27: Intro to Deep Learning + Wrap-up

Lisa Yan

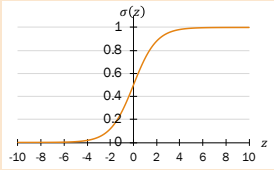
June 8, 2020

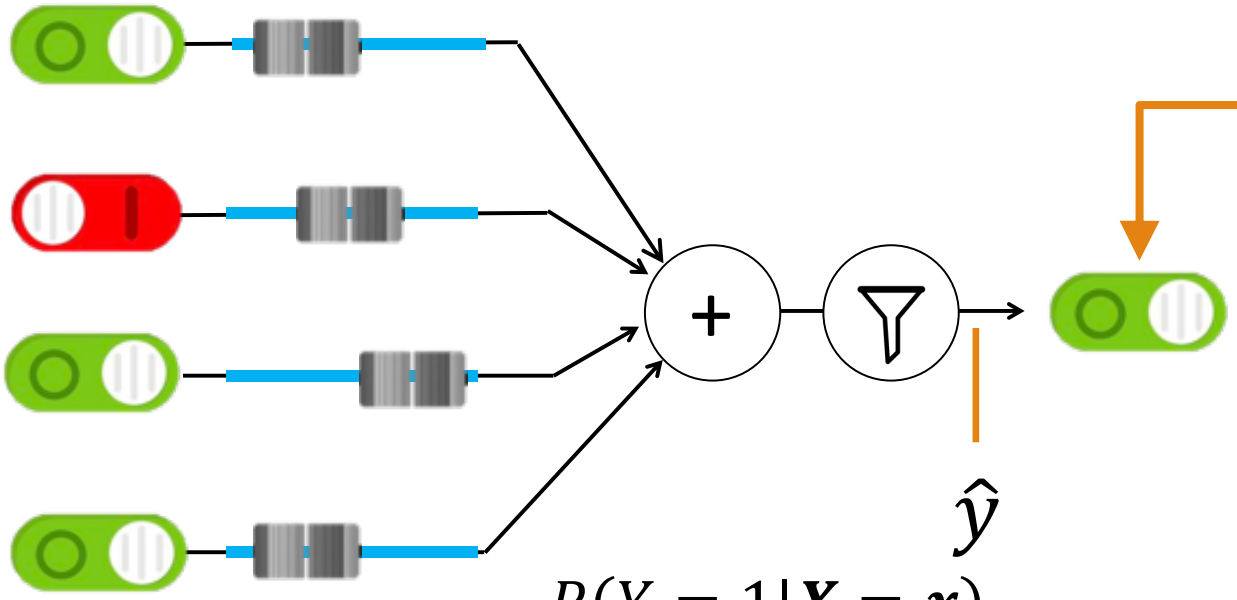
Quick slide reference

Deep Learning

Logistic Regression Model

$\boldsymbol{X}$ → $\theta_0 + \sum_{j=1}^{m} \theta_j X_j$ → sigmoid function $\sigma(z) = \frac{1}{1+e^{-z}}$ → $P(Y = 1|\boldsymbol{X})$





$\boldsymbol{x}$

$\hat{y}$

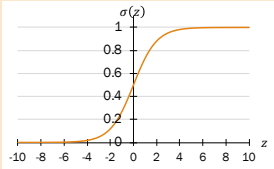
$P(Y = 1|\boldsymbol{X} = \boldsymbol{x})$

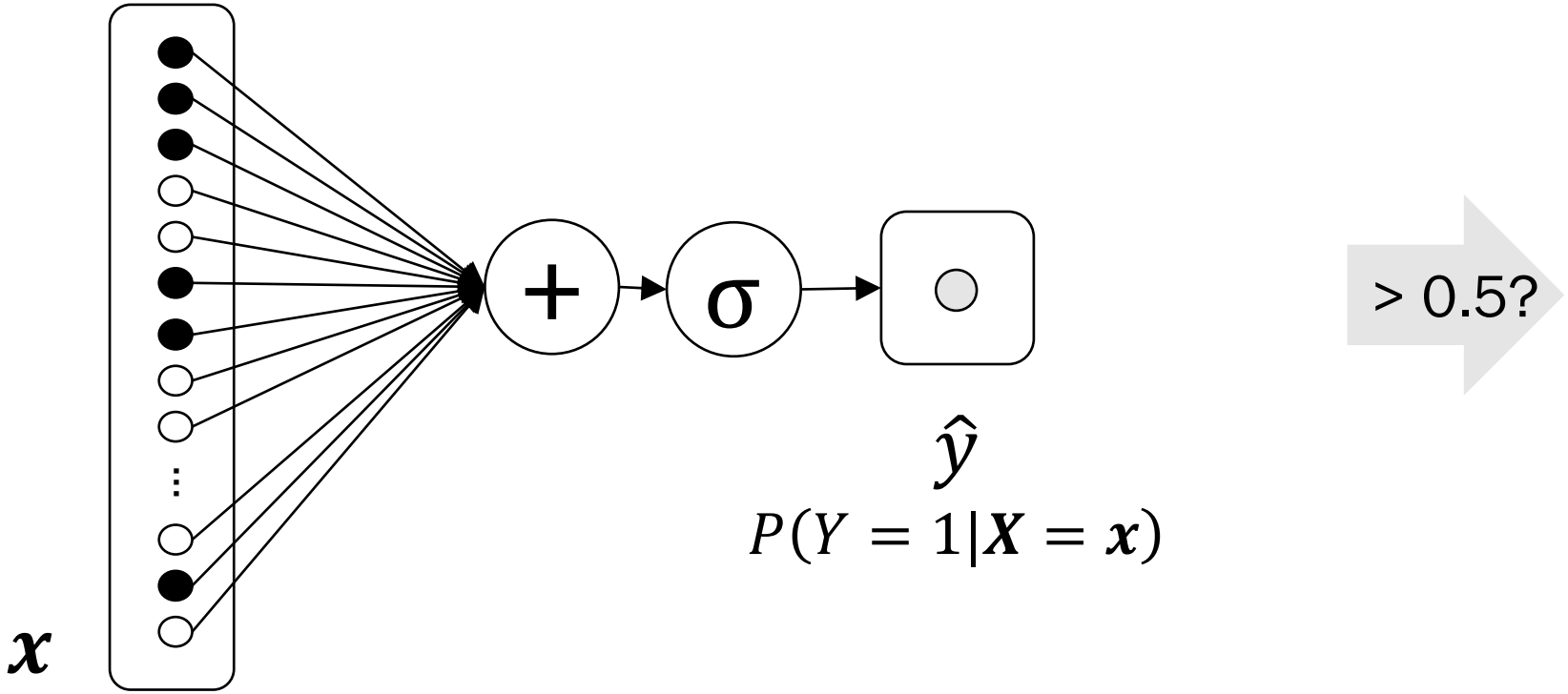
$$\underset{y=\{0,1\}}{\arg\max}\, P(Y \mid \boldsymbol{X})$$

Not actually part of modeling
only part of prediction
(so ignore for now)

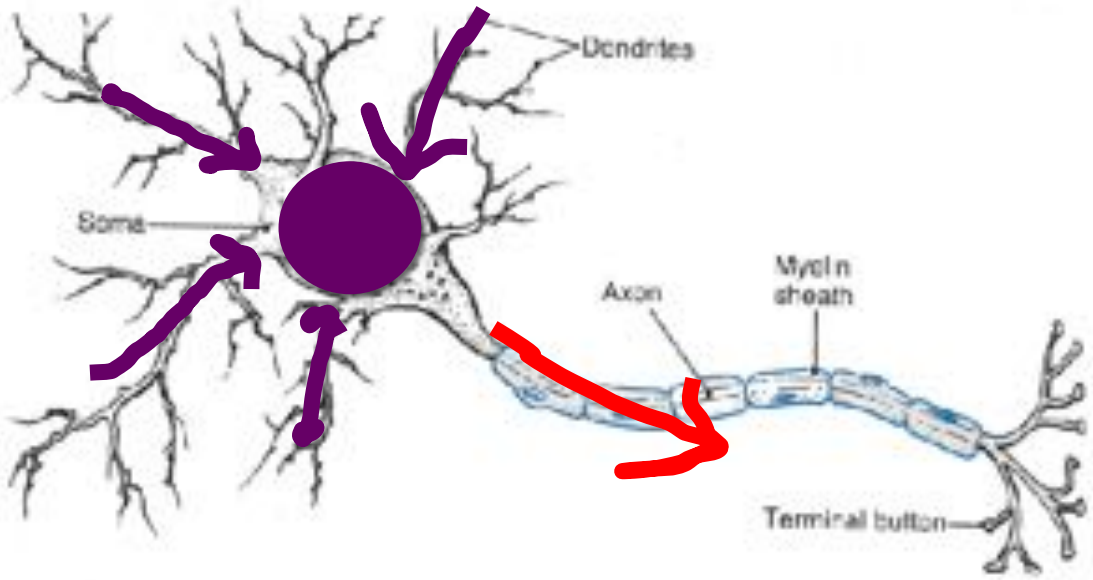
Big ideas from Logistic Regression

$\boldsymbol{X}$ → $\theta_0 + \sum_{j=1}^{m} \theta_j X_j$ → sigmoid function $\sigma(z) = \frac{1}{1+e^{-z}}$ → $\hat{P}(Y = 1|\boldsymbol{X})$

