Cosmology and AI: the role of machine learning

Raul Jimenez

ICREA & ICC-UB
BARCELONA

https://sites.google.com/site/rauljimenez

ICREA | UNIVERSITAT DE BARCELONA | ICCUB | GOBIERNO DE ESPAÑA | MINISTERIO DE ECONOMÍA, INDUSTRIA Y COMPETITIVIDAD

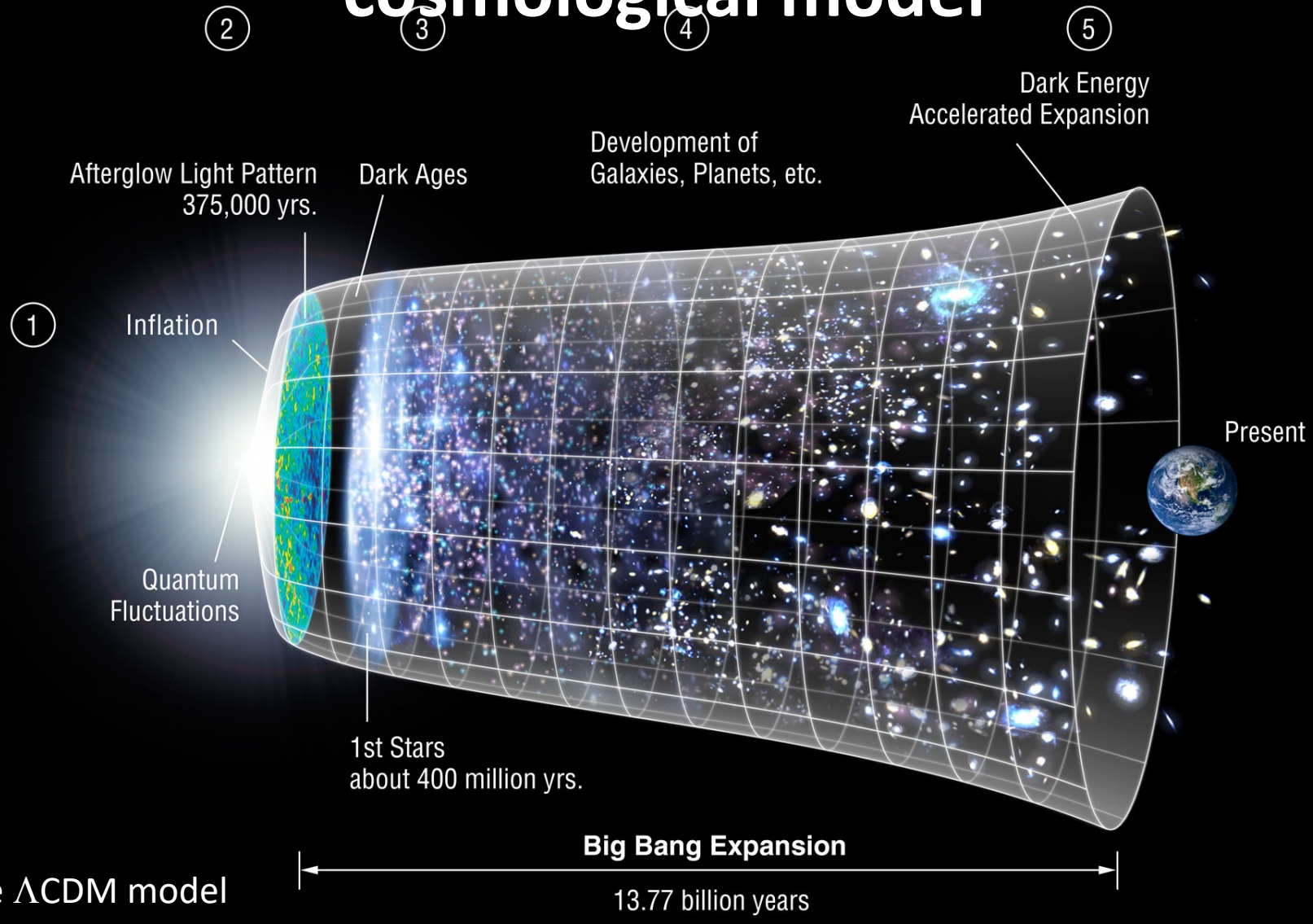
Why is AI crucial in numerical cosmology today and in the future?

What is all the fuss about?

One example: you can simulate universes by just constructing ONE single example using direct integration and then simulate the others by MACHINE LEARNING

See this paper https://arxiv.org/pdf/2001.05519.pdf

The extremely successful standard cosmological model



the ΛCDM model

Precision cosmology

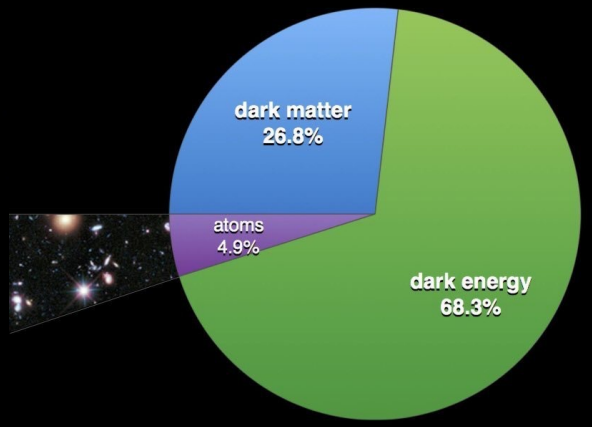
Early 2000s'

ΛCDM: The standard cosmological **model**

Just 6 numbers…..

describe the Universe composition and evolution

Homogenous background



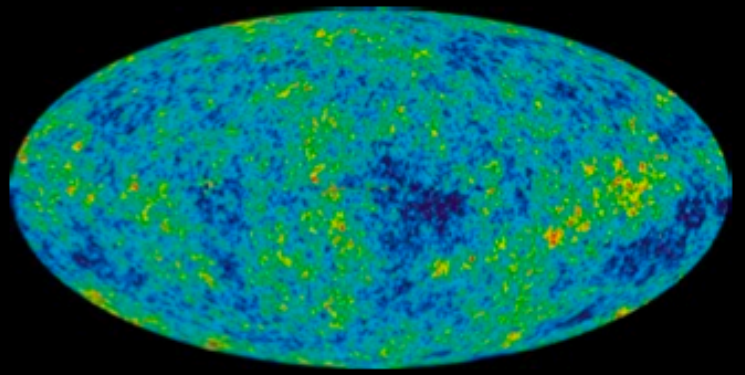
@AstroKatie/Planck13

- atoms 4%
- cold dark matter 23%

$\Omega_b, \Omega_c, \Omega_\Lambda, H_0, \tau$

Λ? CDM?

Perturbations



A_s, n_s

- nearly scale-invariant
- adiabatic
- Gaussian

ORIGIN??

Cosmology is special

We can’t make experiments, only observations

We have to use the entire Universe as a detector:
the detector is given, we can’t tinker with it.

A mixed blessing

The curse of cosmology

We only have one observable universe

We can only make observations (and only of the observable Universe)
not experiments: we fit models (i.e. constrain numerical values of parameters) to the observations: (Almost) <u>any statement is model dependent</u>

*"Gastrophysics"** and non-linearities get in the way

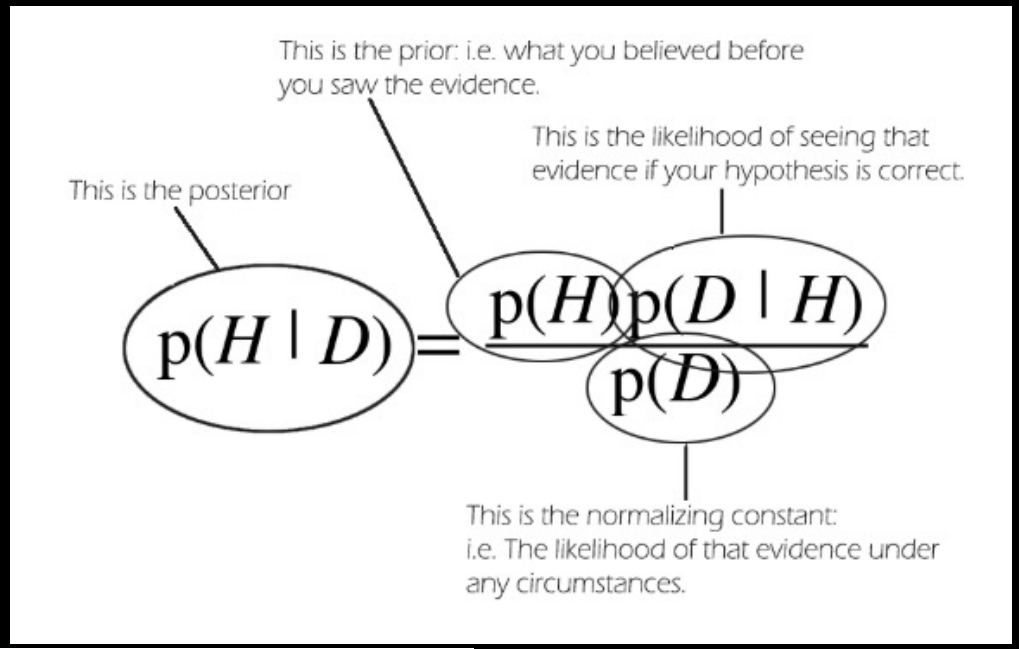
….And the Blessing

We can observe all there is to see

** Not a typo, means complex astrophysics that is poorly understood/hard to model*

challenges

Big data; Cosmology is special we only observe one sky; we only fit models



$$p(D|\mathcal{H}) = \int p(D|\alpha, \mathcal{H}) p(\alpha|\mathcal{H}) d\alpha$$

Evidence Likelihood prior

What is a prior? What to use?

Exp(accuracy-complexity)

Prior choice: unconscious bias

There is a lot of noise out there, must be clarified.

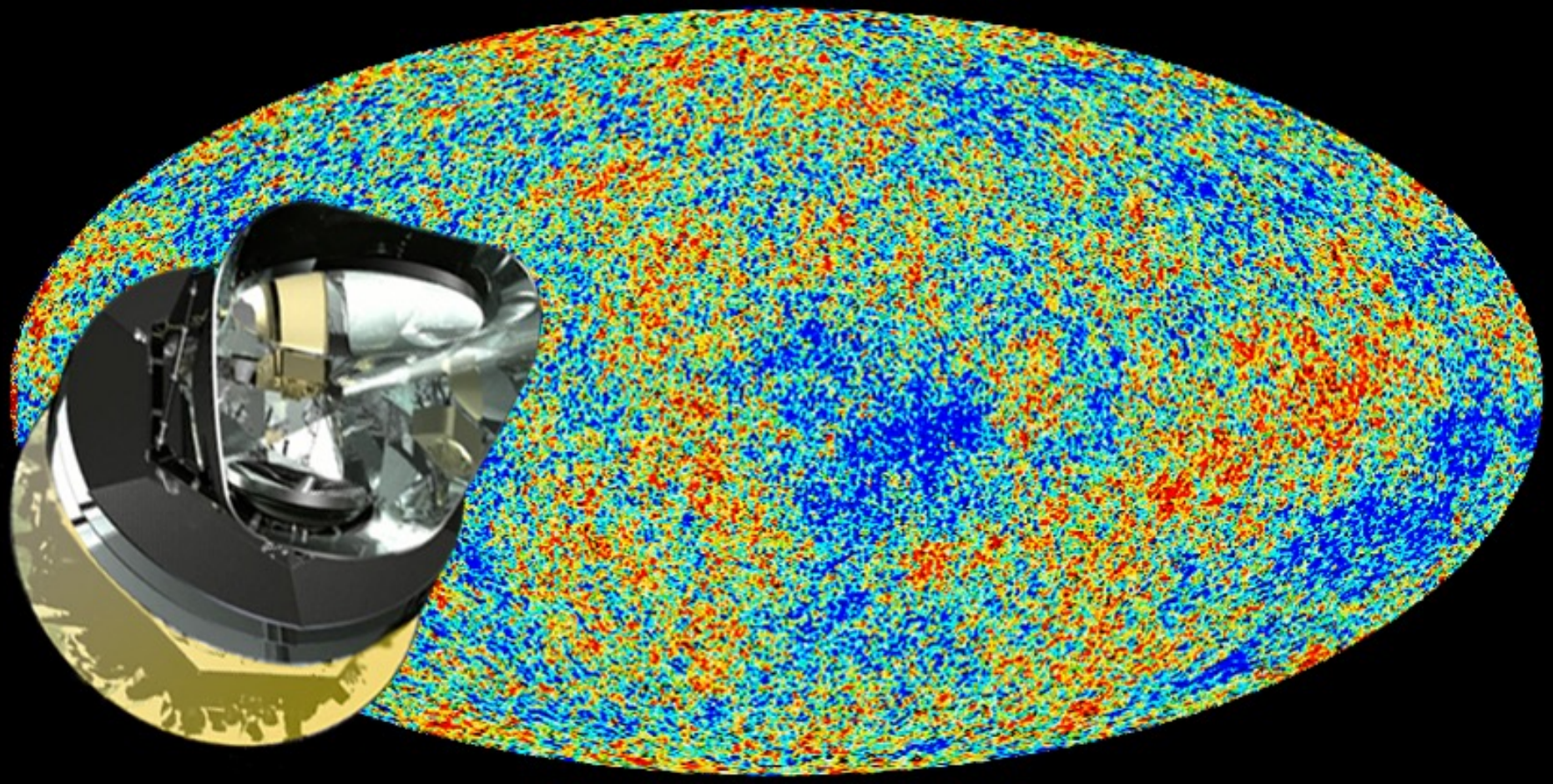
Gist: what is a prior?

- Information that “reflects our state of belief before the data arrived.”
  - use some information about the underlying physical theory/mechanism
  - scientist’s a priori choice to (not) have a personal preference
- Information coming from previous experiments (e.g., “CMB prior”)
- The prior that is most easily overwritten by the data for a given experiment (“Objective Bayesian”)

This choice matters **a lot** especially for model comparison!!!

