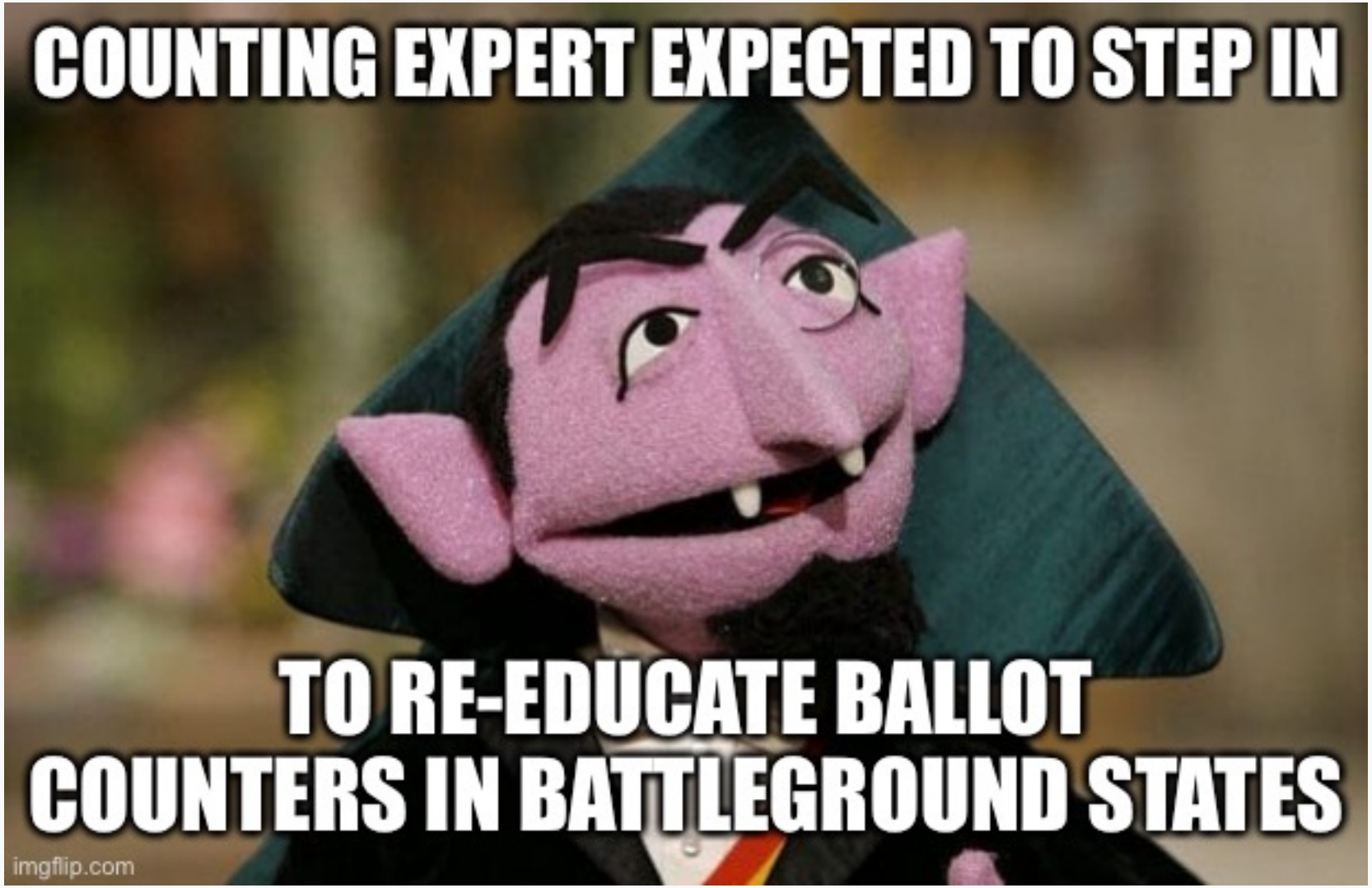
Single Neuron Network aka Perceptron

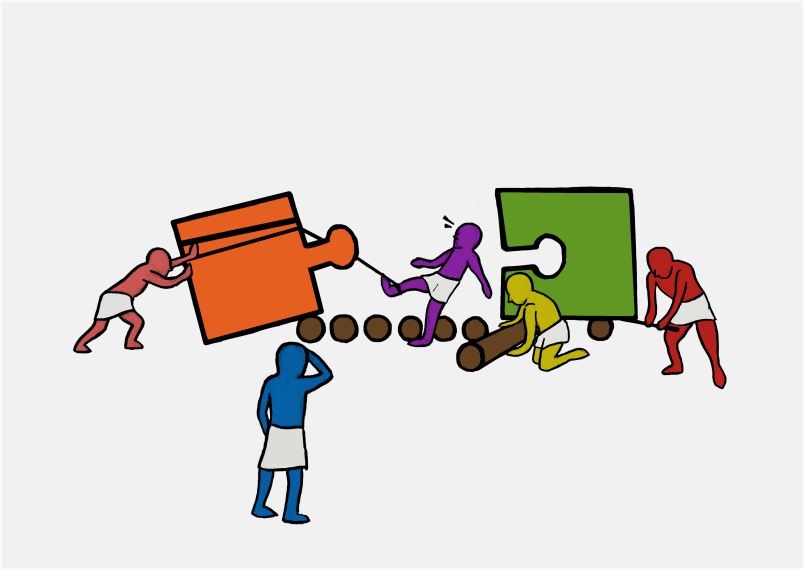
Build our first ANN





Single Neuron Network aka Perceptron


COUNTING EXPERT EXPECTED TO STEP IN
TO RE-EDUCATE BALLOT COUNTERS IN BATTLEGROUND STATES
imgflip.com



Today's lucky student: NOT PAVLOS

Build a Single Neuron by Hand