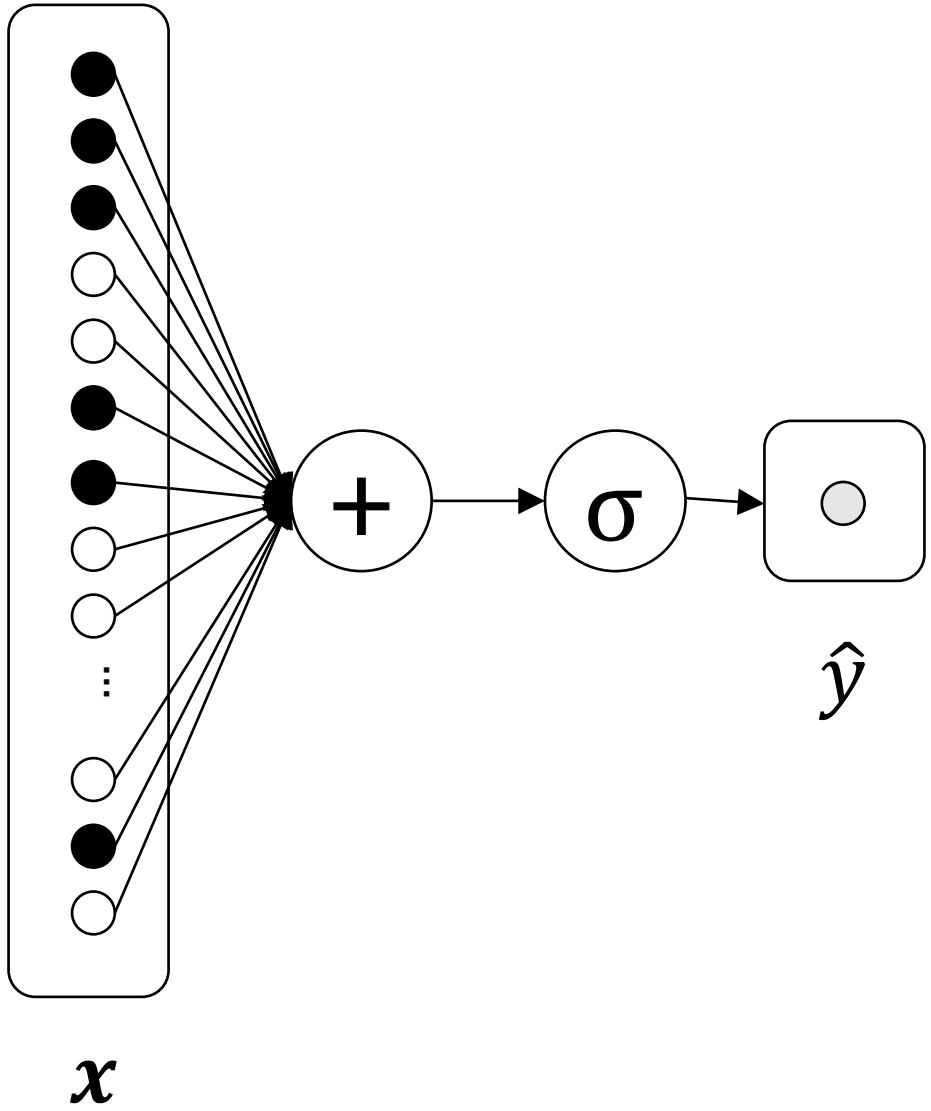




One neuron = One logistic regression

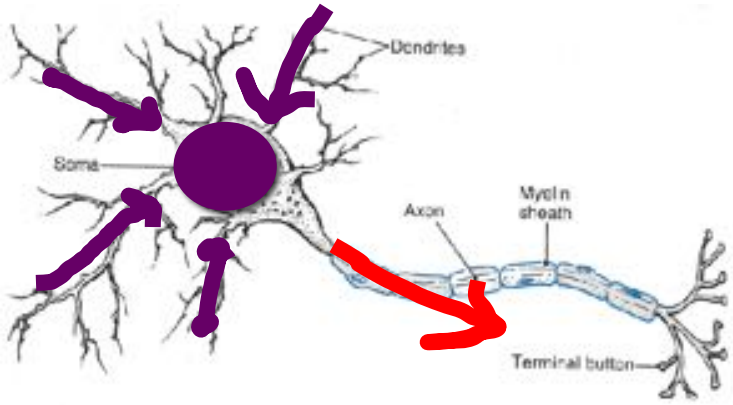


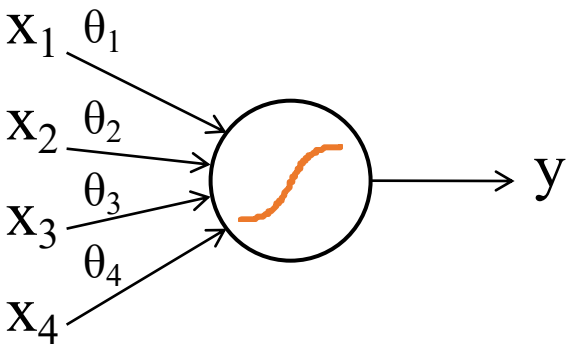
=



Biological basis for neural networks

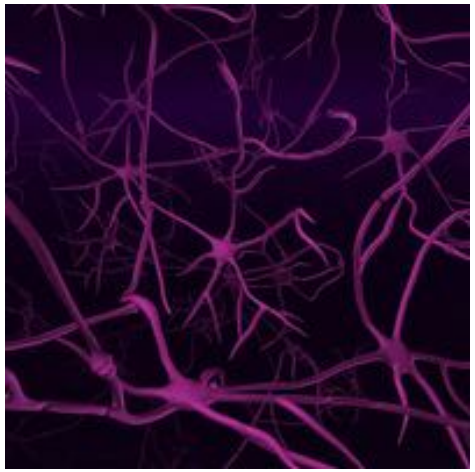
A neuron



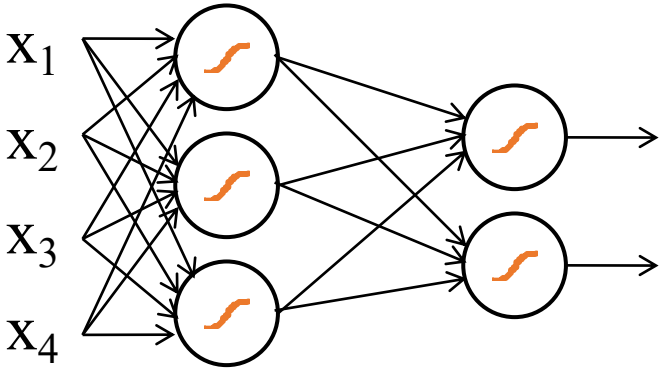


One neuron = one logistic regression

Your brain



(actually, probably someone else's brain)



Neural network = many logistic regressions

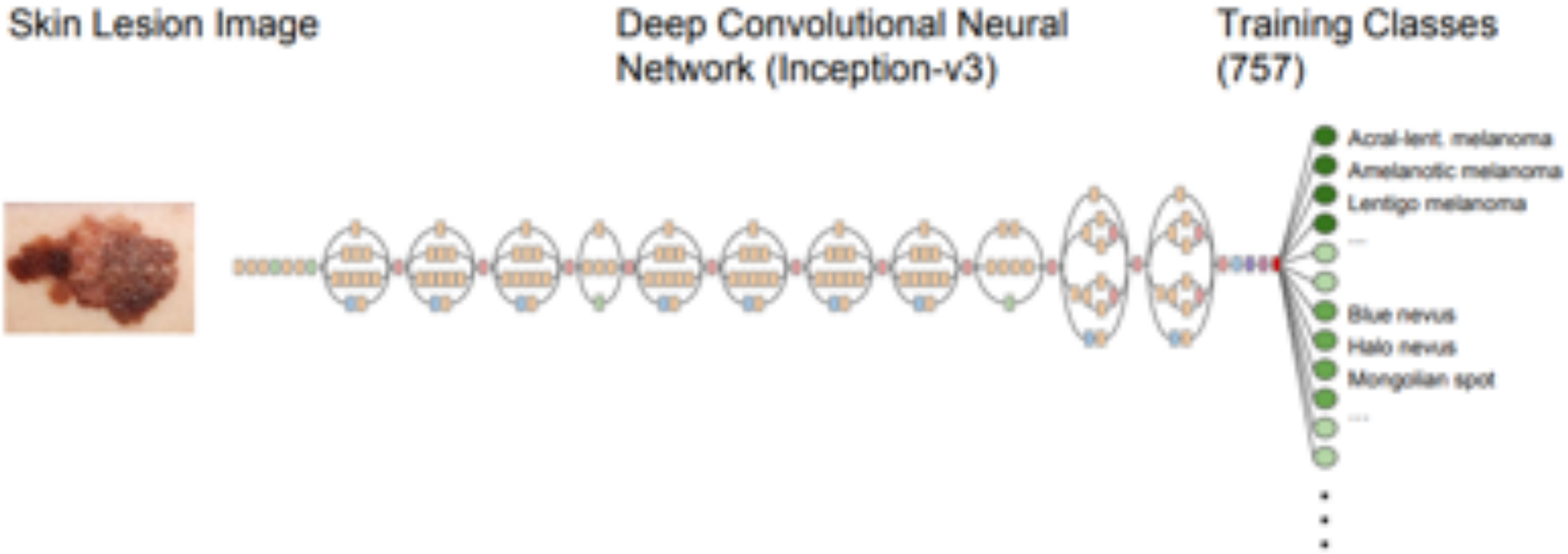
Innovations in deep learning

AlphaGO (2016)

Deep learning (neural networks) is the core idea behind the current revolution in AI.

Computers making art

Detecting skin cancer



Esteva, Andre, et al. "Dermatologist-level classification of skin cancer with deep neural networks." *Nature* 542.7639 (2017): 115-118.

Deep learning

<u>def</u> **Deep learning** is
maximum likelihood estimation with neural networks.

<u>def</u> A **neural network** is
(at its core) many logistic regression pieces stacked on top of each other.

$[1,0,\ldots,1]$
$\boldsymbol{x}$, input

LOL

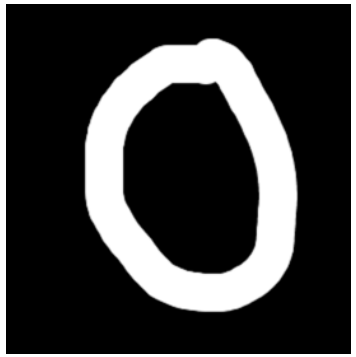
Lots of Logistic
(regressions)

$\hat{y}$, output
$P(Y|\boldsymbol{X}=\boldsymbol{x})$

> 0.5?

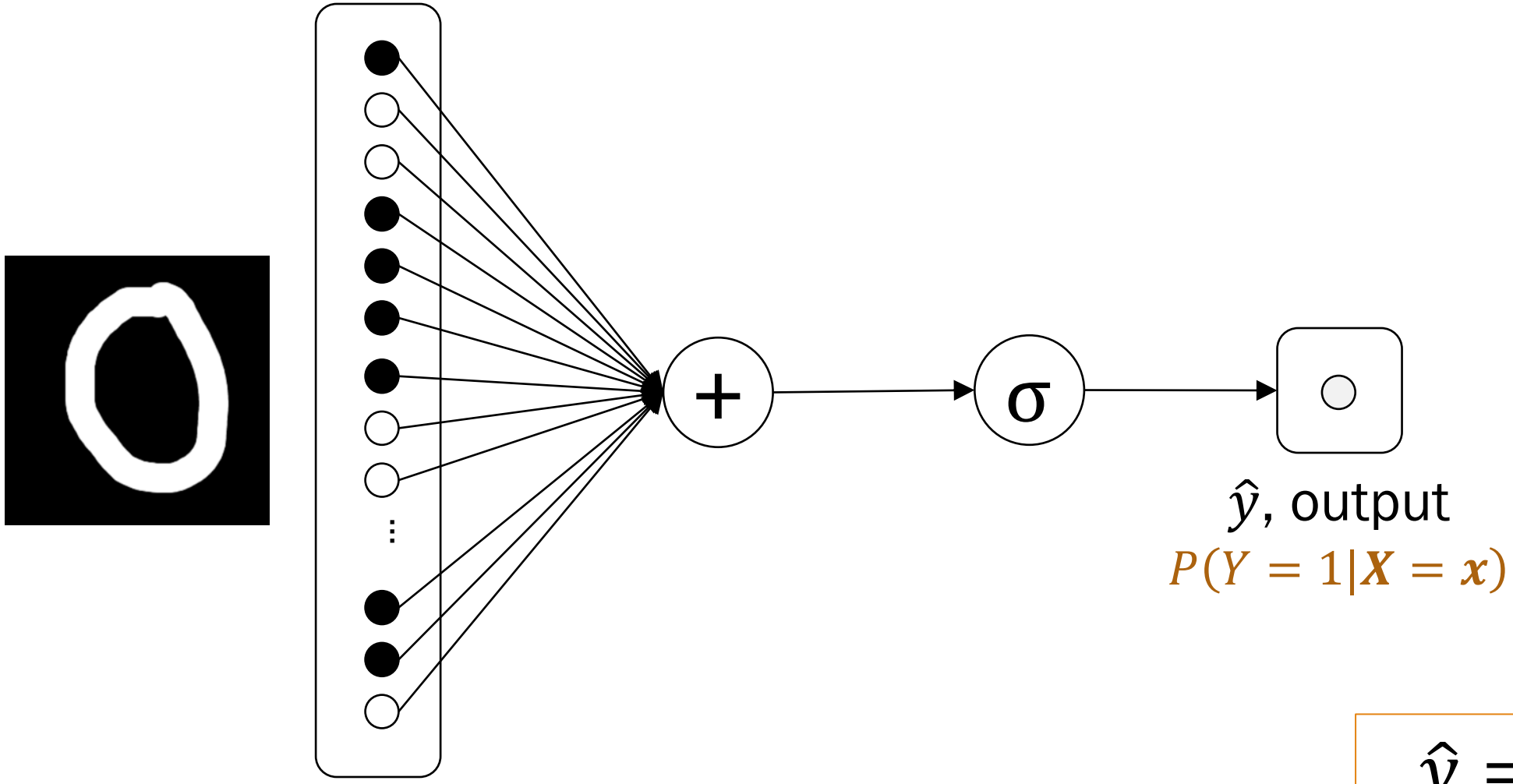
Yes.
Predict 1

Digit recognition example

Input image	Input feature vector	Output label
	$\boldsymbol{x}^{(i)} = [0,0,0,0, \dots, 1,0,0,1, \dots, 0,0,1,0]$	$y^{(i)} = 0$
	$\boldsymbol{x}^{(i)} = [0,0,1,1, \dots, 0,1,1,0, \dots, 0,1,0,0]$	$y^{(i)} = 1$

We make feature vectors from (digitized) pictures of numbers.