Coincidences (as told to me by Fergus Simpson)

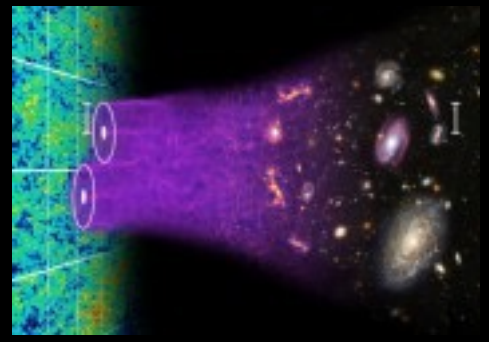
Example of an ultimate experiment

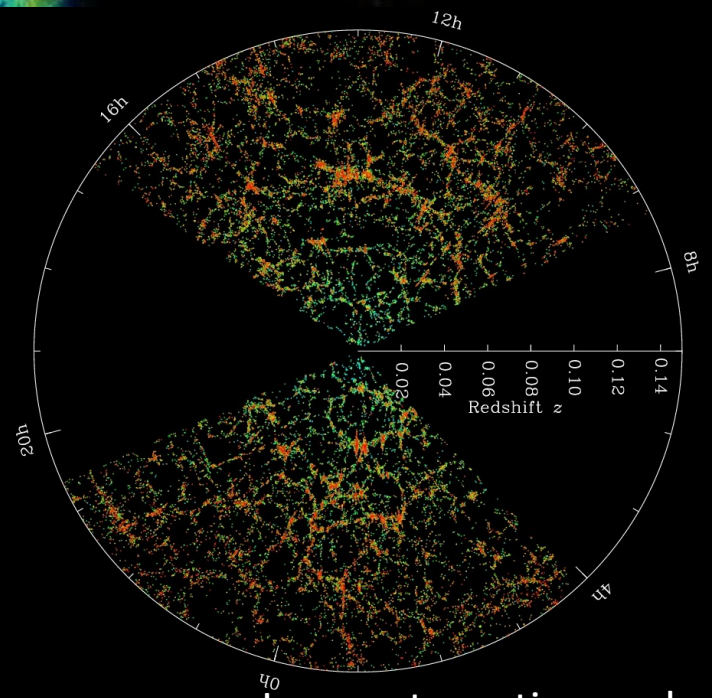


Planck

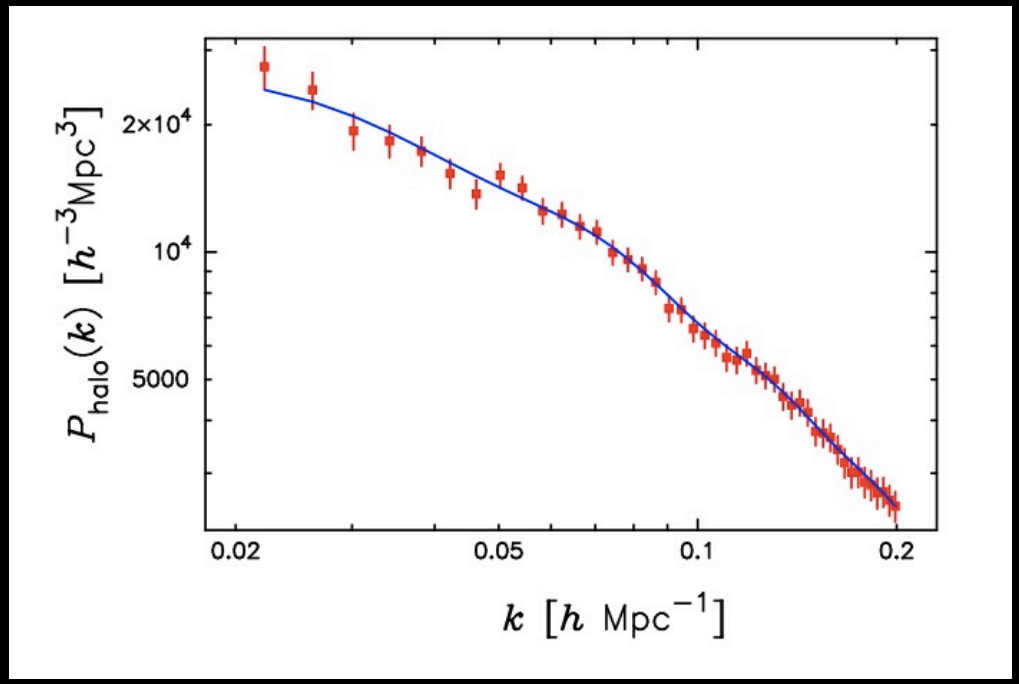
But also ACT, SPT, and in the near future S4 and SO.

Primary CMB temperature information content has been saturated. The near future is large-scale structure.





SDSS LRG galaxies power spectrum (Reid et al. 2010)



13 billion years of gravitational evolution

Longer-term timescale: CMB polarization

Physical information from large-scale structure

Clustering

What are the constituents of matter?
What is the physics of inflation?
e.g. Neutrino masses, Primordial P(k),
Nature of dark matter,
growth of perturbations

Standard ruler
(geometry)

What is the expansion history of the Universe?
e.g. Dark energy

Redshift space distortions

How does structure form within this background?
e.g. modified gravity, GR

Understanding cosmic acceleration

Large-scales

Homogenetiy, non-gaussianity

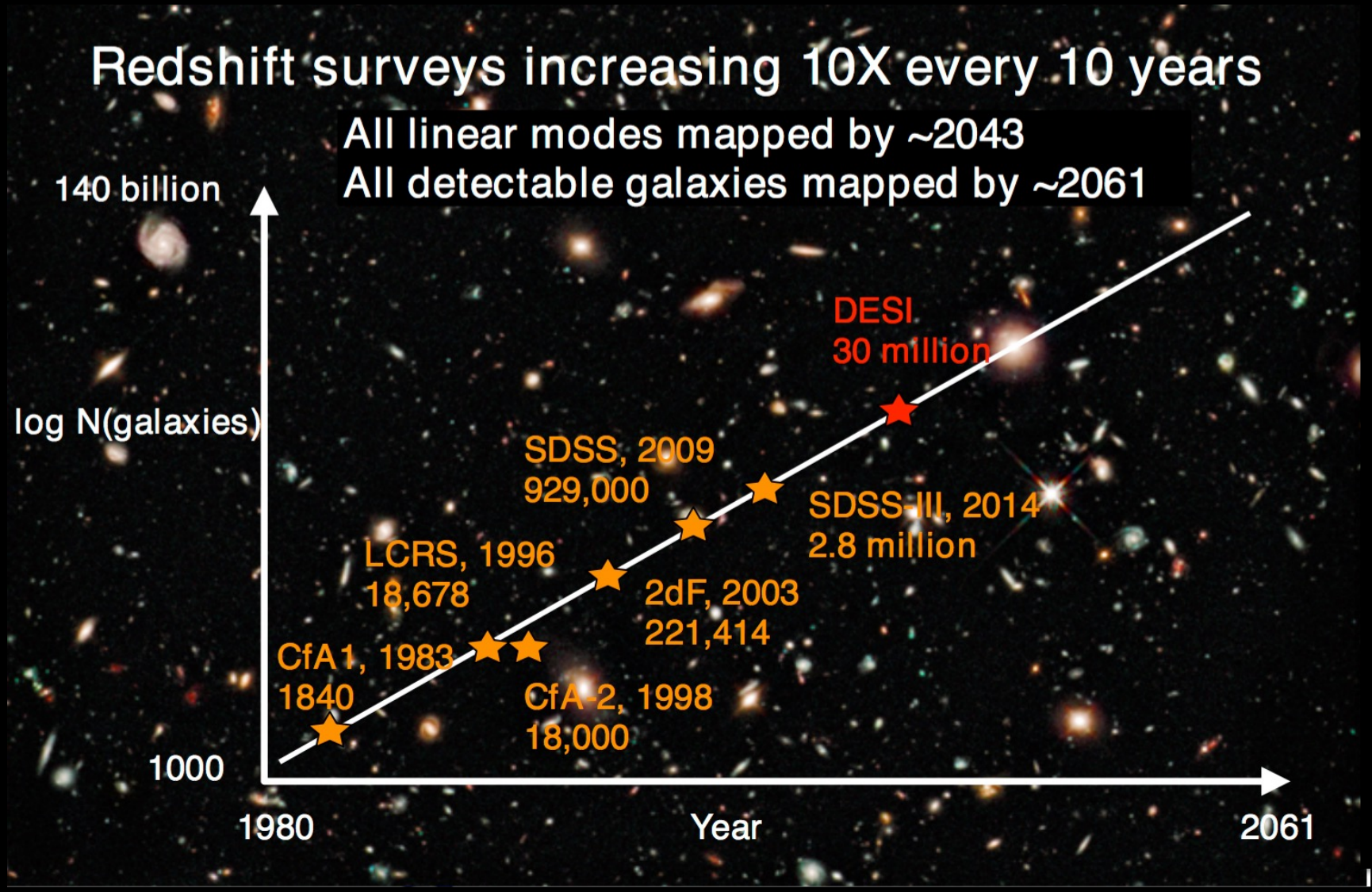
Spectral analysis

other non-cosmological info
e.g. Galaxy formation

May be Key to ensure robustness

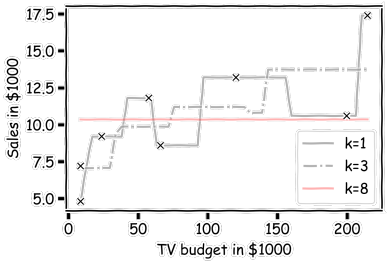
Fig. adapted from W. Percival

Golden age or Gold rush?



Courtesy of D. Schlegel

Linear Models



- Note that in building our kNN model for prediction, we did not compute a closed form for $\hat{f}$.
- What if we ask the question:
- *"how much more sales do we expect if we double the TV advertising budget?"*
- **Alternatively**, we can build a model by first assuming a simple form of f:
-

$$f(x) = \beta_0 + \beta_1 X$$

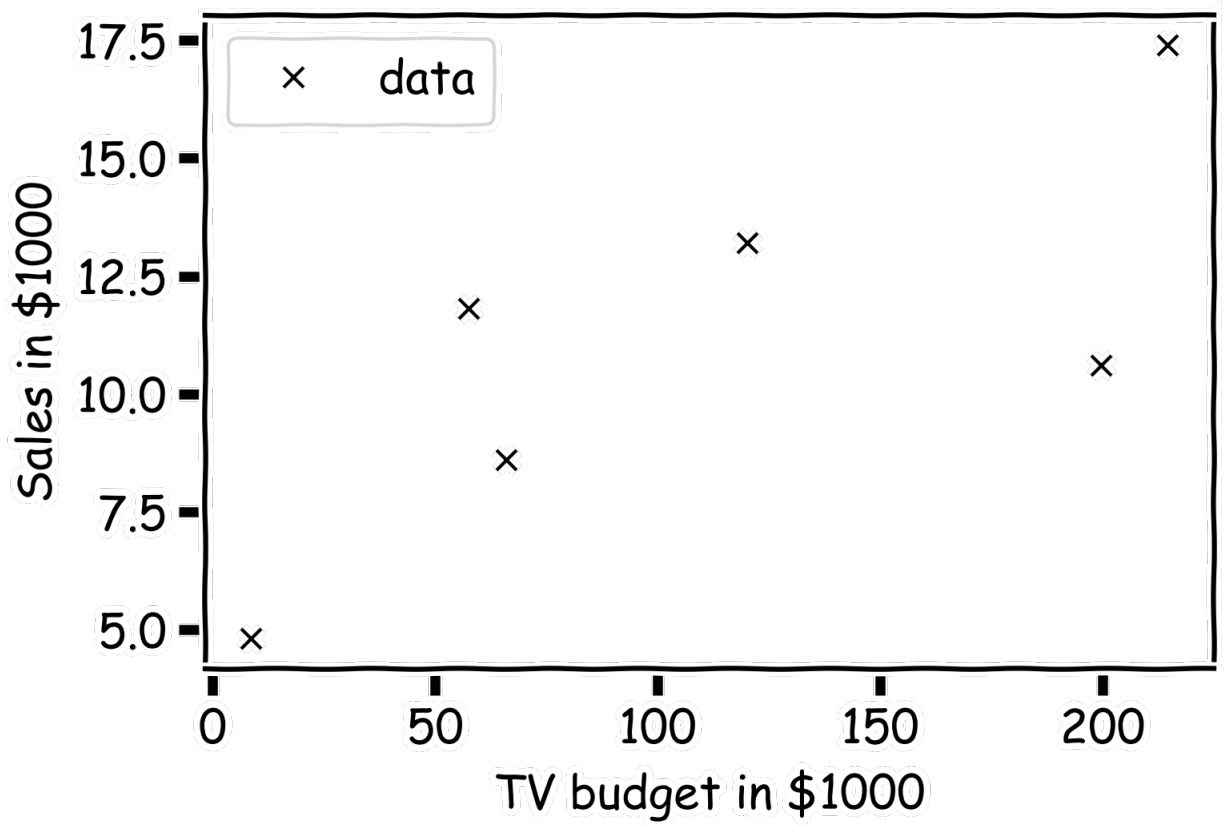
Linear Regression

- … then it follows that our estimate is:
-
$$\widehat{Y} = \widehat{f}(X) = \widehat{\beta_1} X + \widehat{\beta_0}$$
-

- where $\hat{\beta}_1$ and $\hat{\beta}_0$ are **estimates** of β_1 and β_0 respectively, that we compute using observations.