Logistic Regression



$\hat{y}$, output

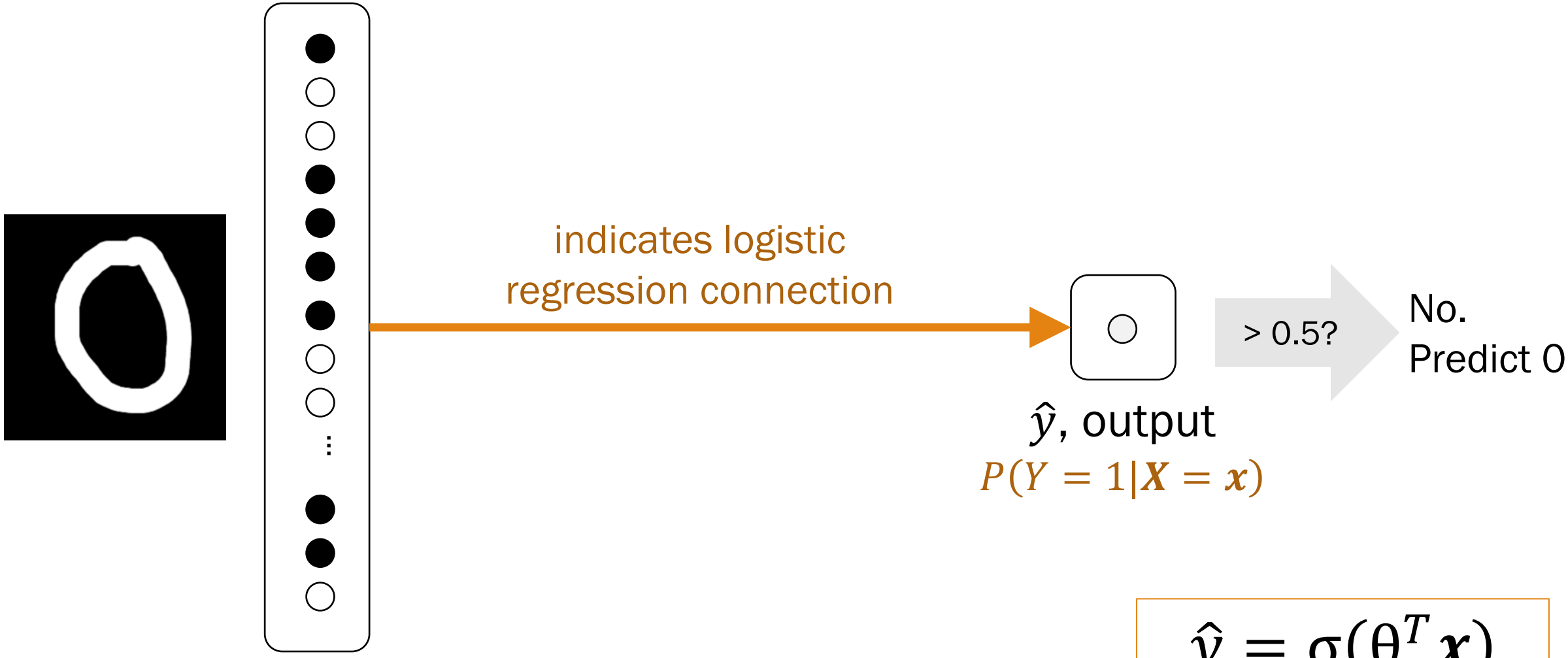
$P(Y = 1 | \boldsymbol{X} = \boldsymbol{x})$

$\boldsymbol{x}$, input features

(pixels, on/off)

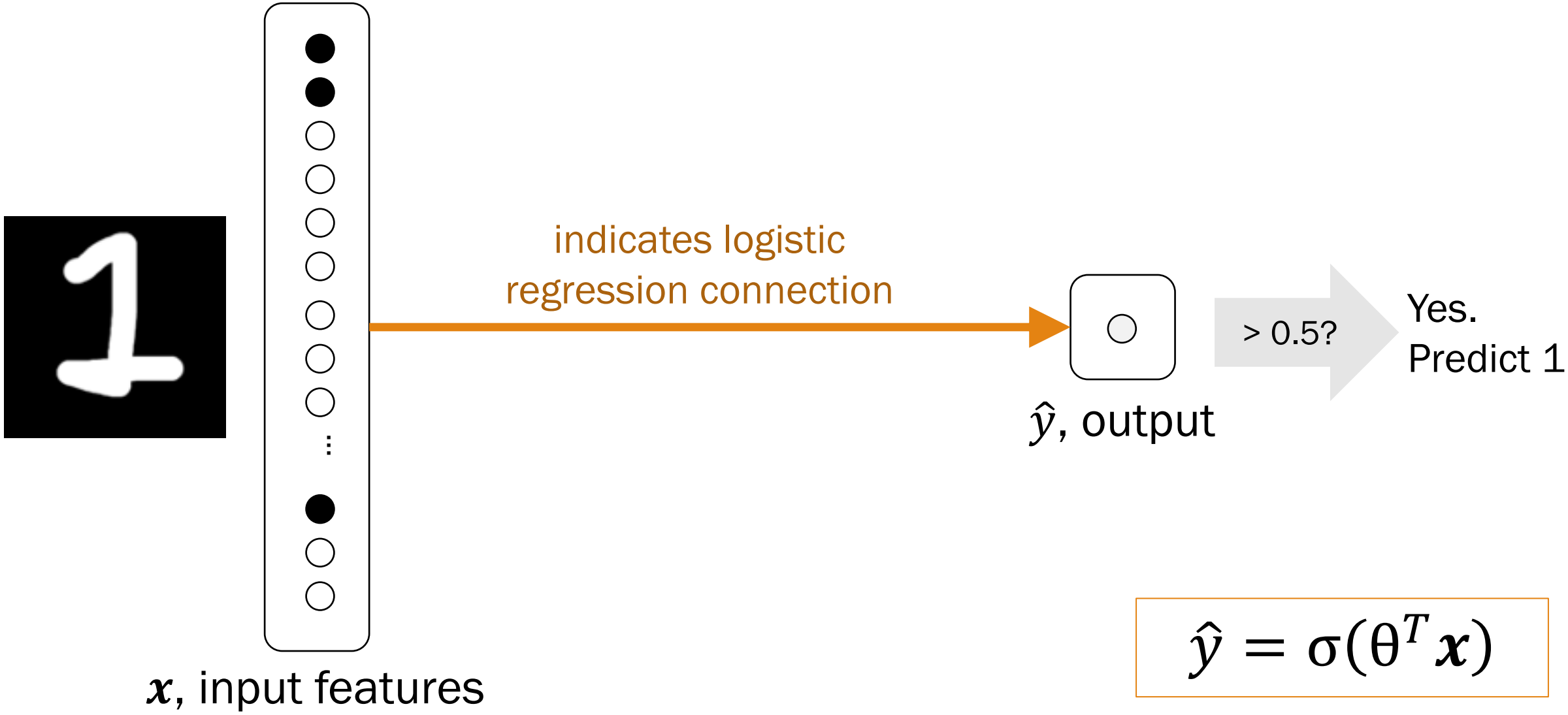
$$\hat{y} = \sigma(\theta^T \boldsymbol{x})$$

Logistic Regression



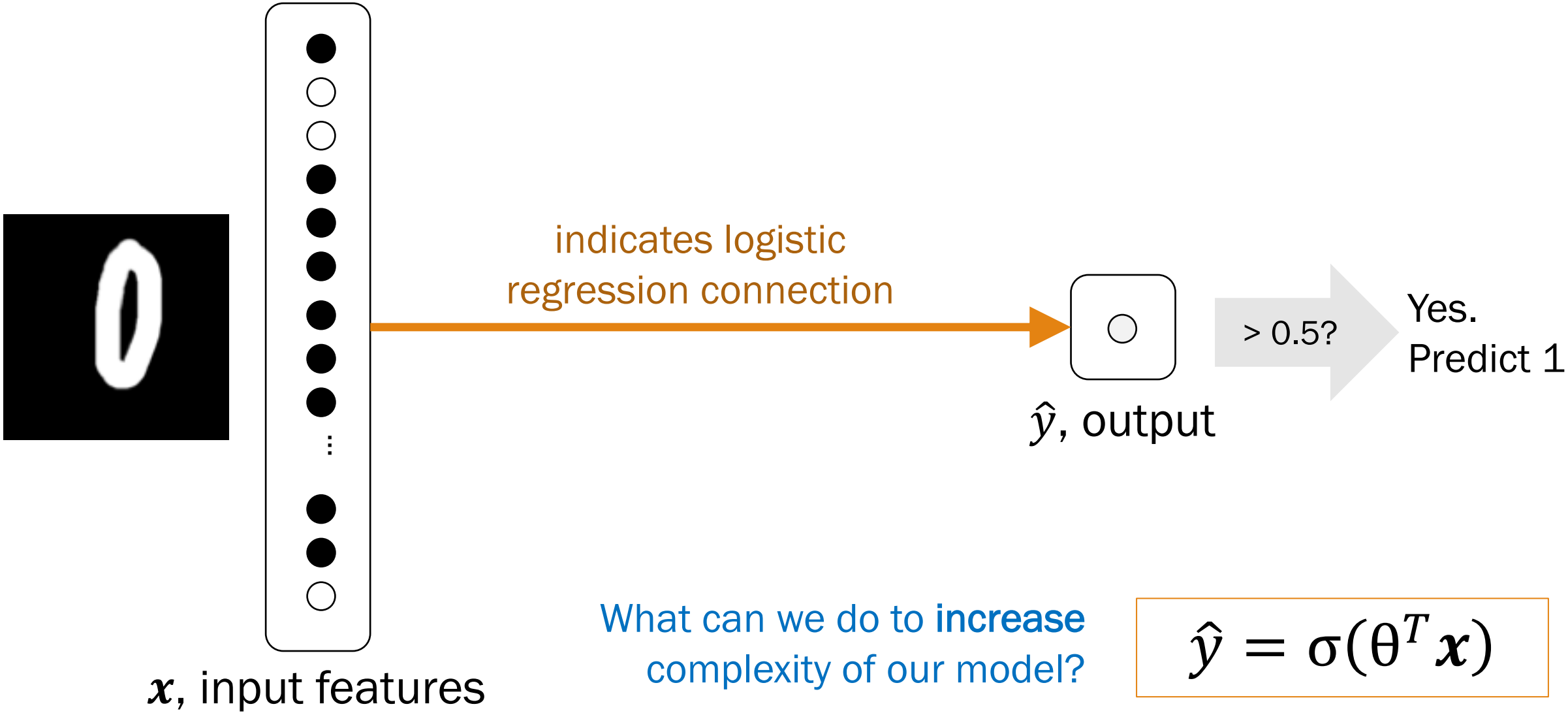
$$\hat{y} = \sigma(\theta^T \boldsymbol{x})$$

Logistic Regression



$$\hat{y} = \sigma(\theta^T \boldsymbol{x})$$

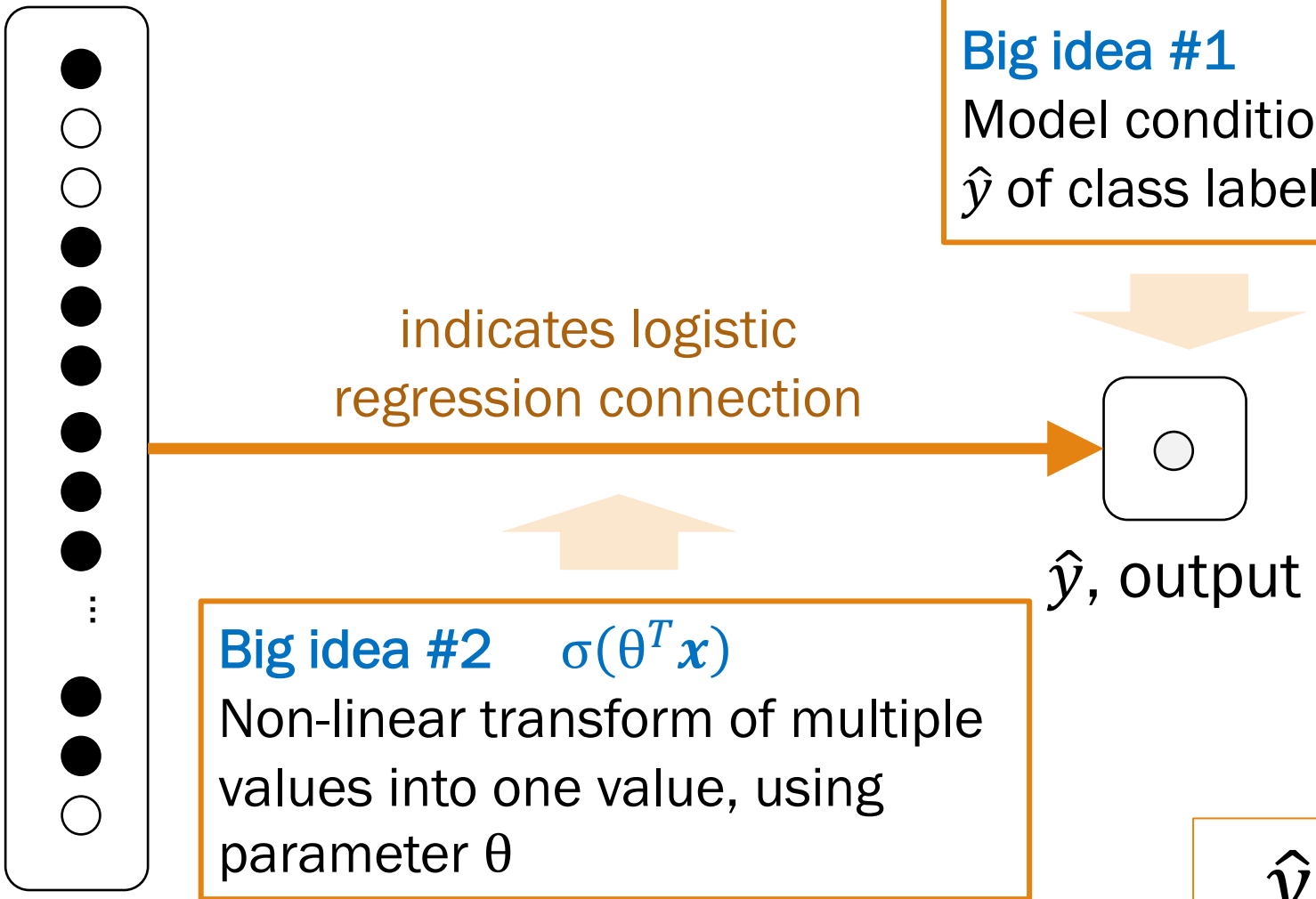
Logistic Regression



What can we do to **increase** complexity of our model?

$$\hat{y} = \sigma(\theta^T \boldsymbol{x})$$

Big ideas from logistic regression



$\boldsymbol{x}$, input features

Big idea #1 $P(Y|\boldsymbol{X} = \boldsymbol{x})$

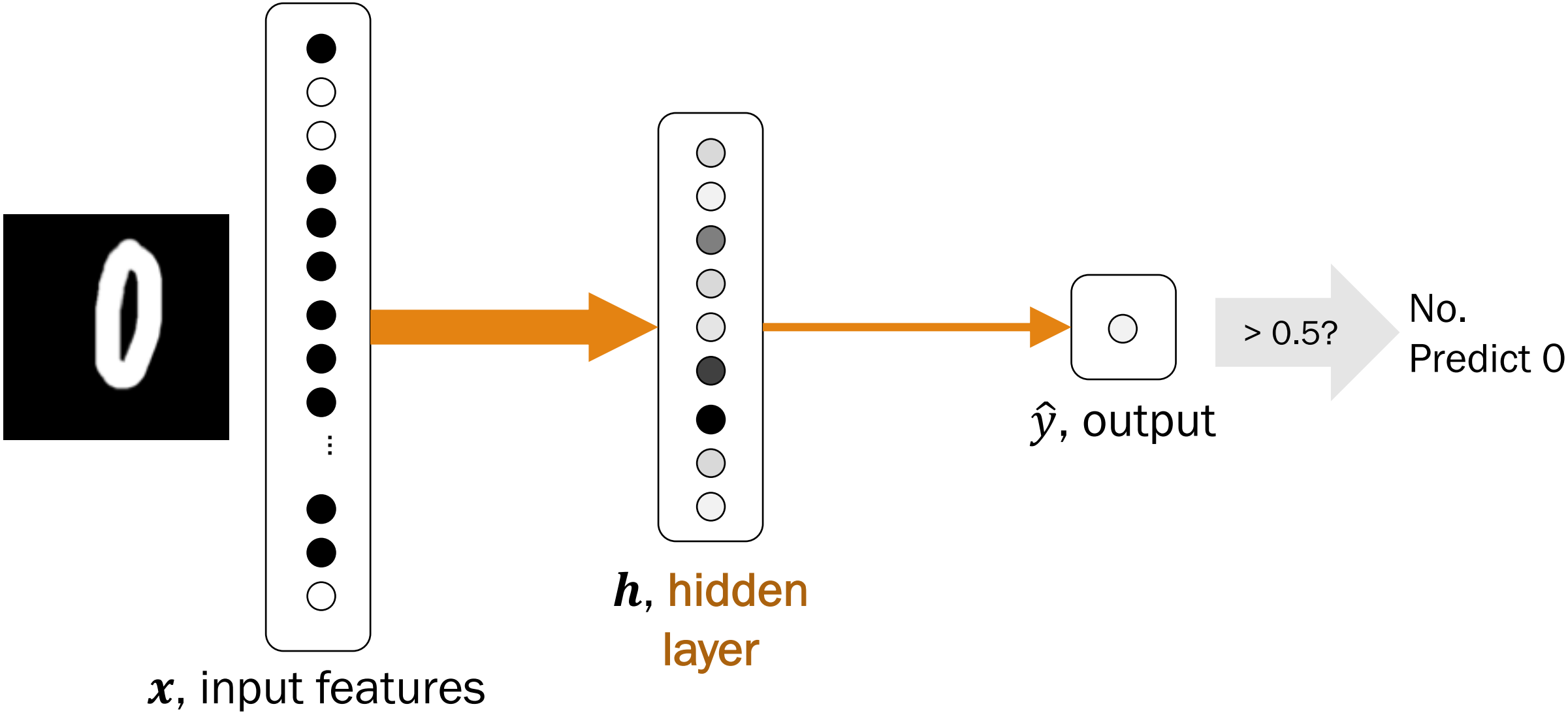
Model conditional probability $\hat{y}$ of class label given input

Big idea #2 $\sigma(\theta^T \boldsymbol{x})$

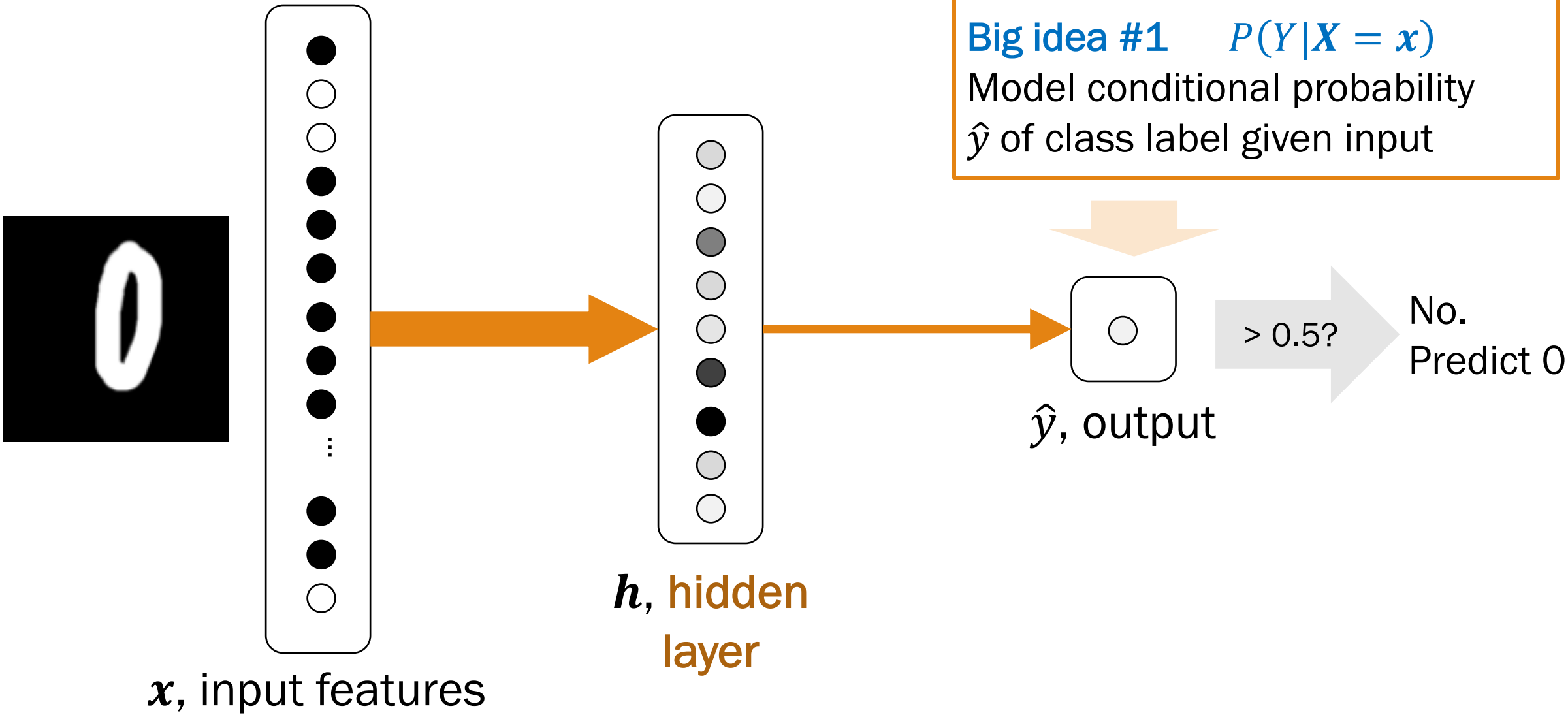
Non-linear transform of multiple values into one value, using parameter θ