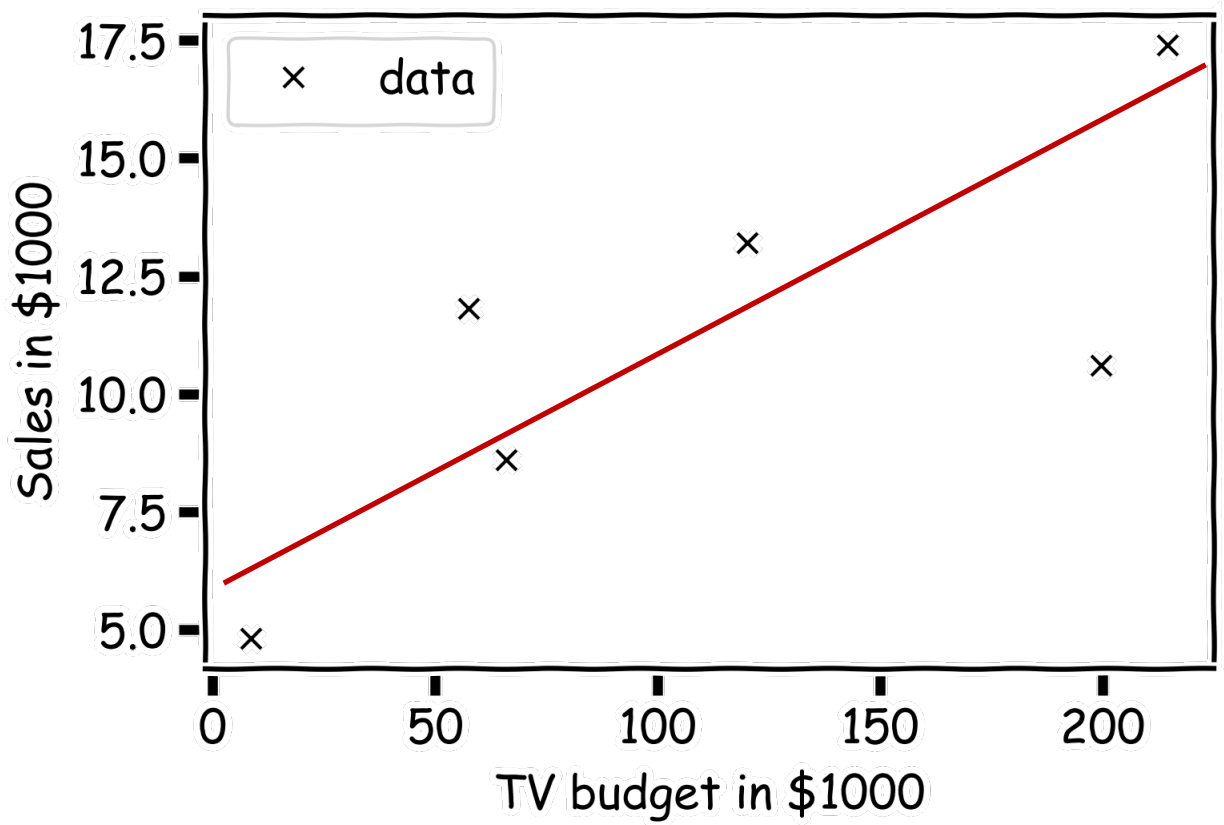
Estimate of the regression coefficients

For a given data set



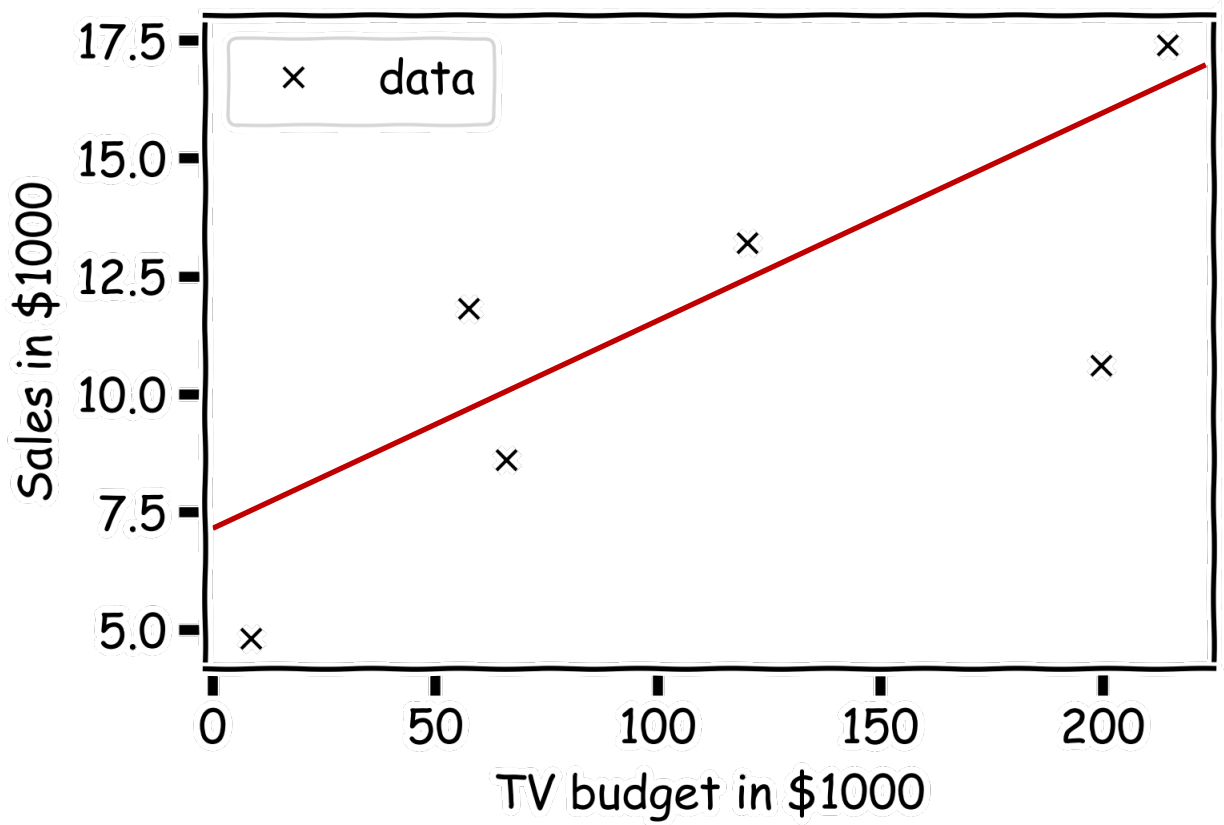
Estimate of the regression coefficients (cont)

Is this line good?



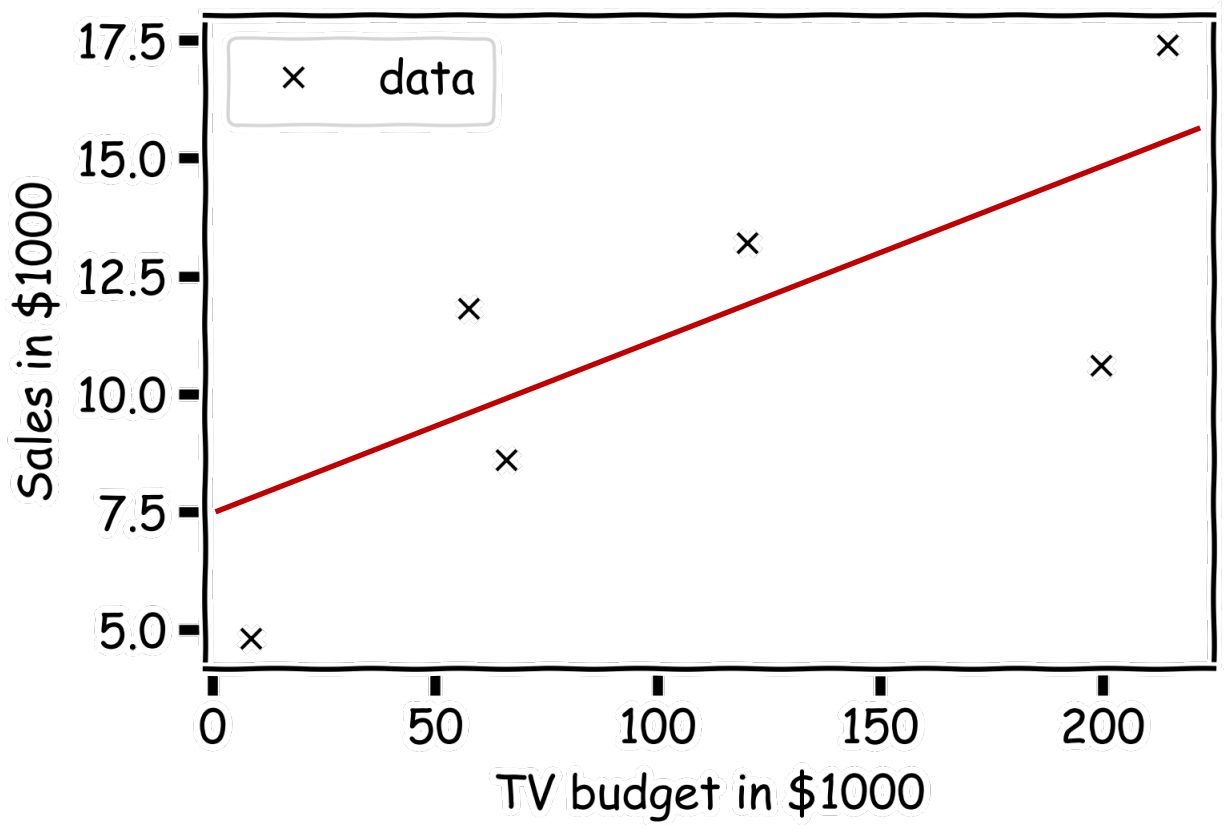
Estimate of the regression coefficients (cont)

Maybe this one?



Estimate of the regression coefficients (cont)

Or this one?



Error Evaluation

In order to quantify how well a model performs, we aggregate the errors and we call that the ***loss*** or ***error*** *or* ***cost function.***

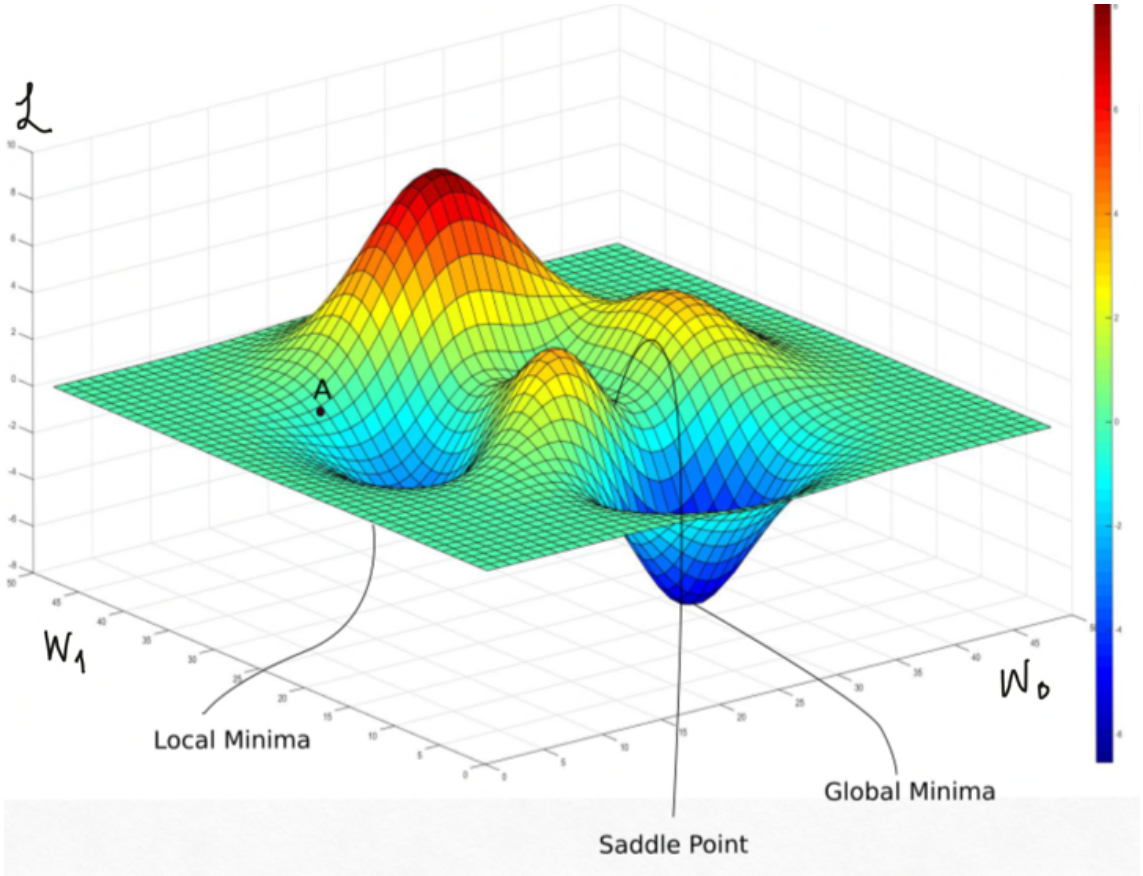
A common loss function for quantitative outcomes is the **Mean Squared Error (MSE):**

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2$$

Note: Loss and cost function refer to the same thing. Cost usually refer to the total loss where loss refers to a single training point.

Optimization

- How does one minimize a loss function?



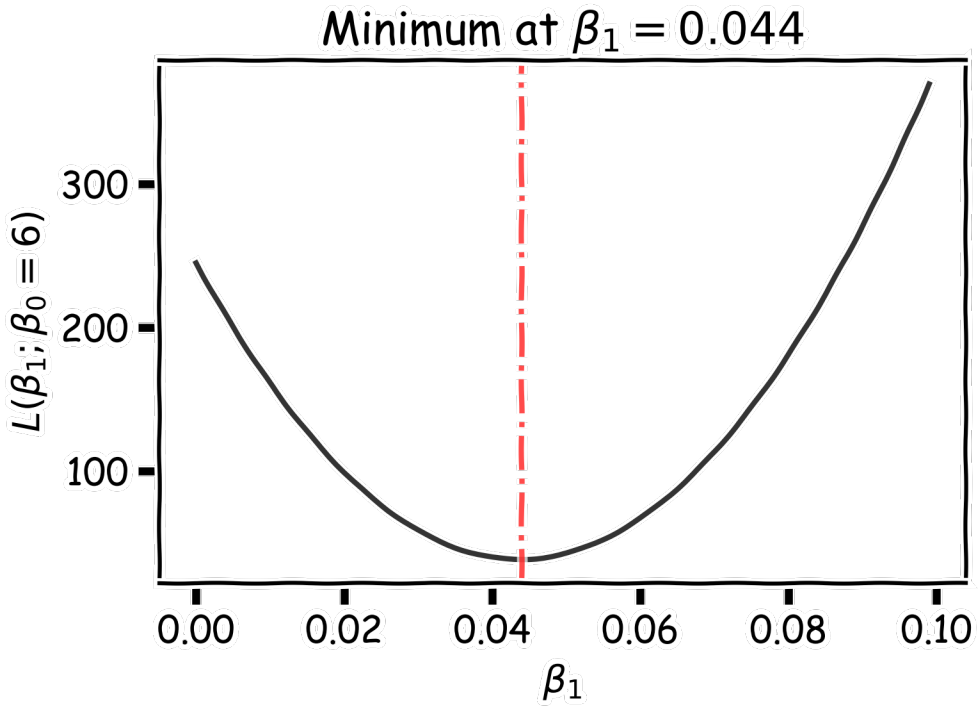
The global minima or maxima of $L(\beta_0, \beta_1)$ must occur at a point where the gradient (slope)

$$\nabla L = \left[\frac{\partial L}{\partial \beta_0}, \frac{\partial L}{\partial \beta_1}\right] = 0$$

- Brute Force: Try every combination
- Exact: Solve the above equation
- Greedy Algorithm: Gradient Descent

Optimization: Estimate of the regression coefficients

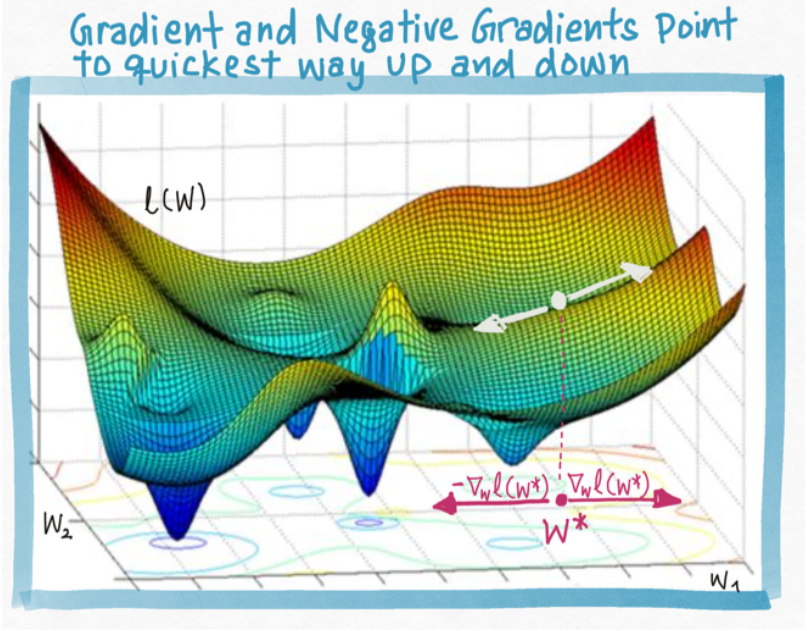
- **Brute force**
- A way to estimate $\text{argmin}_{\beta_0,\beta_1} L$ is to calculate the loss function for every possible β_0 and β_1. Then select the β_0 and β_1 where the loss function is minimum.
- E.g. the loss function for different β_1 when β_0 is fixed to be 6:

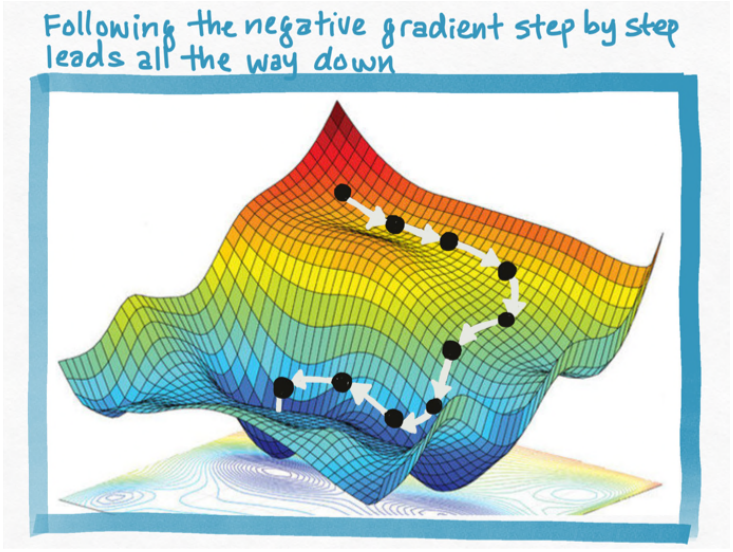


Very **computationally expensive** with many coefficients

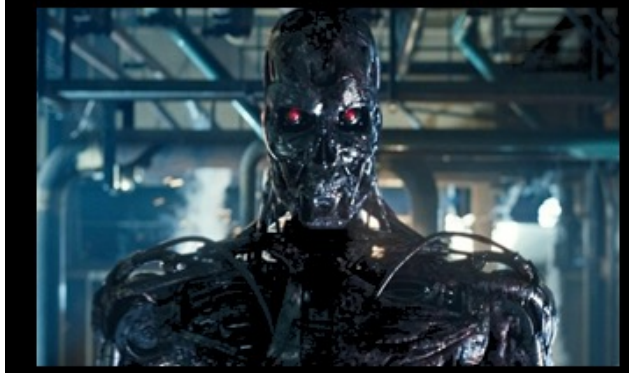
Gradient Descent

- When we can't analytically solve for the stationary points of the gradient, we can still exploit the information in the gradient.
- The gradient ∇L at any point is the direction of the steepest increase. The negative gradient is the direction of steepest decrease.
- By following the –ve gradient, we can eventually find the lowest point.
- This method is called Gradient Descent. [MORE ON THIS LATER IN THE COURSE]

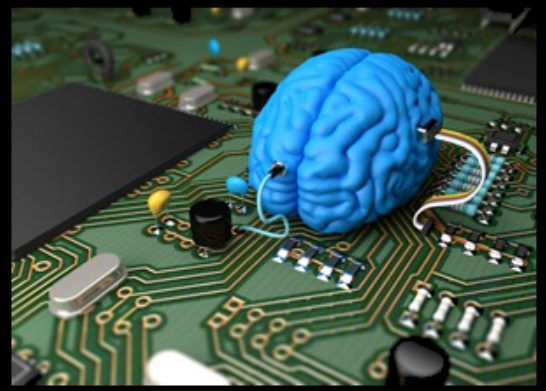




Deep Learning



What society thinks I do



What my friends think I do

What other computer scientists think I do

What mathematicians think I do

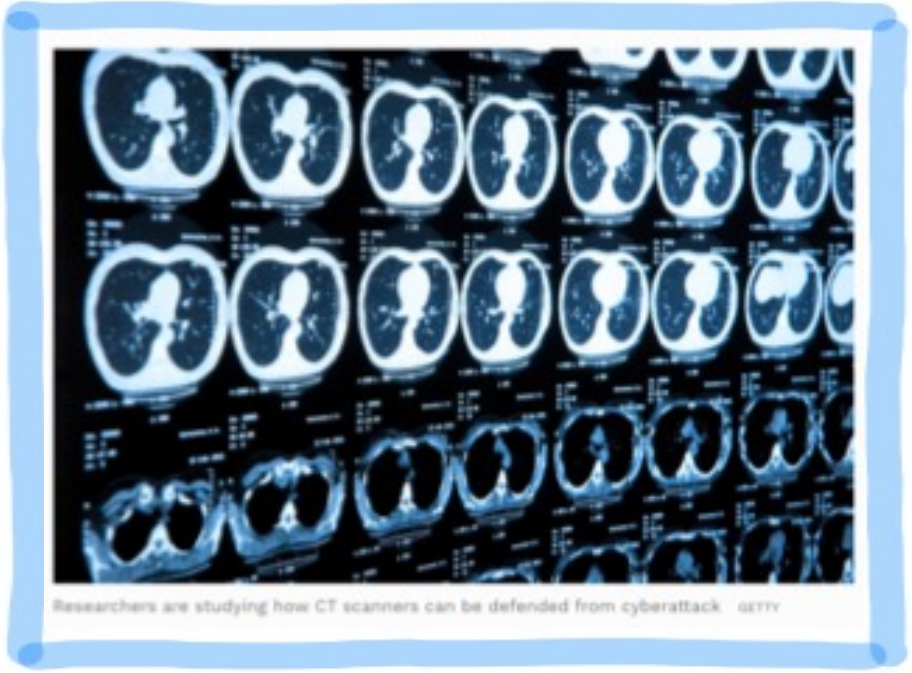
What I think I do

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

What I actually do

Today's news

Stopping Cyperattacks



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

Skin Conditions

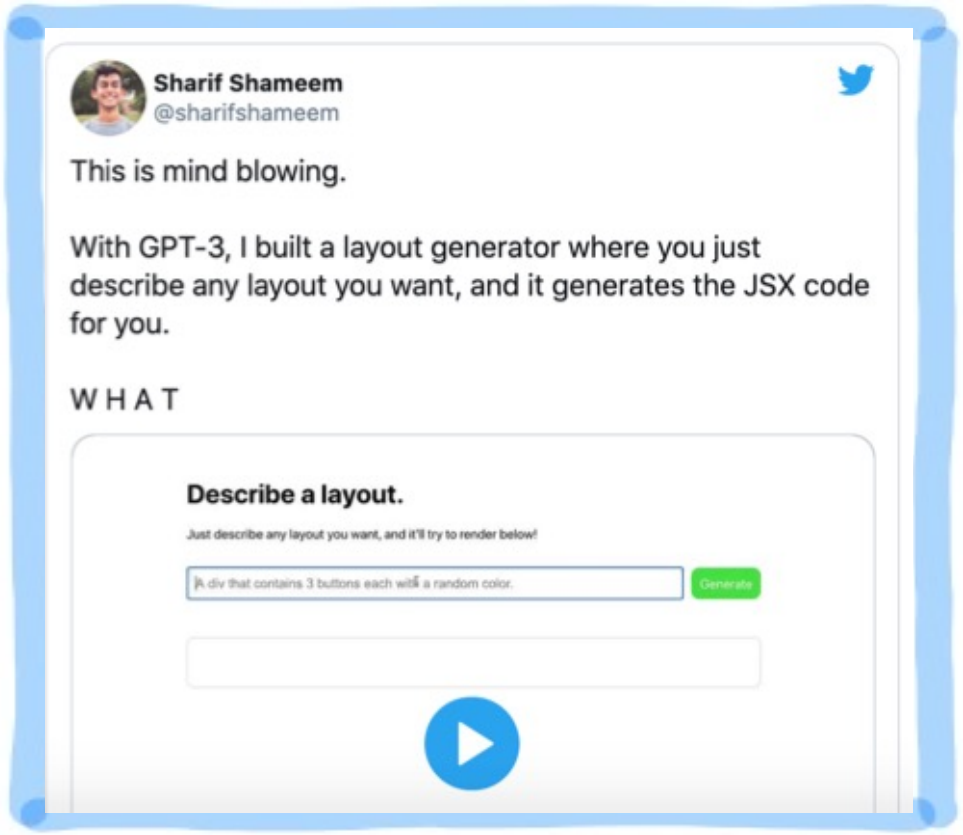
Using Deep Learning in diagnosing skin conditions

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

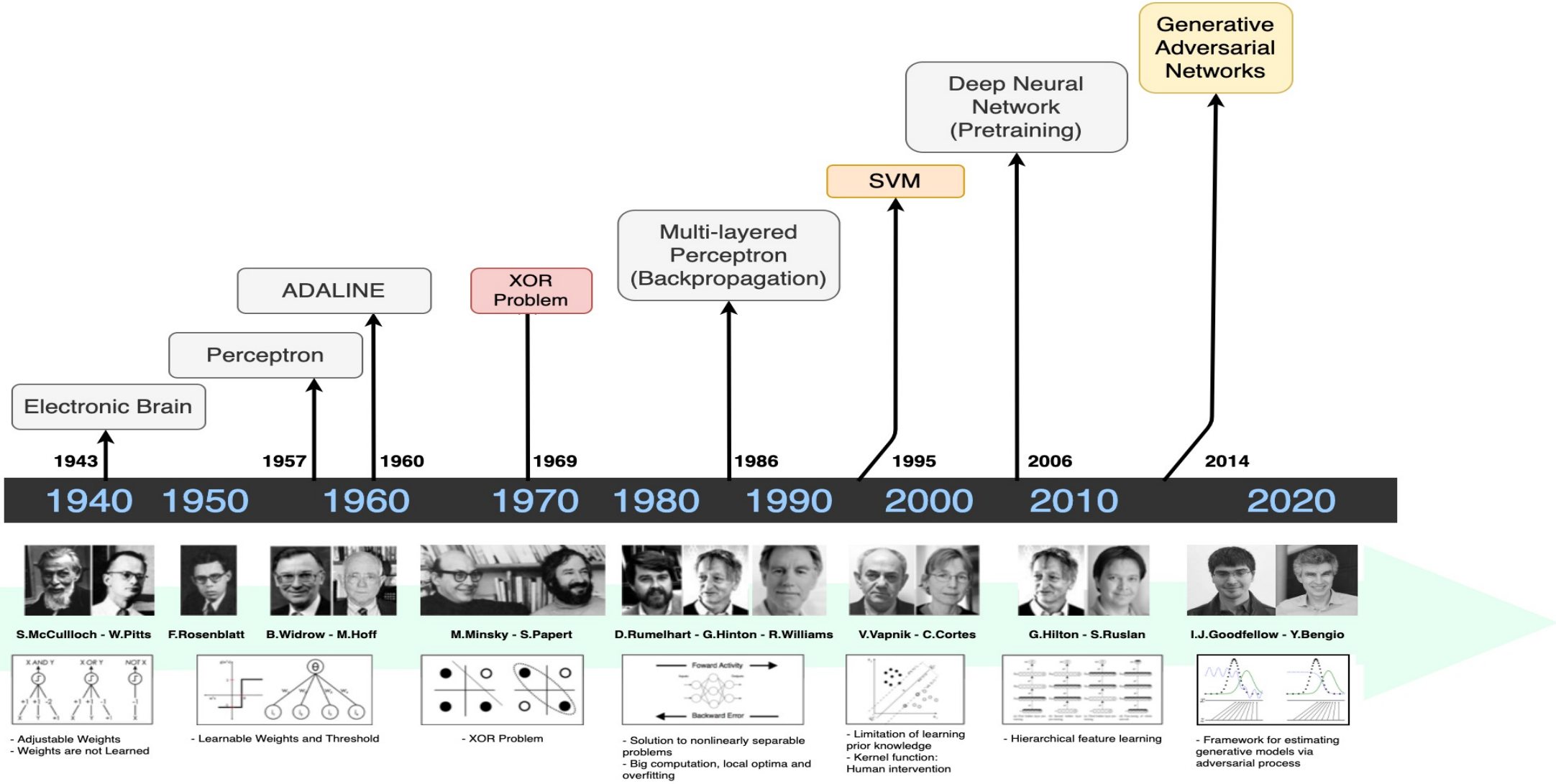
Gender Bias

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

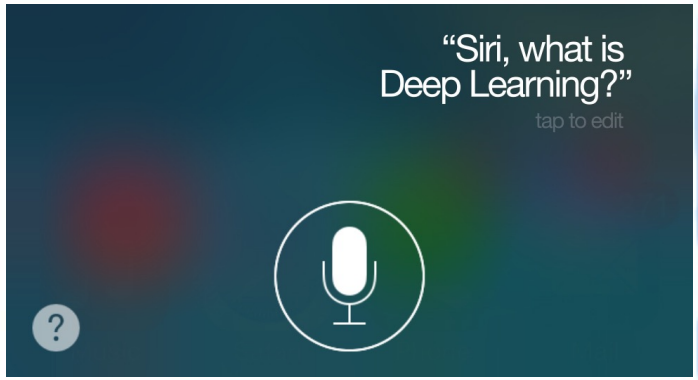
Game strategy

DeepMind

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

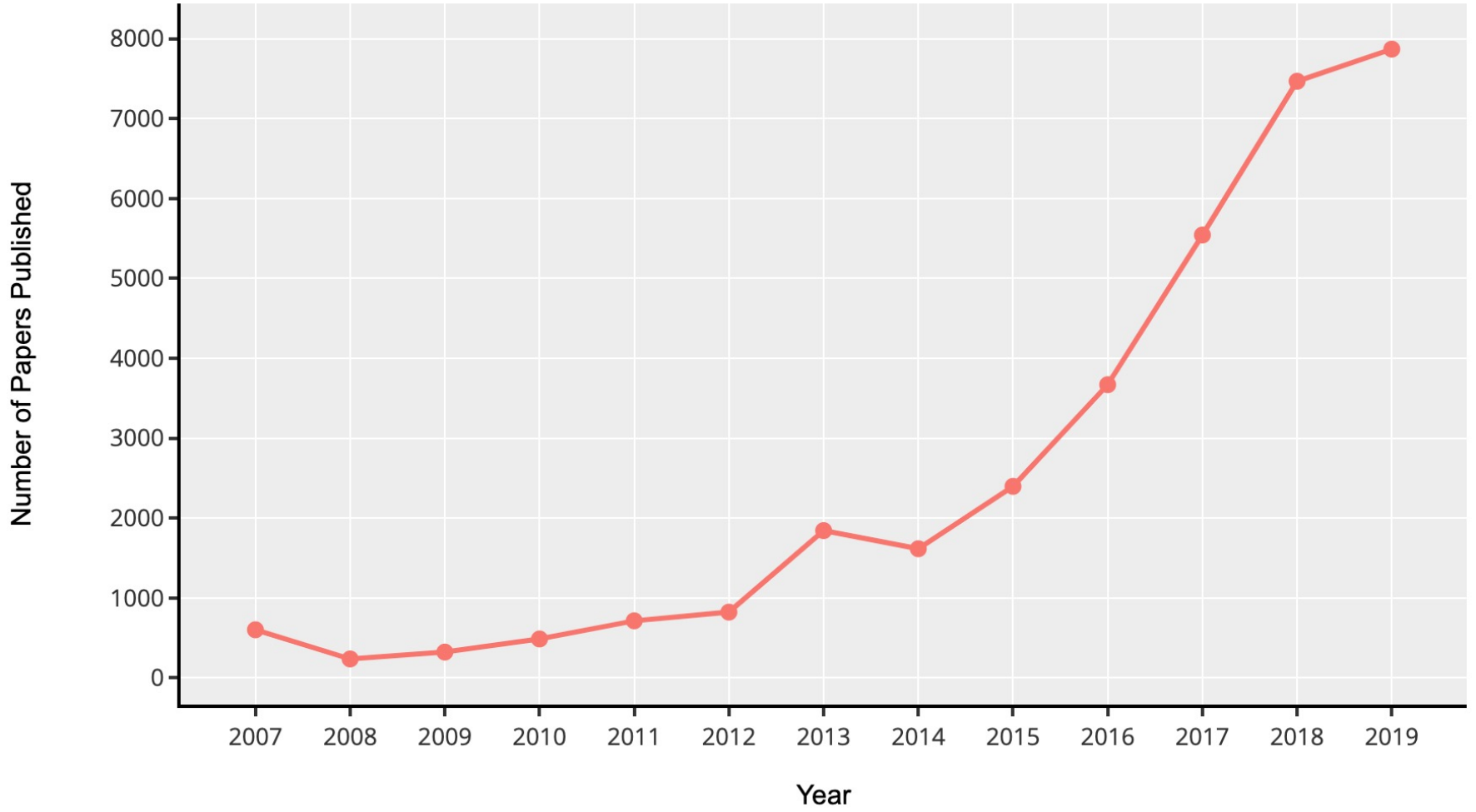
Google's artificial intelligence sibling DeepMind repurposes Go-playing AI to conquer chess and shogi without aid of human knowledge