$$\hat{y} = \sigma(\theta^T \boldsymbol{x})$$

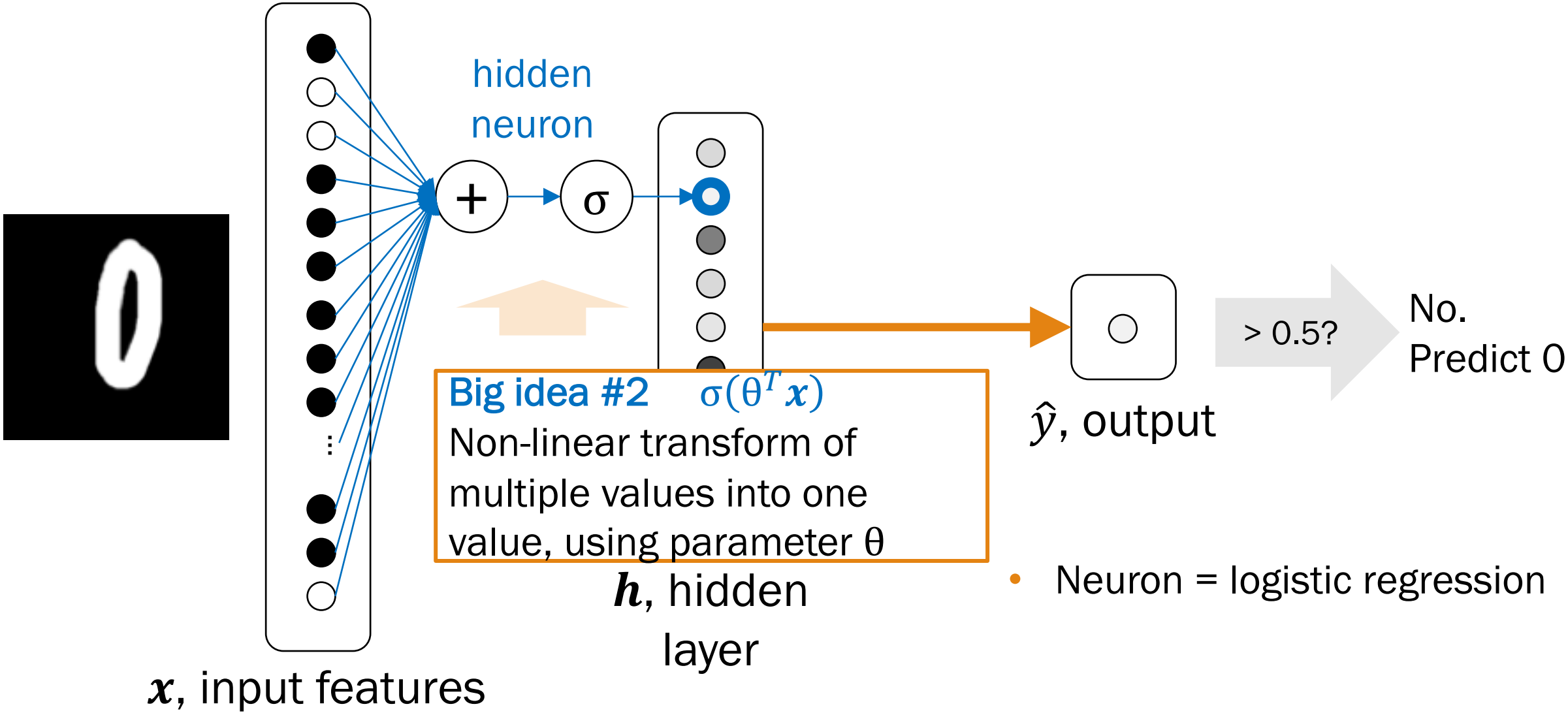
Neural network



Neural network



Feed neurons into other neurons

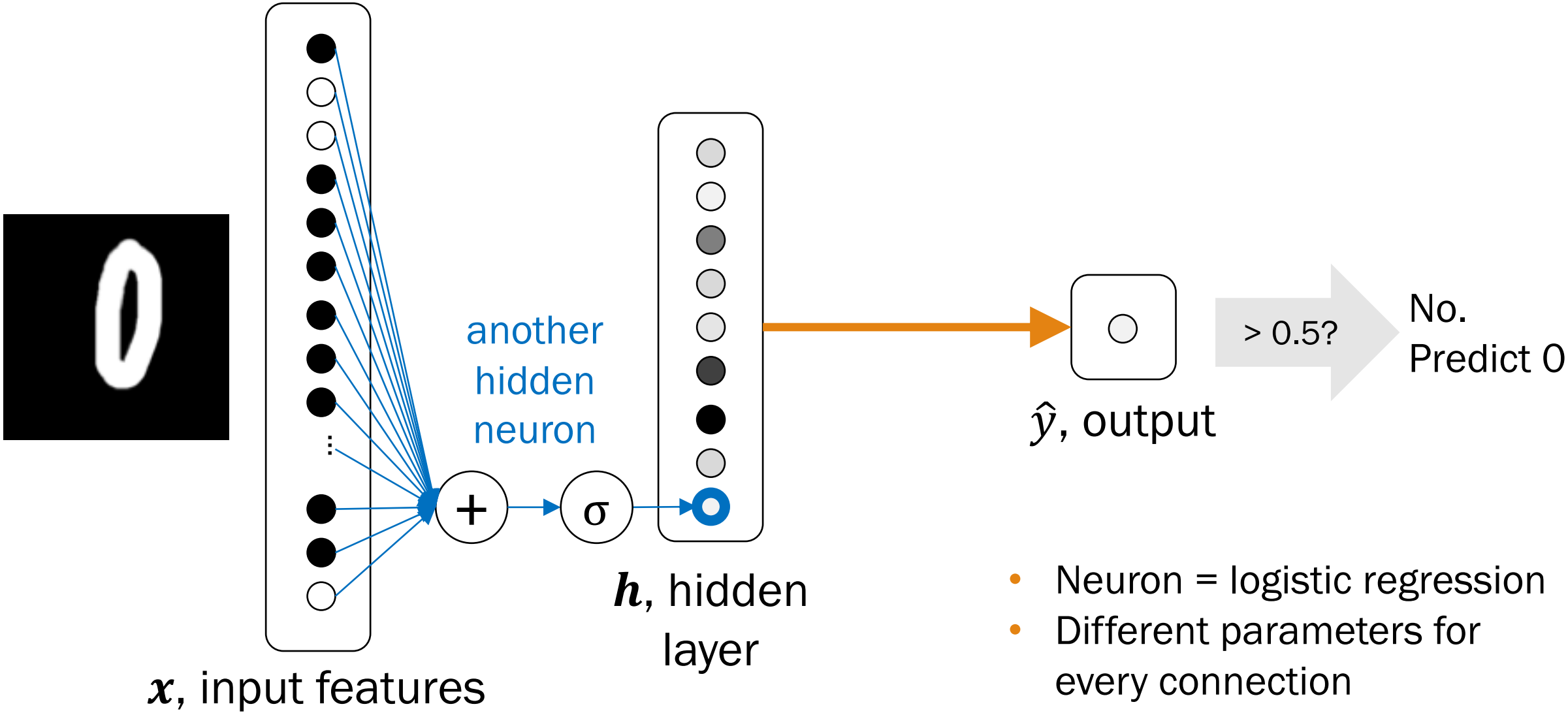


Big idea #2 $\sigma(\theta^T \boldsymbol{x})$

Non-linear transform of multiple values into one value, using parameter θ

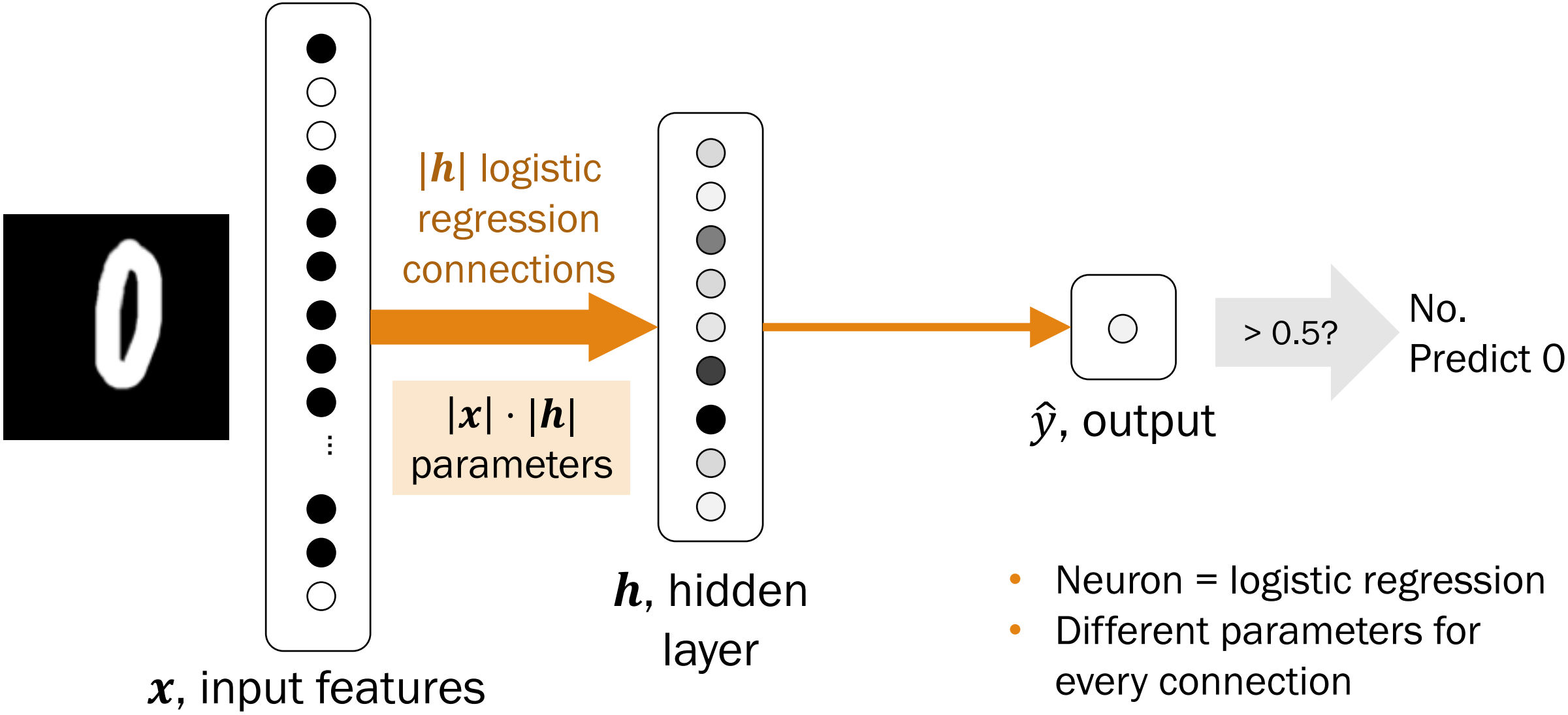
- Neuron = logistic regression

Feed neurons into other neurons



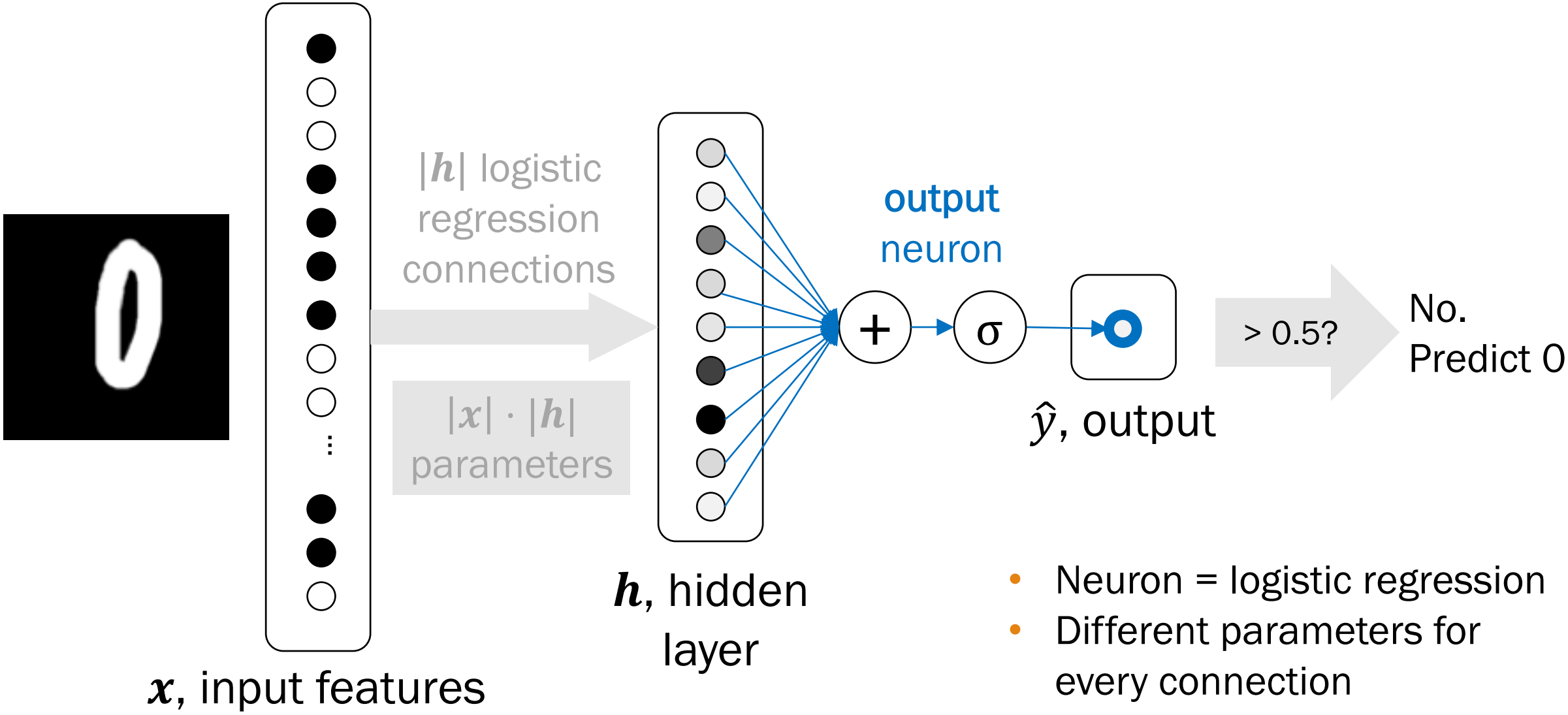
- Neuron = logistic regression
- Different parameters for every connection