Natural Language Processing

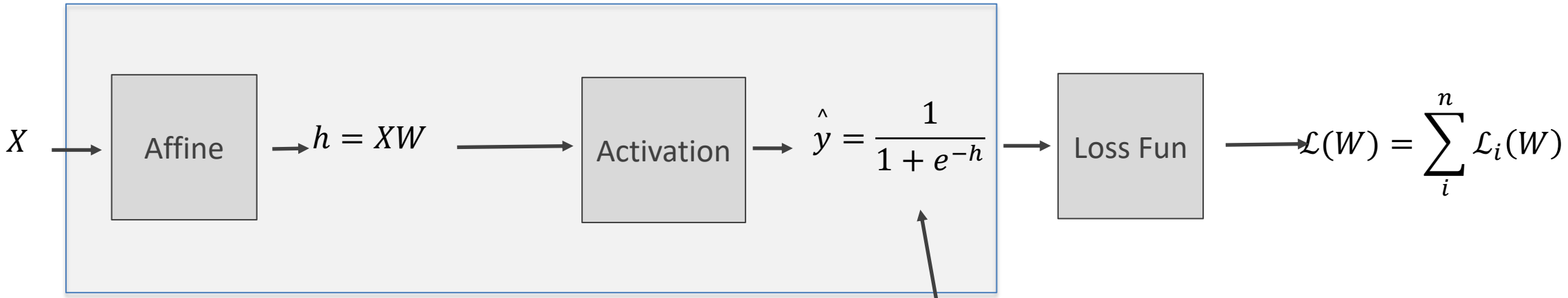


Historical Trends

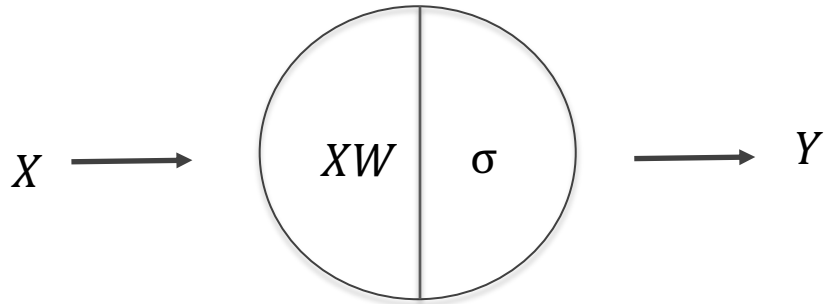
ArXiv papers on Machine Learning and Artificial Intelligence: 2007-2019



Build our first ANN

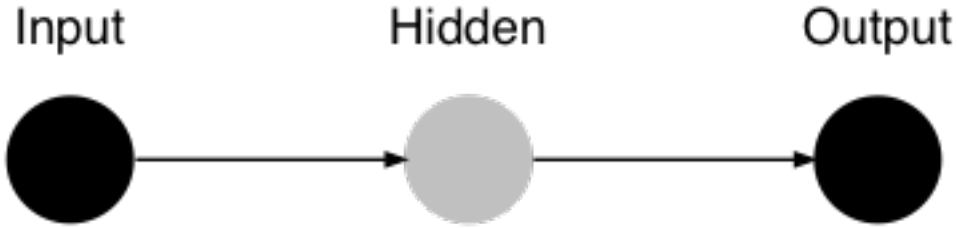


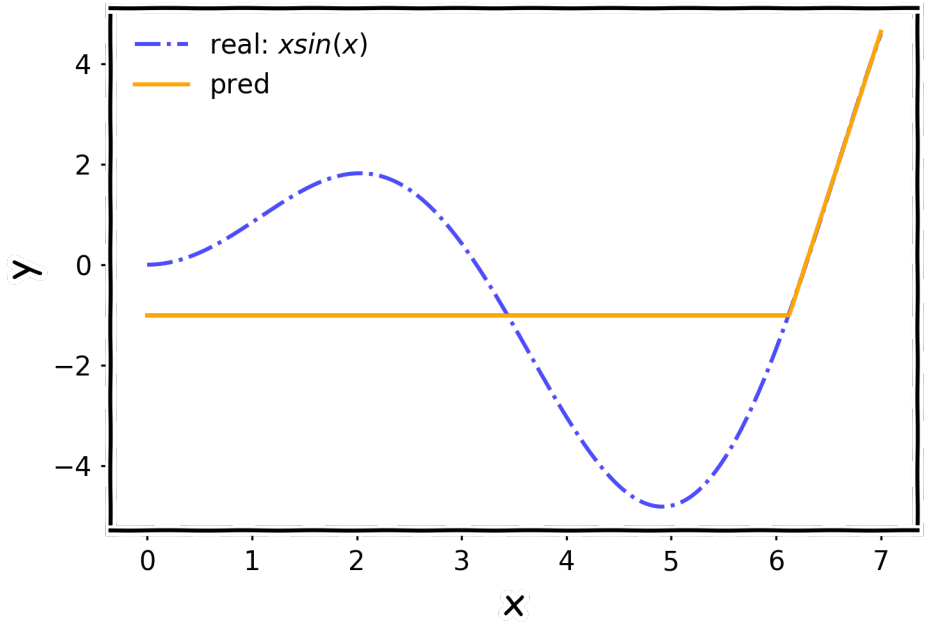
"Sigmoid activation" $\boldsymbol{\sigma}$



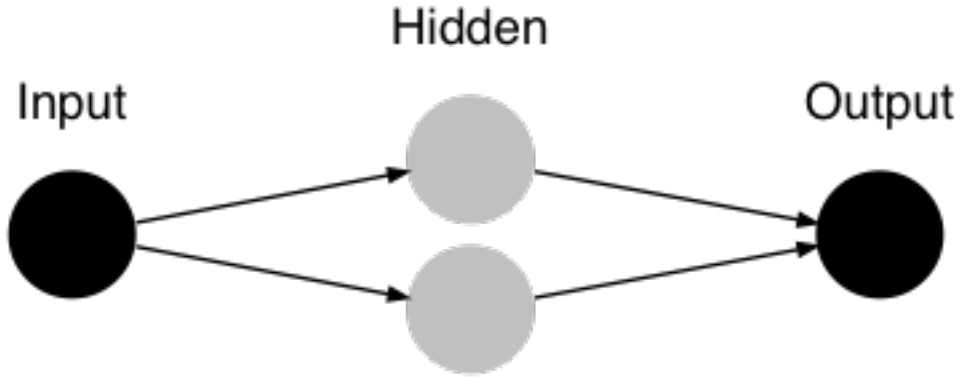
Single Neuron Network aka Perceptron

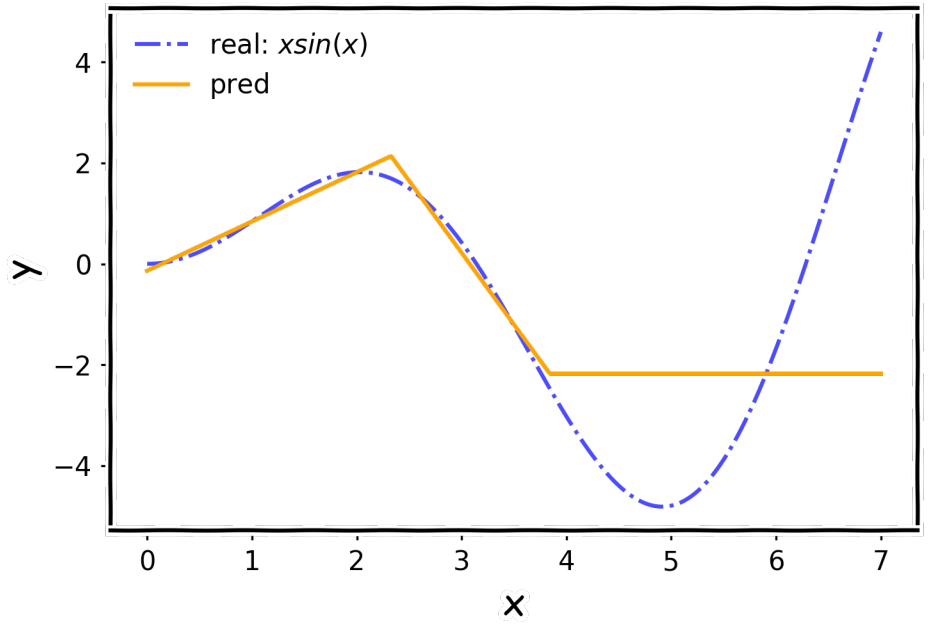
Number of nodes



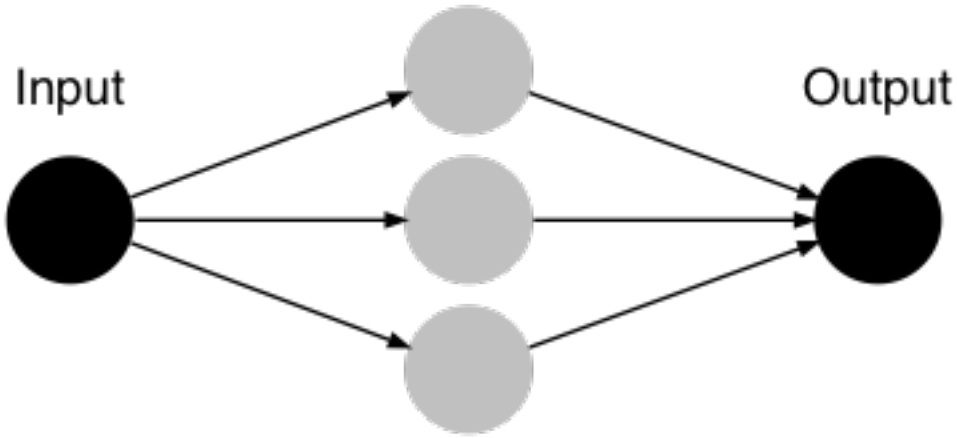


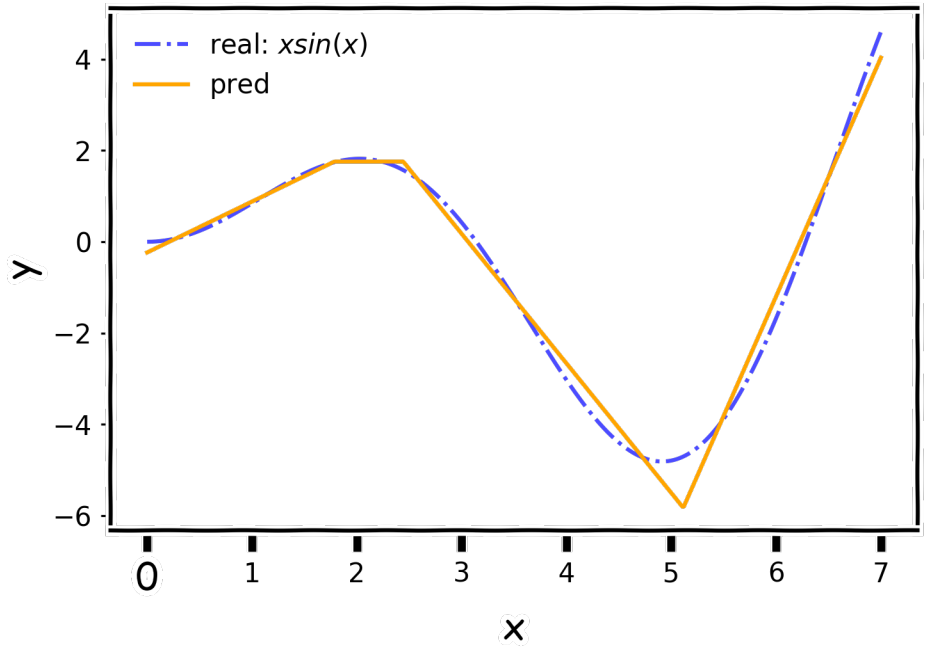
Number of nodes



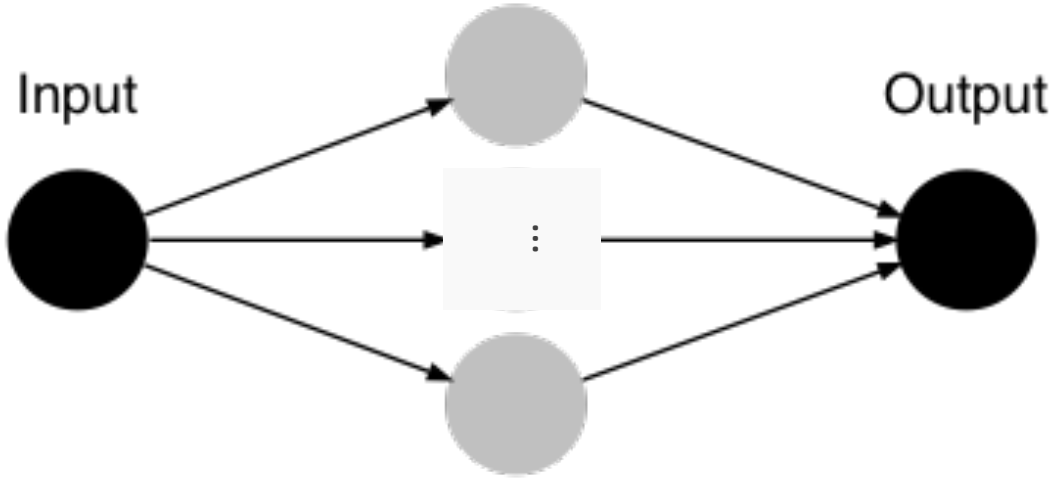


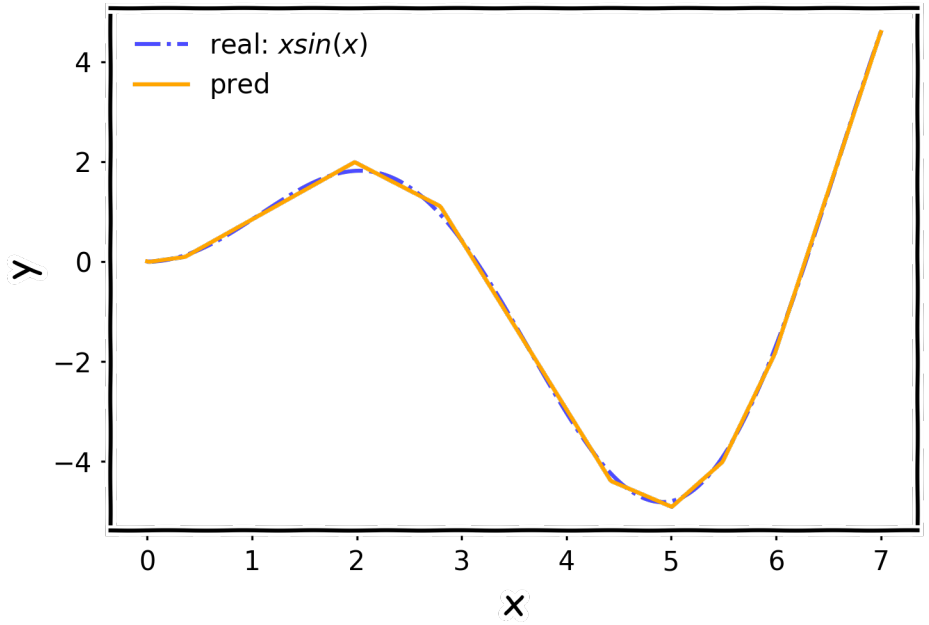
Number of nodes



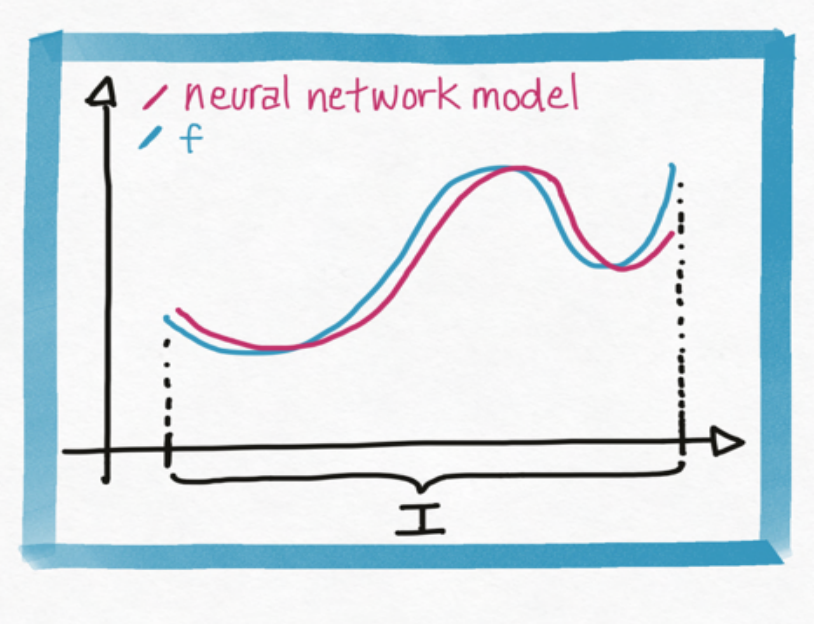


Number of nodes





Neural Networks as Universal Approximators

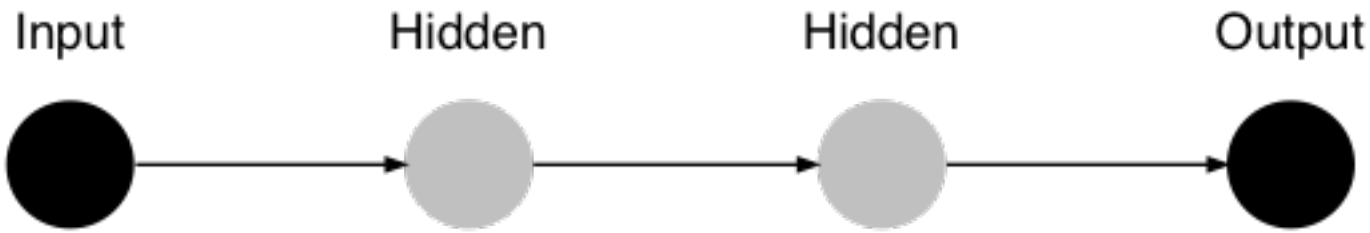


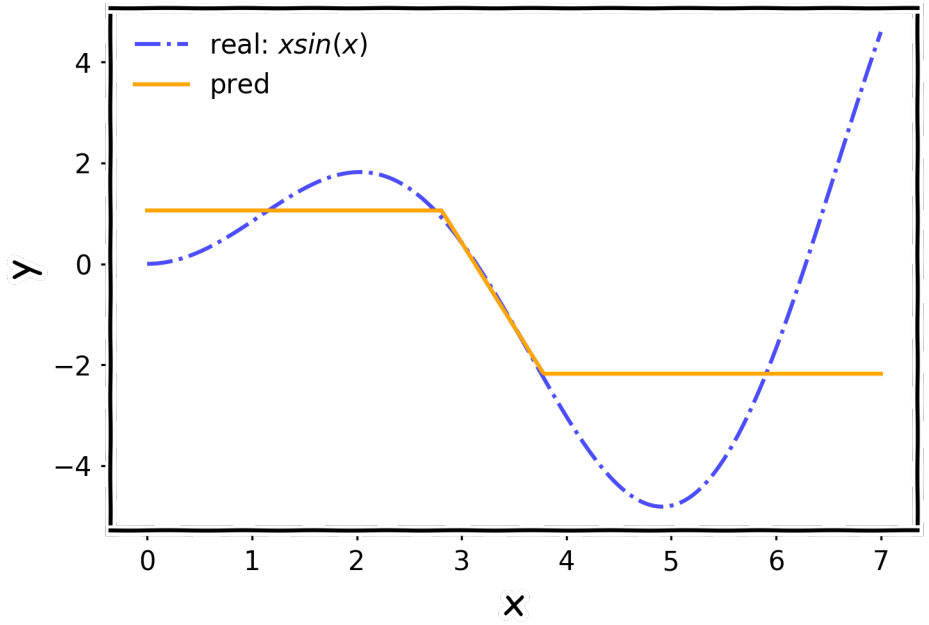
- We have seen that neural networks can represent complex functions, but are there limitations on what a neural network can express?
- **Theorem:**
- *For any continuous function f defined on a bounded domain, we can find a neural network that approximates f with an arbitrary degree of accuracy*.

One hidden layer is enough to represent an approximation of any function to an arbitrary degree of accuracy.

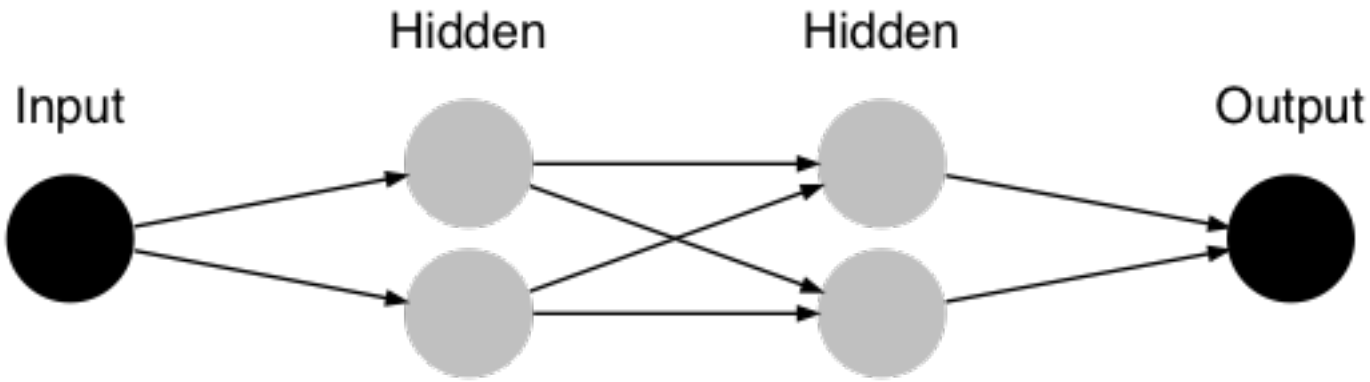
So why deeper?