Feed neurons into other neurons



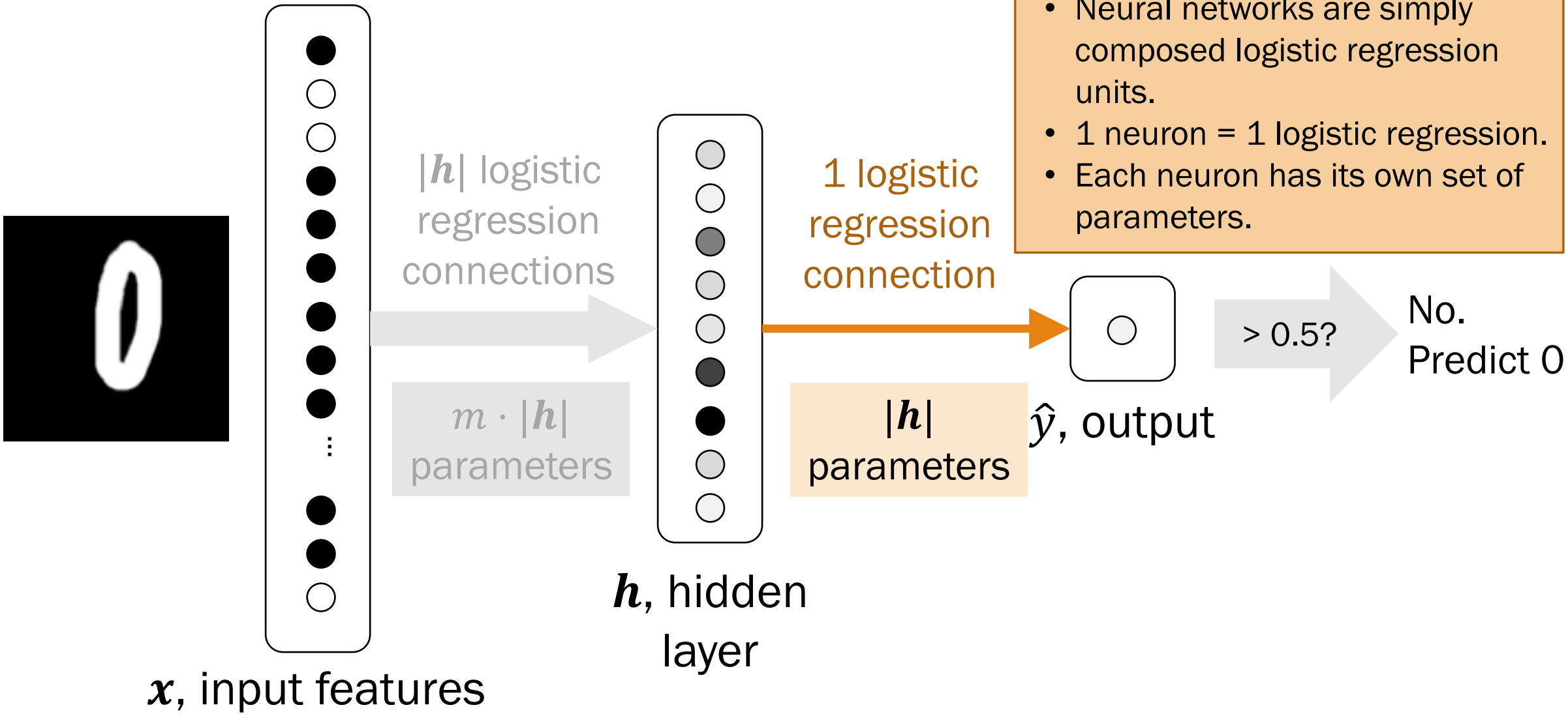
- Neuron = logistic regression
- Different parameters for every connection

Feed neurons into other neurons



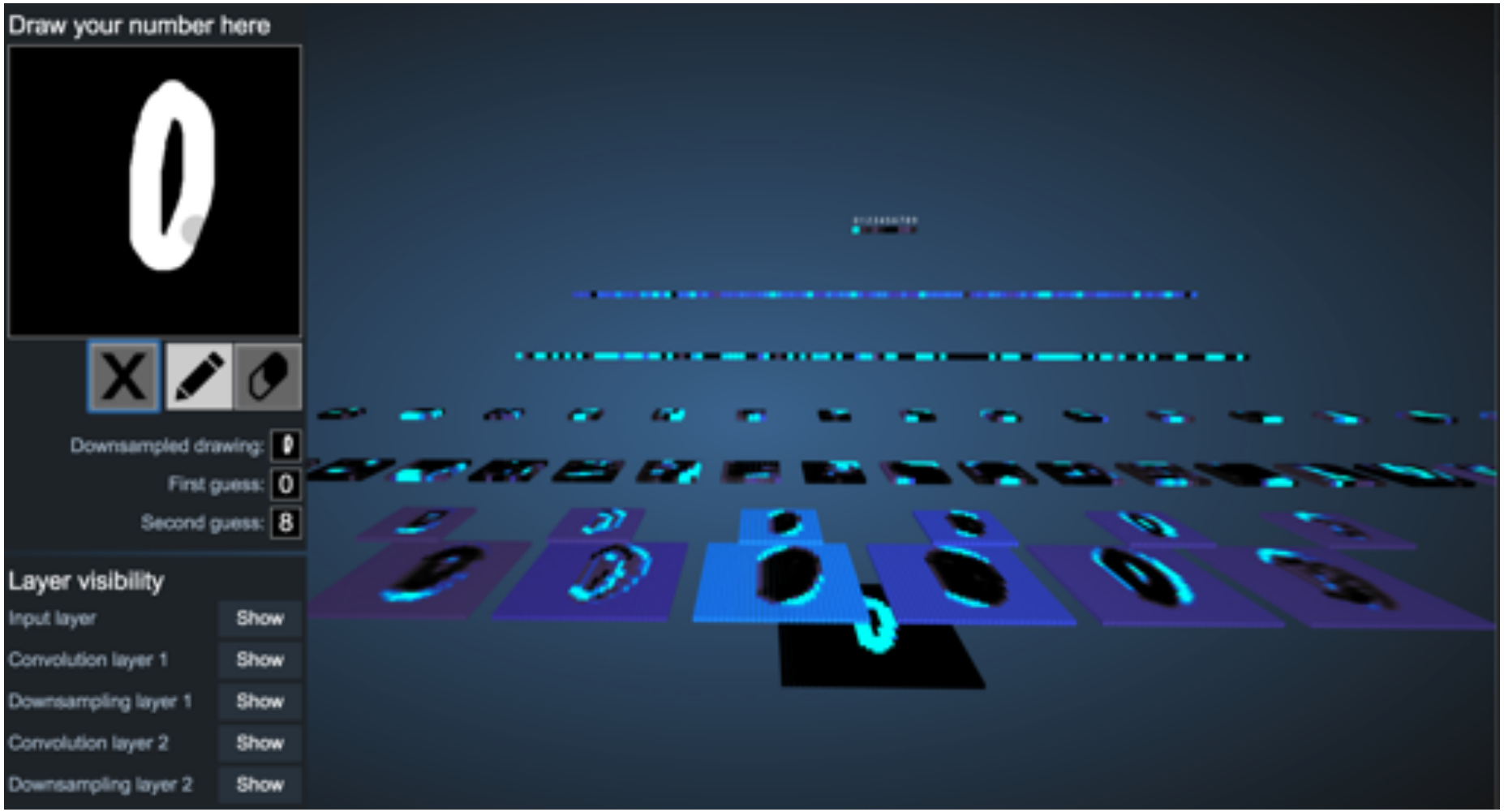
- Neuron = logistic regression
- Different parameters for every connection

Feed neurons into other neurons



- Neural networks are simply composed logistic regression units.
- 1 neuron = 1 logistic regression.
- Each neuron has its own set of parameters.

Demonstration



http://scs.ryerson.ca/~aharley/vis/conv/

Neural networks

A neural network (like logistic regression) gets intelligence from its parameters θ.

Training

- Learn parameters θ
- Find θ_{MLE} that maximizes likelihood of training data (MLE)

Testing/ Prediction

For input feature vector $\boldsymbol{X} = \boldsymbol{x}$:

- Use parameters to compute $\hat{y} = P(Y = 1 | \boldsymbol{X} = \boldsymbol{x})$
- Classify instance as: $\begin{cases} 1 & \hat{y} > 0.5 \\ 0 & \text{otherwise} \end{cases}$

Neural networks

A neural network (like logistic regression) gets intelligence from its parameters θ.

Training

- Learn parameters θ
- Find θ_{MLE} that maximizes likelihood of training data (MLE)

How do we learn the $|\boldsymbol{x}| \cdot |\boldsymbol{h}| + |\boldsymbol{h}|$ parameters?

Gradient ascent + chain rule!

Training Logistic Regression

Review

1. Optimization problem:

$$\theta_{MLE} = \arg\max_{\theta} \prod_{i=1}^{n} f\left(y^{(i)} \mid \boldsymbol{x}^{(i)}, \theta\right) = \arg\max_{\theta} \; LL(\theta)$$

$$LL(\theta) = \sum_{i=1}^{n} y^{(i)} \log \hat{y}^{(i)} + \left(1 - y^{(i)}\right) \log\left(1 - \hat{y}^{(i)}\right)$$

$$\hat{y} = \sigma\left(\theta^T \boldsymbol{x}^{(i)}\right)$$

2. Compute gradient

$$\frac{\partial LL(\theta)}{\partial \theta_j} = \sum_{i=1}^{n} \left[y^{(i)} - \hat{y}\right] x_j^{(i)}$$

3. Optimize

```
initialize params
repeat many times:
  compute gradient
   params += η * gradient
```

Training a neural net

1. Optimization problem:

$$\theta_{MLE} = \arg\max_{\theta} \prod_{i=1}^{n} f\left(y^{(i)} \mid \boldsymbol{x}^{(i)}, \theta\right) = \arg\max_{\theta} \; LL(\theta)$$

2. Compute gradient

3. Optimize

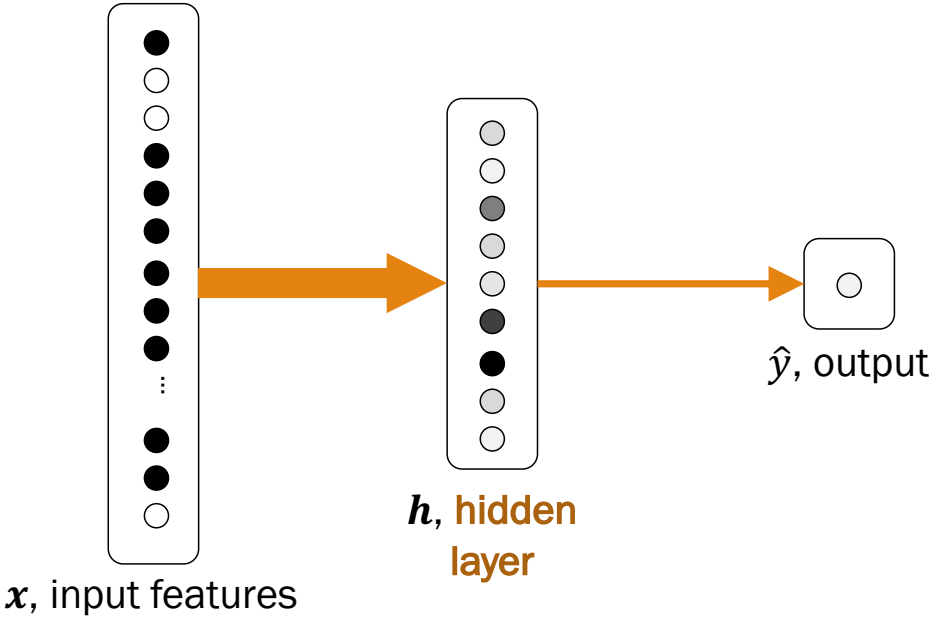
1. Same output $\hat{y}$, same conditional log-likelihood

$$P(Y = y | \boldsymbol{X} = \boldsymbol{x}) = \begin{cases} \hat{y} & \text{if } y = 1 \\ 1 - \hat{y} & \text{if } y = 0 \end{cases}$$

$$L(\theta) = \prod_{i=1}^{n} P\left(Y = y^{(i)} | \boldsymbol{X} = \boldsymbol{x}^{(i)}, \theta\right)$$

$$= \prod_{i=1}^{n} \left(\hat{y}^{(i)}\right)^{y^{(i)}} \left(1 - \hat{y}^{(i)}\right)^{1 - y^{(i)}}$$