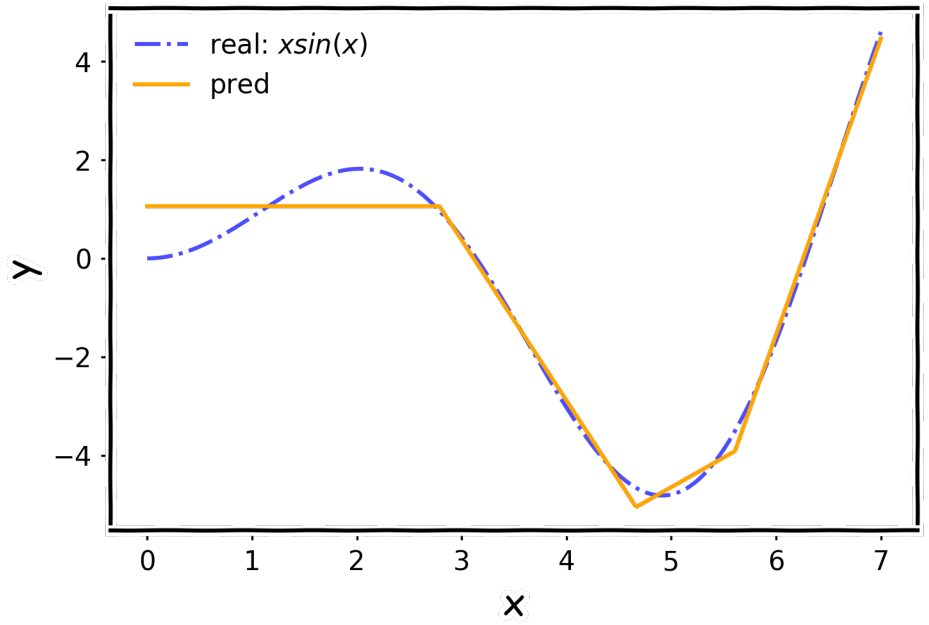
Layers


Input
Hidden
Hidden
Output

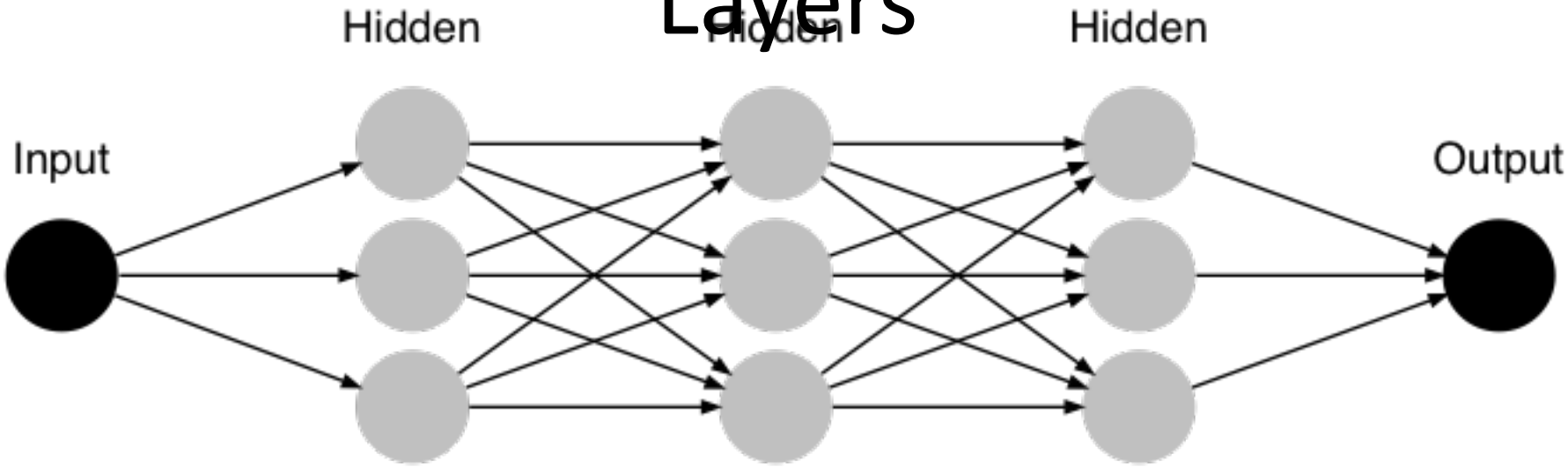

real: xsin(x)
pred
4
2
0
−2
−4
y
0
1
2
3
4
5
6
7
x

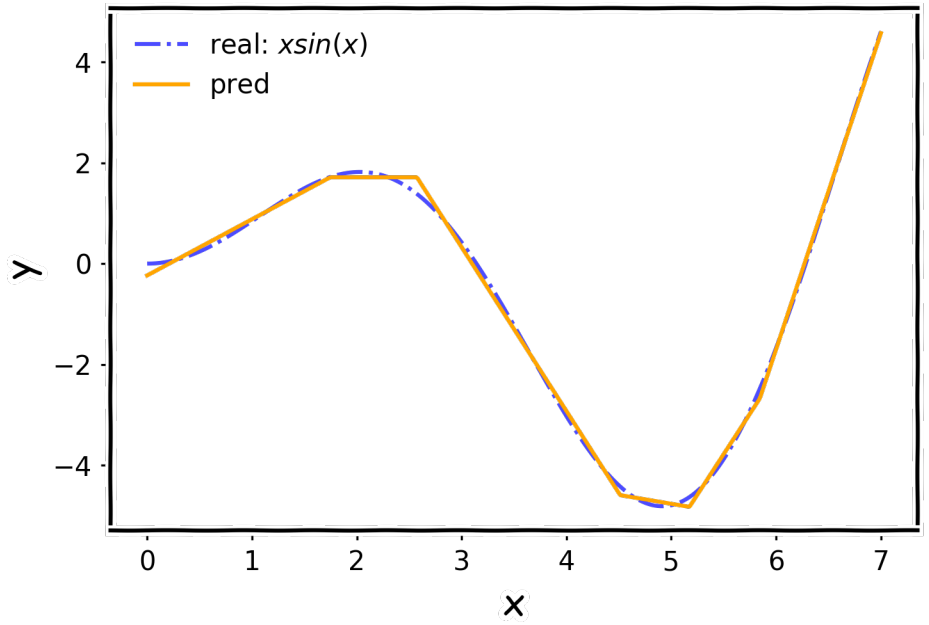
Layers


Input
Hidden
Hidden
Output


real: xsin(x)
pred
4
2
0
−2
−4
y
0
1
2
3
4
5
6
7
x

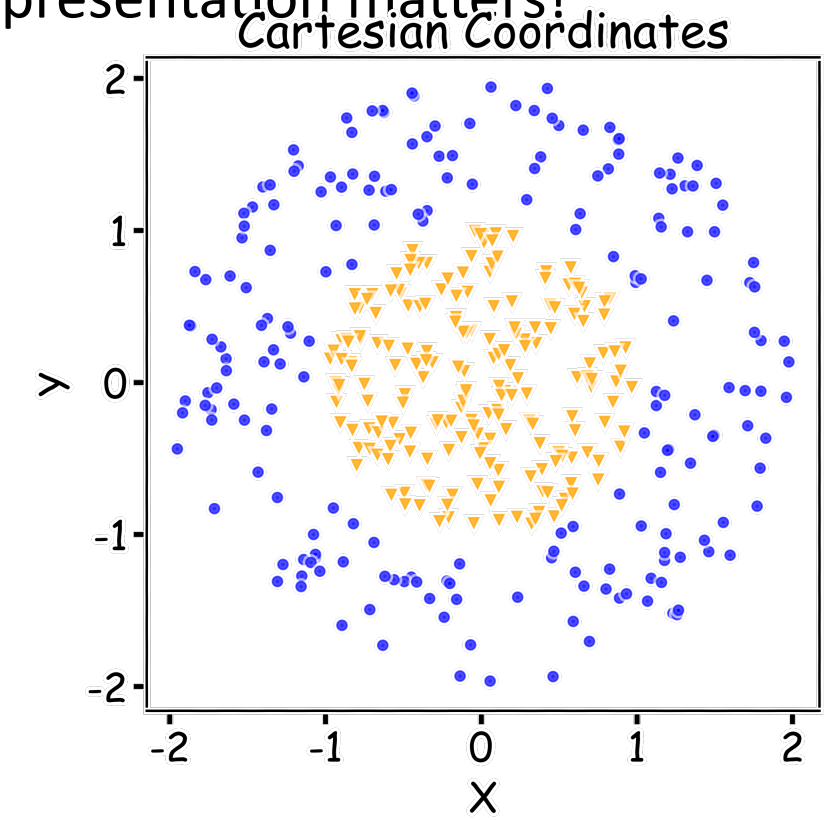
Layers

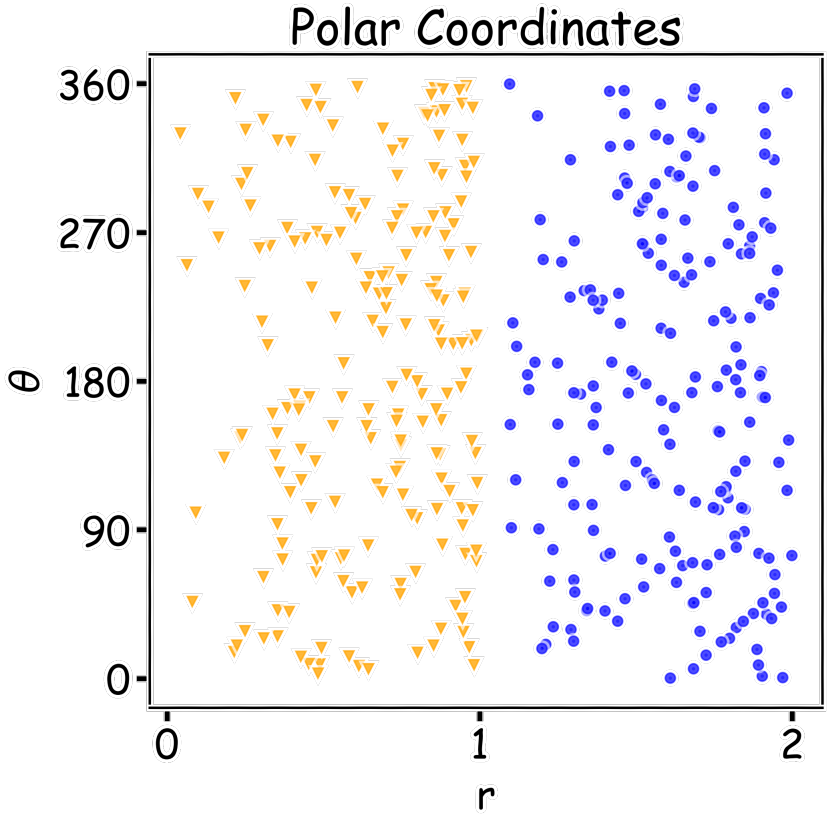

Hidden
Hidden
Hidden
Input
Output


real: xsin(x)
pred
4
2
0
−2
−4
y
0
1
2
3
4
5
6
7
x

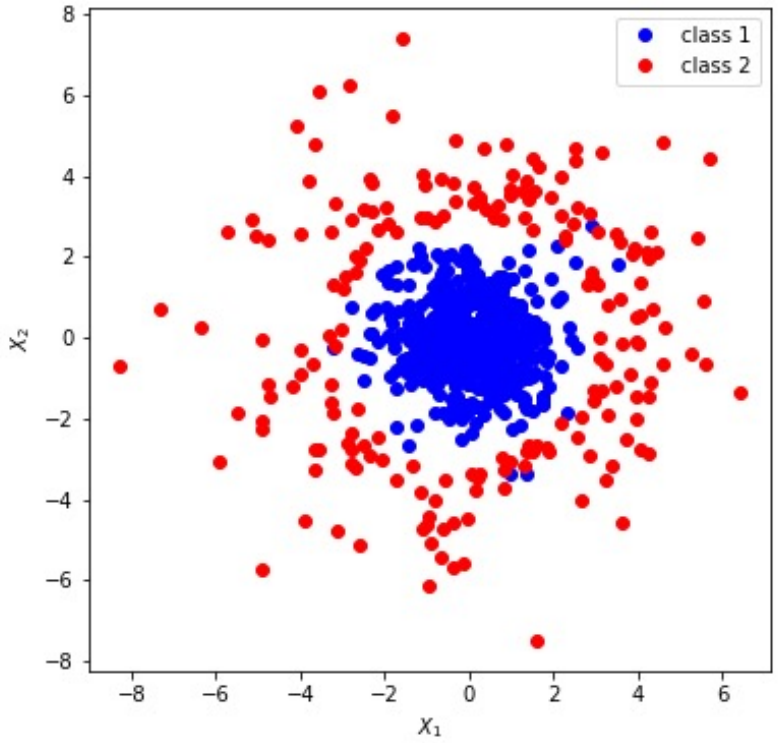
Why layers?

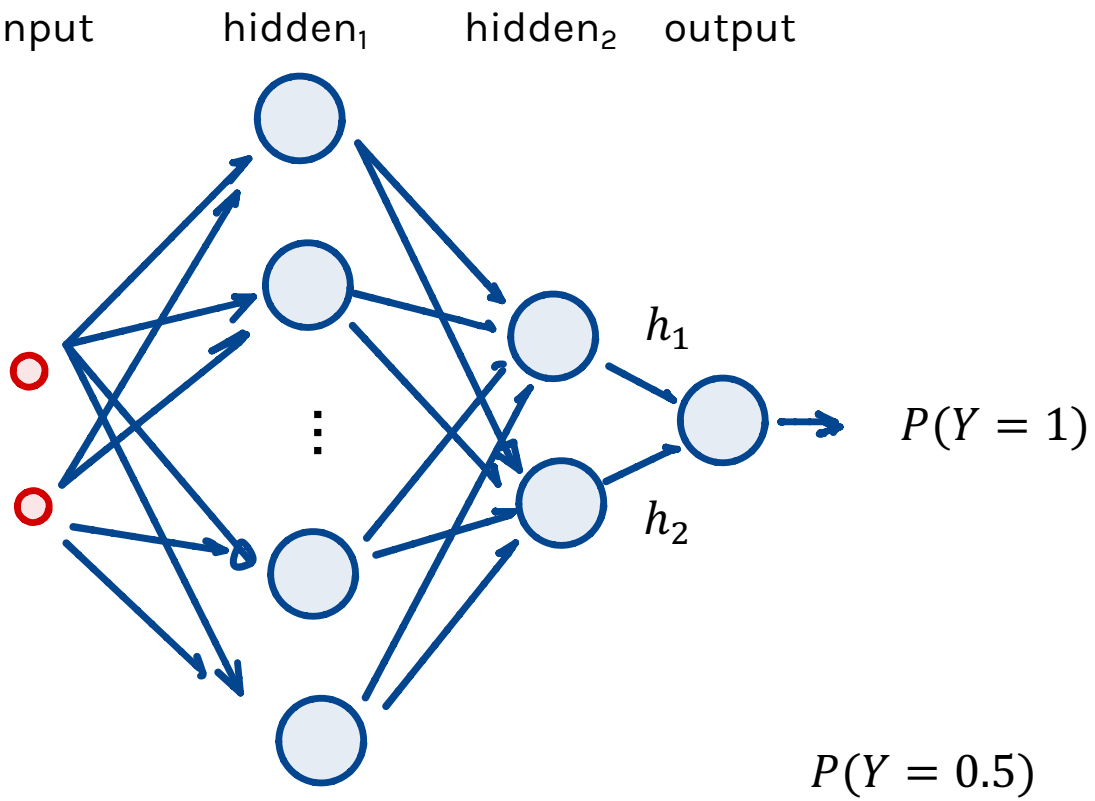
- Representation matters!

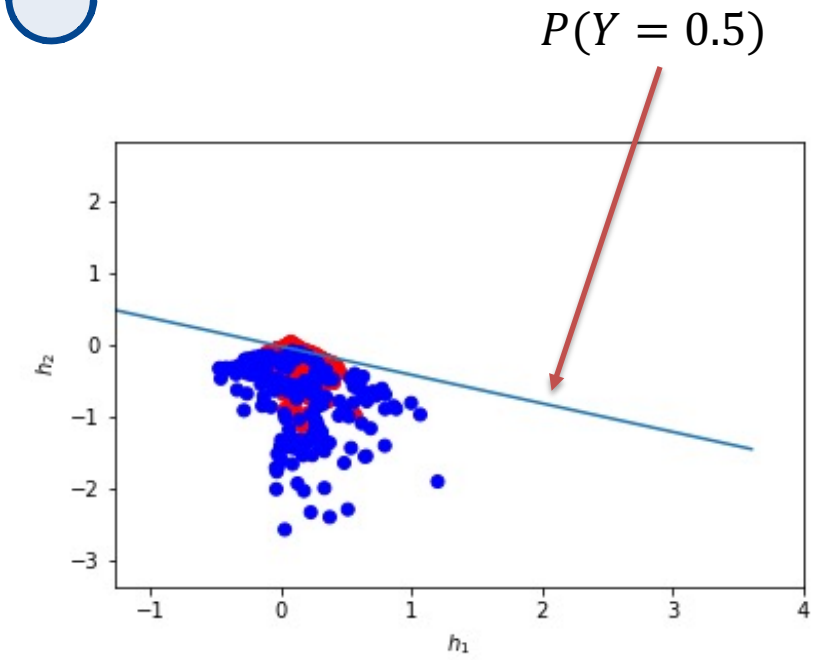




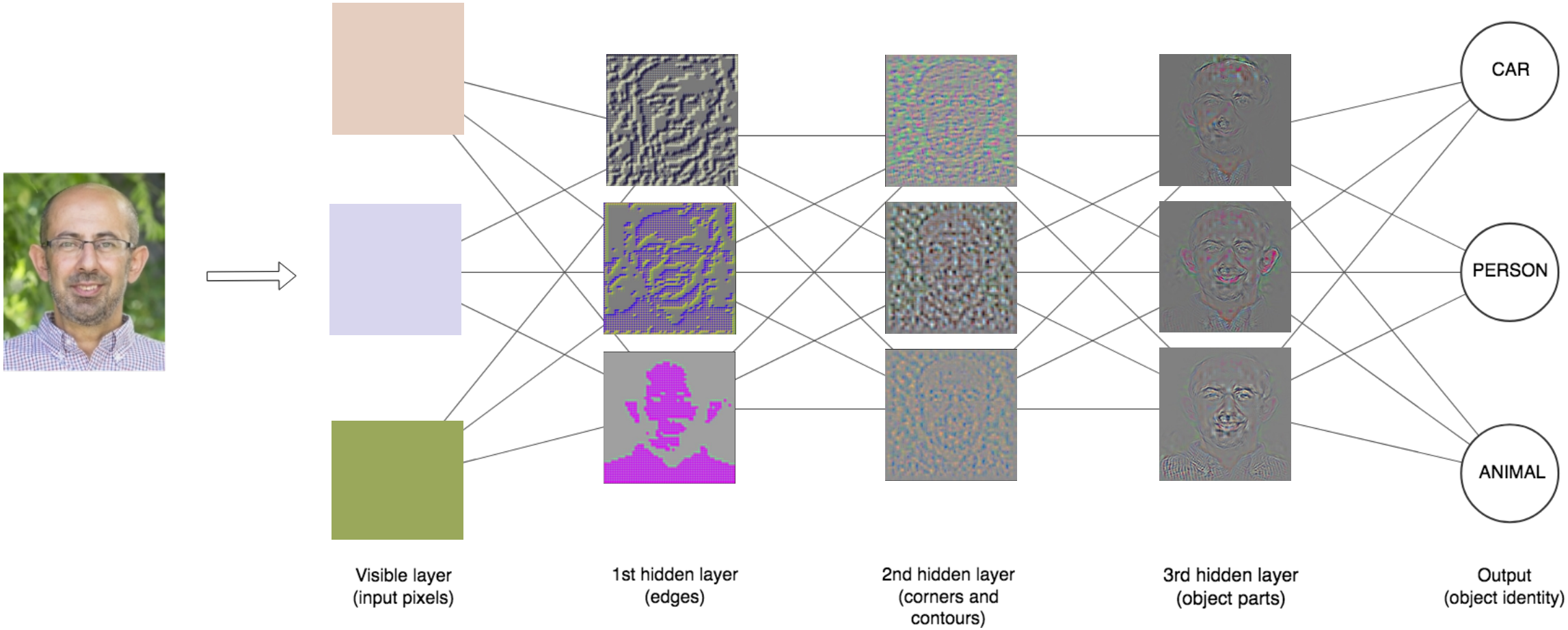
Neural networks can **learn useful representations** for the problem. This is another reason why they can be so powerful!