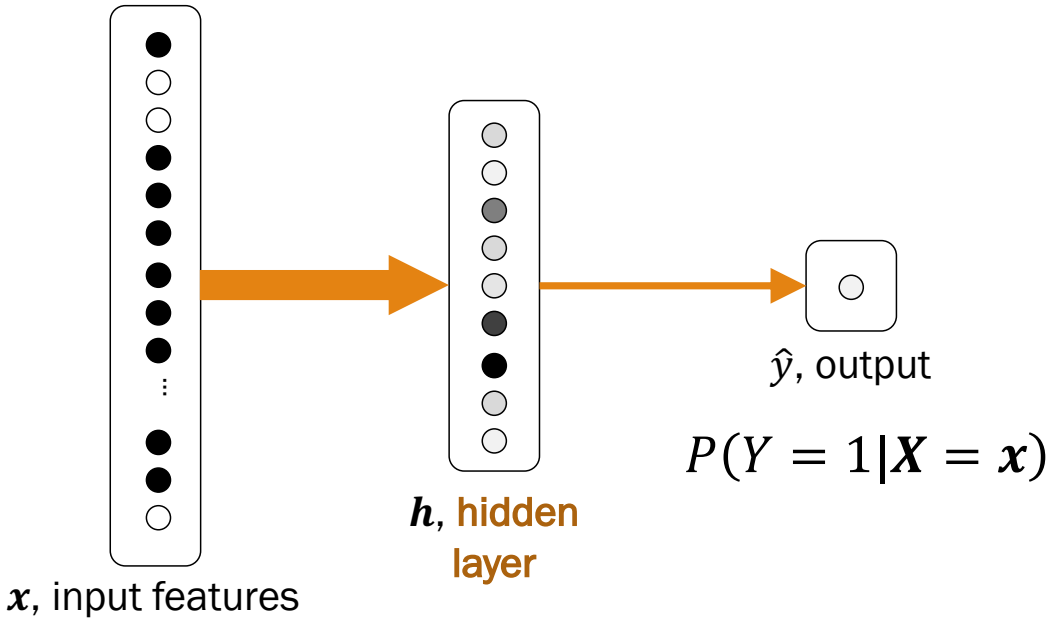
$$LL(\theta) = \sum_{i=1}^{n} y^{(i)} \log \hat{y}^{(i)} + \left(1 - y^{(i)}\right) \log\left(1 - \hat{y}^{(i)}\right)$$



1. (model is a little more complicated, though)

$$P(Y = y|\boldsymbol{X} = \boldsymbol{x}) = \begin{cases} \hat{y} & \text{if } y = 1 \\ 1 - \hat{y} & \text{if } y = 0 \end{cases}$$

$$LL(\theta) = \sum_{i=1}^{n} y^{(i)} \log \hat{y}^{(i)} + \left(1 - y^{(i)}\right) \log\left(1 - \hat{y}^{(i)}\right)$$



$$h_j = \sigma\left(\boxed{\theta_j^{(h)}}^T \boldsymbol{x}\right) \text{ for } j = 1, \ldots, |\boldsymbol{h}|$$

$$\hat{y} = \sigma\left(\boxed{\theta^{(\hat{y})}}^T \boldsymbol{h}\right)$$

$|\boldsymbol{h}|$ param vectors, each with dimension $|\boldsymbol{x}|$

$|\boldsymbol{h}|$-dimensional param vector

2. Compute gradient

1. Optimization problem:

$$\theta_{MLE} = \arg\max_{\theta} \prod_{i=1}^{n} f\left(y^{(i)} \mid \boldsymbol{x}^{(i)}, \theta\right) = \arg\max_{\theta} \; LL(\theta)$$

$$LL(\theta) = \sum_{i=1}^{n} y^{(i)} \log \hat{y}^{(i)} + \left(1 - y^{(i)}\right) \log\left(1 - \hat{y}^{(i)}\right)$$

$$h_j = \sigma\left(\boxed{\theta_j^{(h)}}^T \boldsymbol{x}\right) \quad \text{for } j = 1, \ldots, |\boldsymbol{h}| \qquad \hat{y} = \sigma\left(\boxed{\theta^{(\hat{y})}}^T \boldsymbol{h}\right)$$

2. Compute gradient

Take gradient with respect to all θ parameters

3. Optimize

Calculus refresher #1:
Derivative(sum) = sum(derivative)

Calculus refresher #2:
Chain rule ⭐ ⭐ ⭐

3. Optimize

1. Optimization problem:

$$\theta_{MLE} = \arg\max_{\theta} \prod_{i=1}^{n} f\left(y^{(i)} \mid \boldsymbol{x}^{(i)}, \theta\right) = \arg\max_{\theta} \; LL(\theta)$$

$$LL(\theta) = \sum_{i=1}^{n} y^{(i)} \log \hat{y}^{(i)} + \left(1 - y^{(i)}\right) \log\left(1 - \hat{y}^{(i)}\right)$$

$$h_j = \sigma\left(\boxed{\theta_j^{(h)}}^T \boldsymbol{x}\right) \quad \text{for } j = 1, \ldots, |\boldsymbol{h}| \qquad \hat{y} = \sigma\left(\boxed{\theta^{(\hat{y})}}^T \boldsymbol{h}\right)$$

2. Compute gradient

Take gradient with respect to all θ parameters

3. Optimize

```
initialize params
repeat many times:
  compute gradient
   params += η * gradient
```

Training a neural net

1. Optimization problem:

$$\theta_{MLE} = \arg\max_{\theta} \prod_{i=1}^{n} f\left(y^{(i)} \mid \boldsymbol{x}^{(i)}, \theta\right) = \arg\max_{\theta} \; LL(\theta)$$

$$LL(\theta) = \sum^{n} y^{(i)} \log \hat{y}^{(i)} + \left(1 - y^{(i)}\right) \log\left(1 - \hat{y}^{(i)}\right)$$

$$\theta^{(\hat{y})^T} \boldsymbol{h}\Big)$$

Wait, did we just skip something difficult?

2. Compute ... ers

3. Optimize

```
initialize params
repeat many times:
  compute gradient
   params += η * gradient
```

2. Compute gradient via backpropagation

1. Optimization problem:

$$\theta_{MLE} = \arg\max_{\theta} \prod_{i=1}^{n} f\left(y^{(i)} \mid \boldsymbol{x}^{(i)}, \theta\right) = \arg\max_{\theta} \; LL(\theta)$$

$$LL(\theta) = \sum_{i=1}^{n} y^{(i)} \log \hat{y}^{(i)} + \left(1 - y^{(i)}\right) \log\left(1 - \hat{y}^{(i)}\right)$$

$$h_j = \sigma\left(\boxed{\theta_j^{(h)}}^T \boldsymbol{x}\right) \quad \text{for } j = 1, \ldots, |\boldsymbol{h}| \qquad \hat{y} = \sigma\left(\boxed{\theta^{(\hat{y})}}^T \boldsymbol{h}\right)$$

2. Compute gradient

Take gradient with respect to all θ parameters

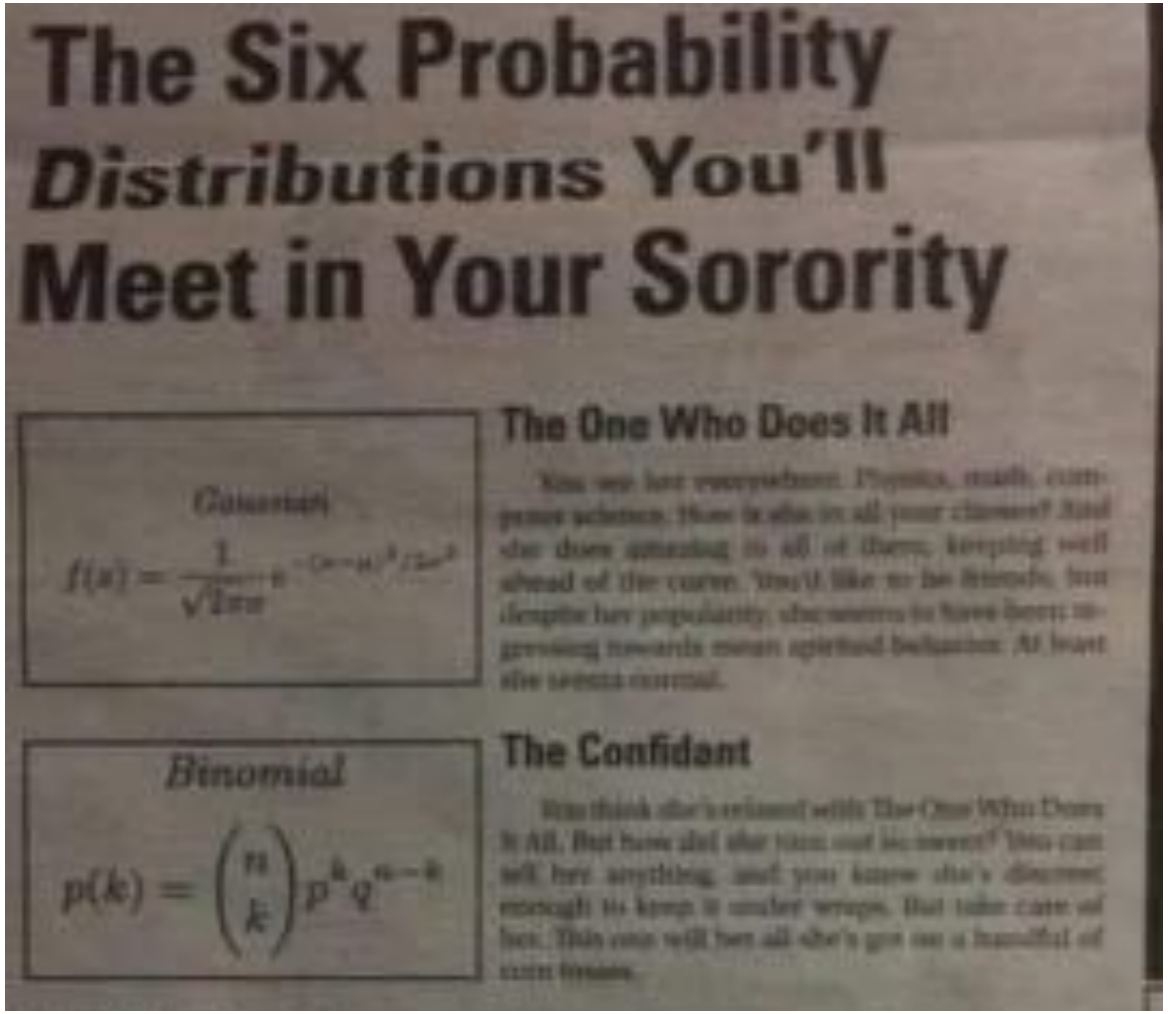
3. Optimize

```
initia
repeat
  comp
   par
```

Learn the tricks behind **backpropagation** in CS229, CS231N, CS224N, etc.