class 1
class 2
X_1
X_2


input
hidden$_1$
hidden$_2$
output
h_1
h_2
$P(Y = 1)$


$P(Y = 0.5)$
h_1
h_2

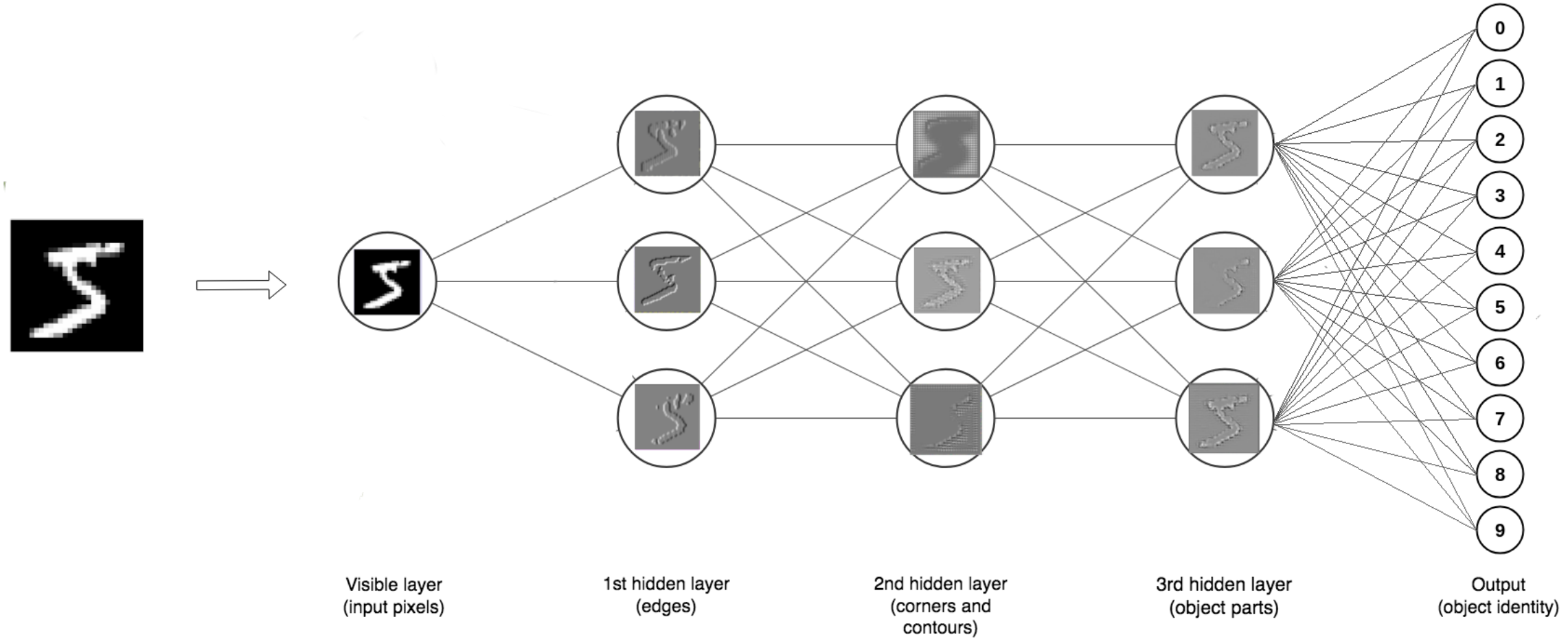
Depth = Repeated Compositions



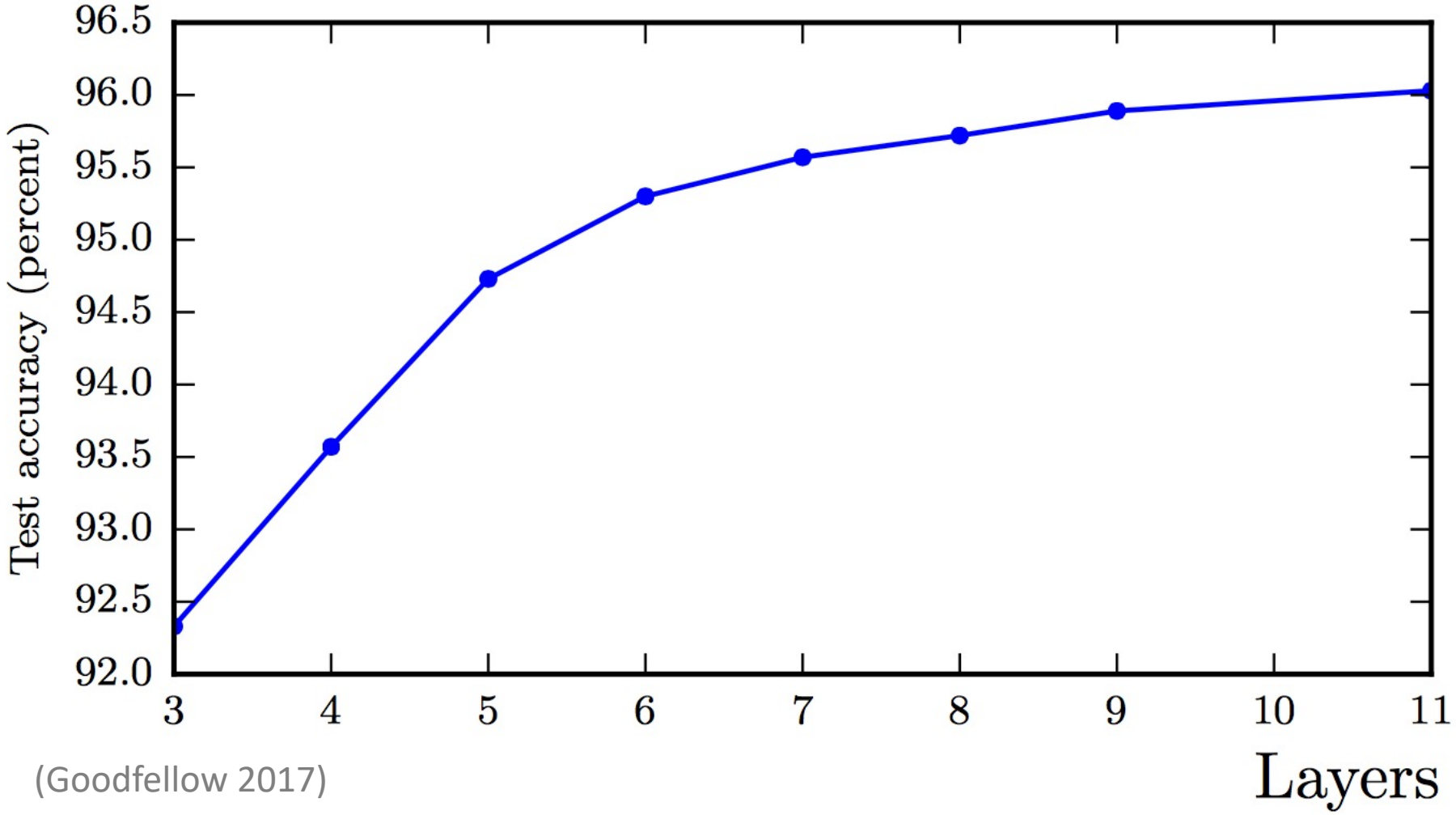
Neural Networks

- Hand-written digit recognition: MNIST data

Depth = Repeated Compositions



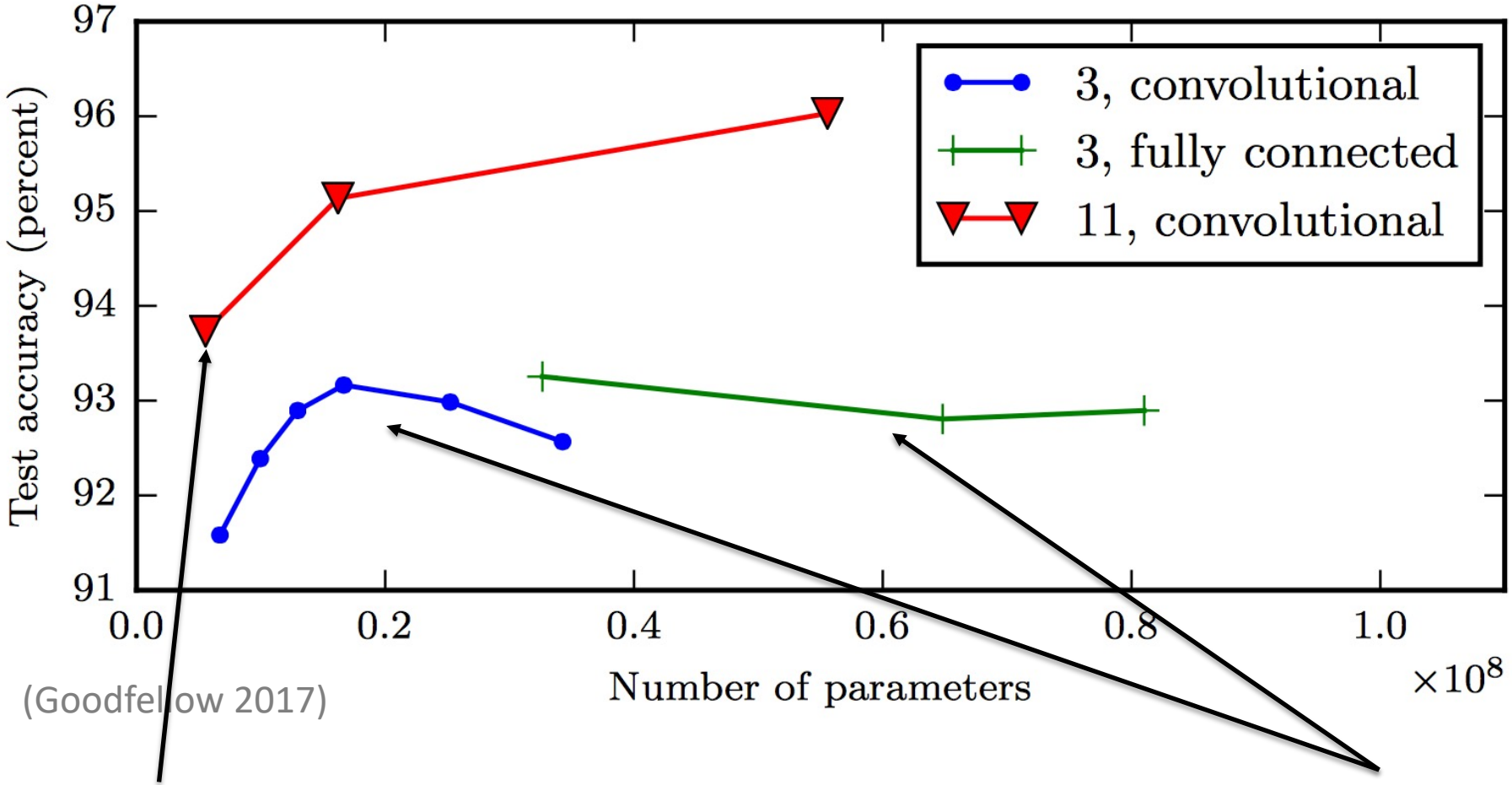
Better Generalization with Depth



(Goodfellow 2017)

Shallow Nets Overfit More

Depth helps, and it's not just because of more parameters



(Goodfellow 2017)

Don't worry about this word "convolutional". It's just a special type of neural network, often used for images.

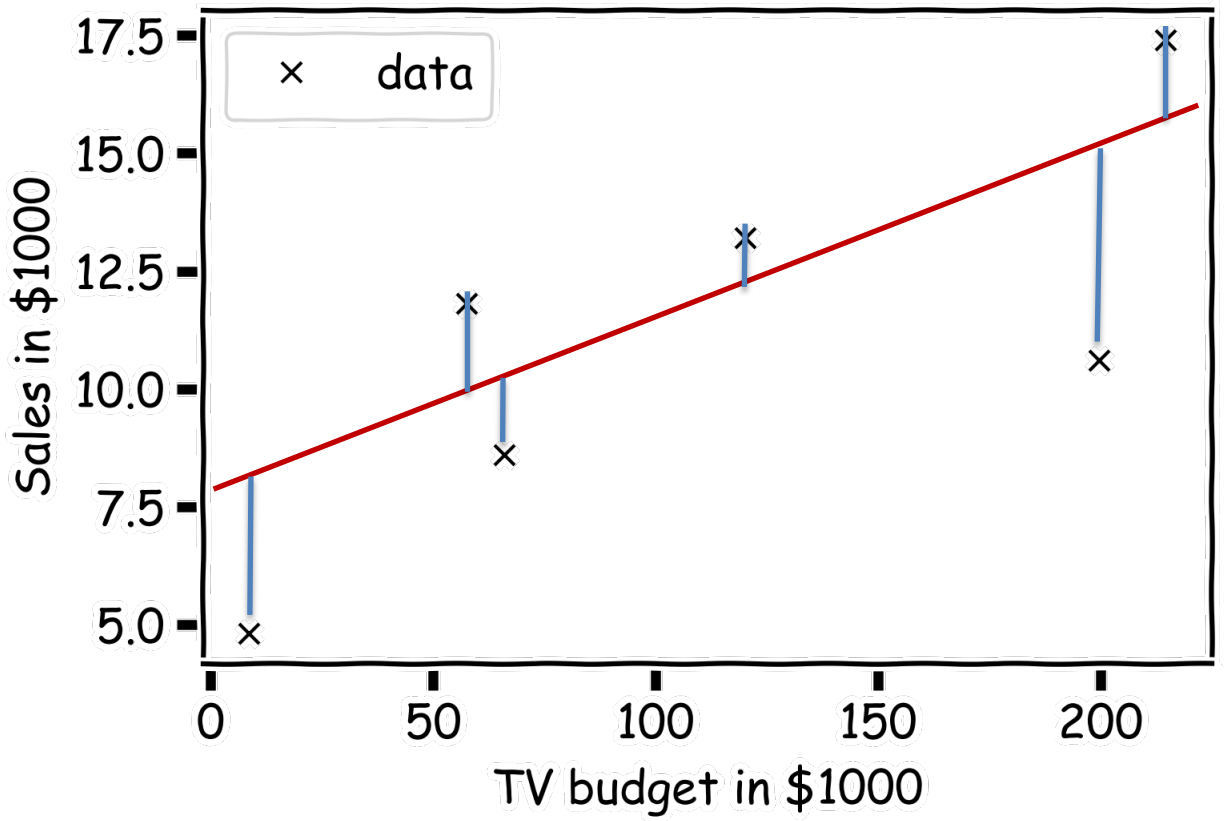
The **11-layer net** generalizes better on the test set when controlling for number of parameters.

The 3-layer nets perform worse on the test set, even with similar number of total parameters.

Estimate of the regression coefficients (cont)

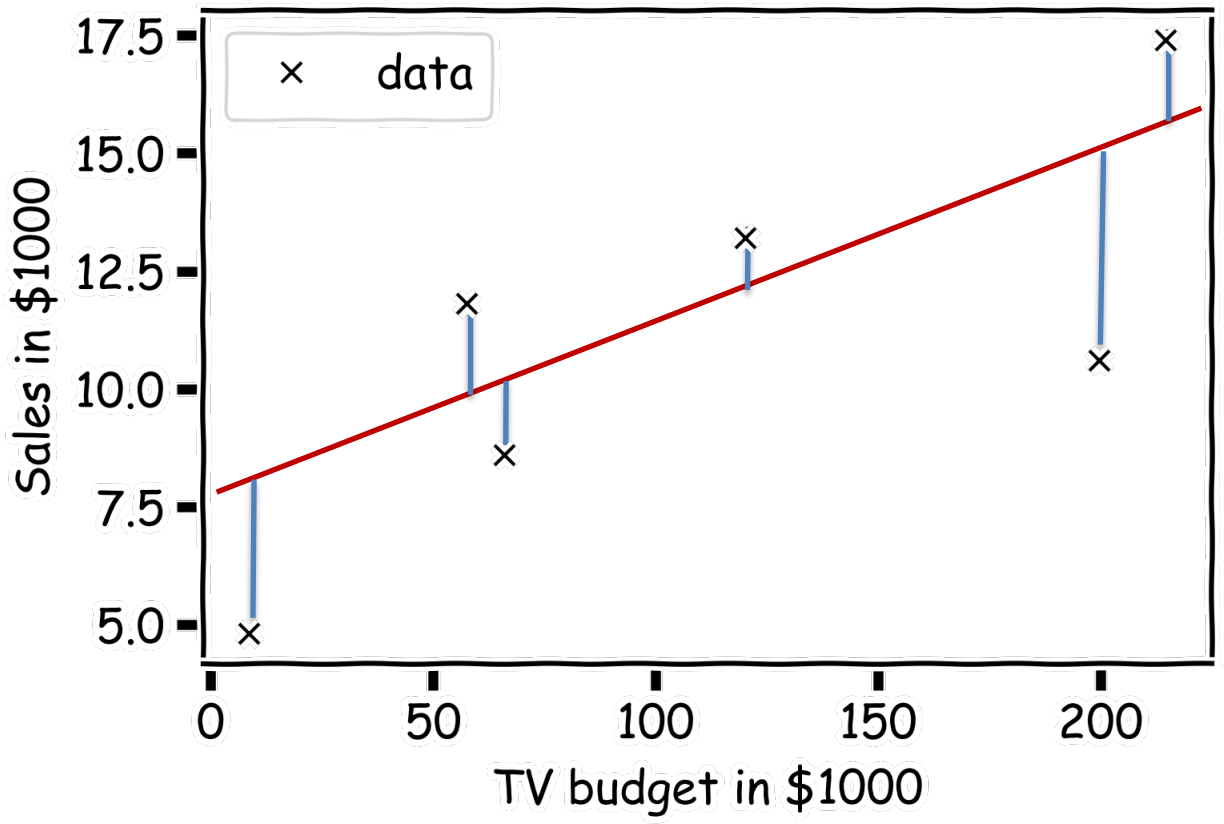
Question: Which line is the best?

For each observation (x_n, y_n), the absolute residual is calculate the residuals $r_i = |y_i - \hat{y}_i|$.



Loss Function: Aggregate Residuals

- How do we aggregate residuals across the entire dateset?



1. Max Absolute Error
2. Mean Absolute Error
3. Mean Squared Error

Estimate of the regression coefficients (cont)

- Again we use MSE as our **loss function**,

$$L(\beta_0, \beta_1) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2 = \frac{1}{n}\sum_{i=1}^{n}[y_i - (\beta_1 X + \beta_0)]^2 .$$

- We choose $\hat{\beta}_1$ and $\hat{\beta}_0$ in order to minimize the predictive errors made by our model, i.e. minimize our loss function.
- Then the optimal values for $\hat{\beta}_0$ and $\hat{\beta}_1$ should be:

$$\widehat{\beta}_0, \widehat{\beta}_1 = \underset{\beta_0, \beta_1}{\operatorname{argmin}} L(\beta_0, \beta_1).$$

WE CALL THIS **FITTING** OR **TRAINING** THE MODEL