Interlude for jokes

Probability as college students



The Six Probability Distributions You'll Meet in Your Sorority

The One Who Does It All

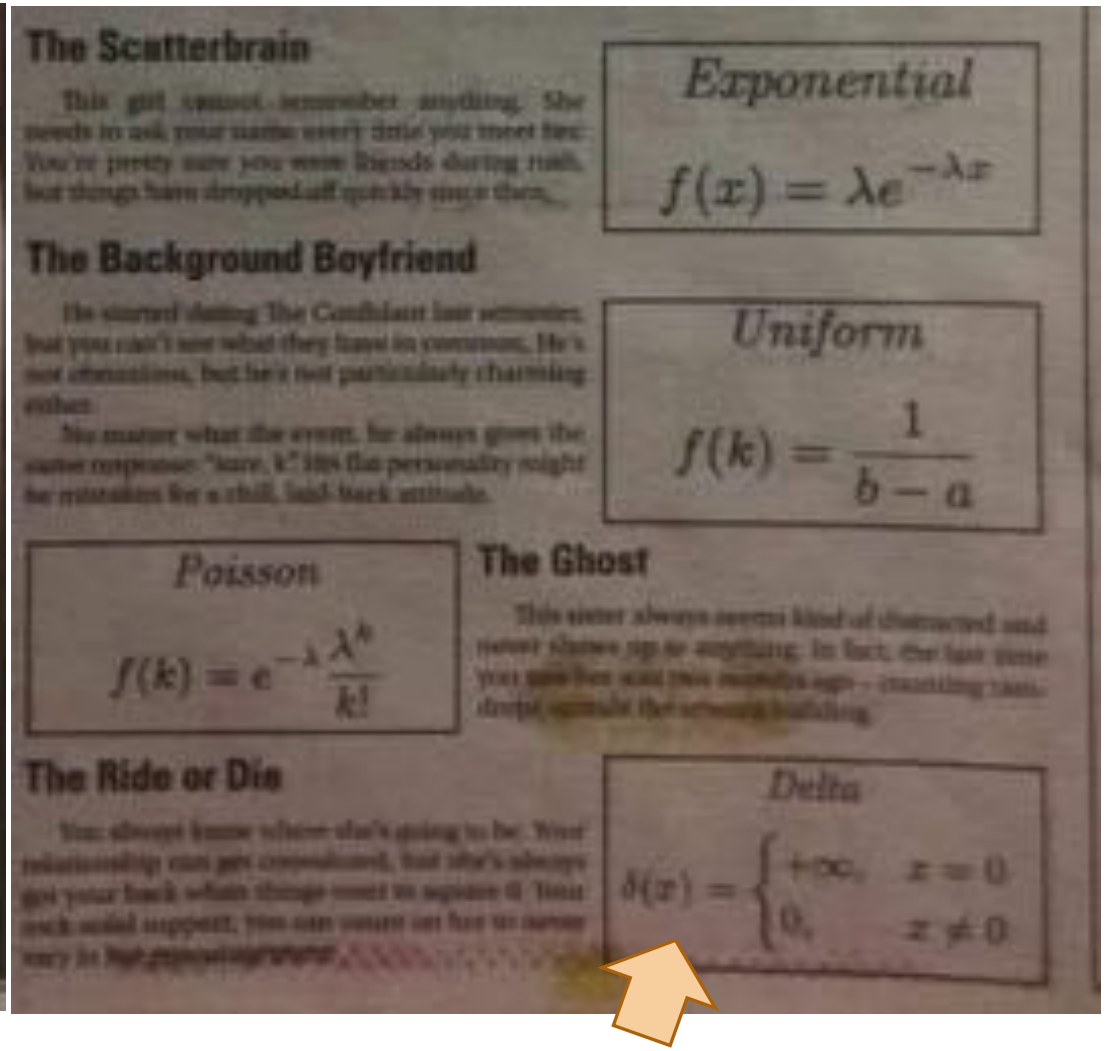
Gaussian

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2}$$

The Confidant

Binomial

$$p(k) = \binom{n}{k} p^k q^{n-k}$$



The Scatterbrain

Exponential

$$f(x) = \lambda e^{-\lambda x}$$

The Background Boyfriend

Uniform

$$f(k) = \frac{1}{b-a}$$

Poisson

$$f(k) = e^{-\lambda} \frac{\lambda^k}{k!}$$

The Ghost

The Ride or Die

Delta

$$\delta(x) = \begin{cases} +\infty, & x = 0 \\ 0, & x \neq 0 \end{cases}$$

(Note: not a valid PDF, but a useful construct nonetheless)

All remaining jokes

(from mid-quarter feedback)

What is similar about ice cream and bees? If you eat too many of either you'll get a stomachache

Knock knock. Whose there. Dishes. Dishes who? Dishes a great class.

What happens to a frog's car when it breaks down? It gets toad away.

Why is your nose in the middle of your face? Because it's the scenter :)

Why does the Norwegian Navy have bar codes on the side of their ships? So when they come back to port they can...

Tell a joke about pizza! Or wait that might be too cheesy ;)

Why do you avoid the top of the bell curve? Because everyone there is mean.

What did the fish say when he swam into a wall?

"Three logicians walk into a bar. The bartender asks, "Will you all have a drink?" The first logician says "I don't know." The second logician says "I don't know." The third logician says "Yes.""

What do you call it when you've been pulling too many all-nighters studying probability, and you see the ghost of Carl Friedrich Gauss? ... ... para-NORMAL activity (I'll take my extra credit now please)

Lisa giving us a house tour.

Two random variables were talking in a bar. They thought they were being discreet but I heard their chatter continuously.

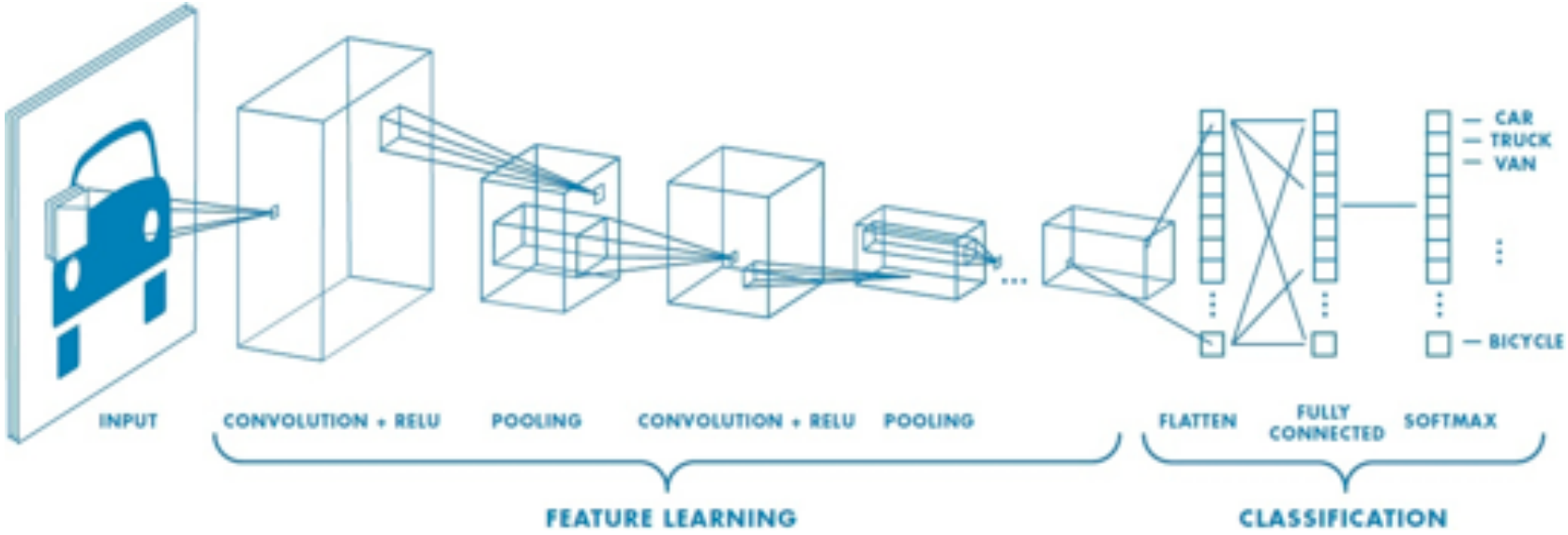
what has two butts and kills people?

"A statistics professor plans to travel to a conference by plane. When he passes the security check, they discover a bomb in his carry-on! Of course, he is hauled off immediately for interrogation. "I don't understand!" the

LIVE

Beyond the basics

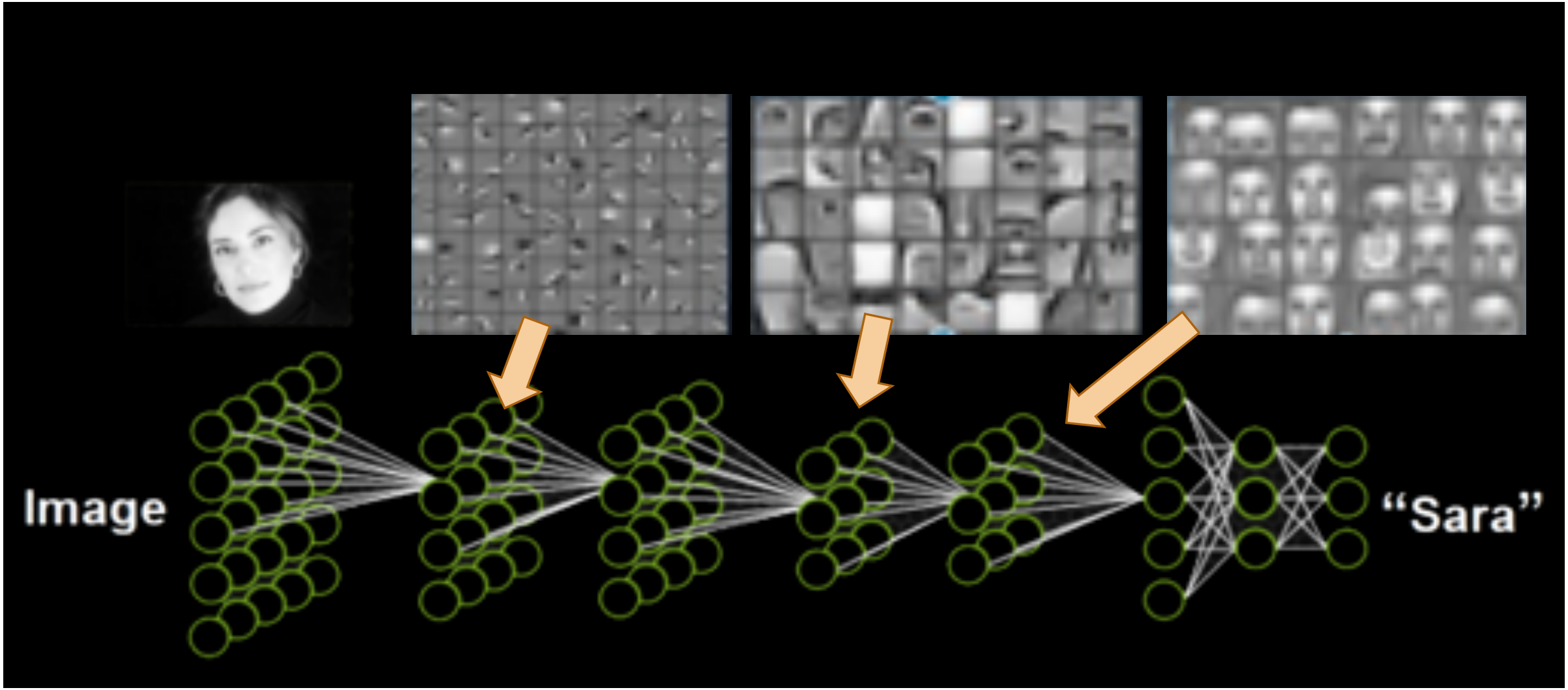
Shared weights?



It turns out if you want to force some of your weights to be shared over different neurons, the math isn't much harder.

Convolution is an example of such weight-sharing and is used a lot for vision (Convolutional Neural Networks, CNN).

Neural networks with multiple layers



x a b c d e f $\hat{y}$ LL

Neurons learn features of the dataset

Neurons in later layers will respond strongly to high-level features of your training data.

If your training data is faces, you will get lots of face neurons.

If your training data is all of YouTube…

…you get a cat neuron.

Top stimuli in test set

Optimal stimulus found by numerical optimization

Le, et al., Building high-level features using large-scale unsupervised learning. ICML 2012


Hire the smartest people in the world
Invent cat detector

Multiple outputs?

Softmax is a generalization of the sigmoid function.

sigmoid(z): value in range [0, 1]

$z \in \mathbb{R}$:

$$P(Y = 1 | \boldsymbol{X} = \boldsymbol{x}) = \sigma(z)$$

(equivalent: Bernoulli p)

softmax(z): k-dimensional values in range[0,1] that add up to 1

$\boldsymbol{z} \in \mathbb{R}^k$:

$$P(Y = i | \boldsymbol{X} = \boldsymbol{x}) = \text{softmax}(\boldsymbol{z})_i$$

(equivalent: Multinomial $p_1, \ldots, p_k$)

Softmax test metric: Top-5 error

$Y = y$	$P(Y = y \mid \boldsymbol{X} = \boldsymbol{x})$
5	0.14
8	0.13
7	0.12
2	0.10
9	0.10
4	0.09
1	0.09
0	0.09
6	0.08
3	0.05

Top-5 classification error

What % of datapoints did *not* have the correct class label in the top-5 predictions?

ImageNet classification

22,000 categories

14,000,000 images

Hand-engineered features (SIFT, HOG, LBP), Spatial pyramid, SparseCoding/Compression

…
smoothhound, smoothhound shark, Mustelus mustelus
American smooth dogfish, Mustelus canis
Florida smoothhound, Mustelus norrisi
whitetip shark, reef whitetip shark, Triaenodon obseus
Atlantic spiny dogfish, Squalus acanthias
Pacific spiny dogfish, Squalus suckleyi
hammerhead, hammerhead shark
smooth hammerhead, Sphyrna zygaena
smalleye hammerhead, Sphyrna tudes
shovelhead, bonnethead, bonnet shark, S
angel shark, angelfish, Squatina squatina, monkfish
electric ray, crampfish, numbfish, torpedo
smalltooth sawfish, Pristis pectinatus
guitarfish
roughtail stingray, Dasyatis centroura
butterfly ray
eagle ray
spotted eagle ray, spotted ray, Aetobatus narinari
cownose ray, cow-nosed ray, Rhinoptera bonasus
manta, manta ray, devilfish
Atlantic manta, Manta birostris
devil ray, Mobula hypostoma
grey skate, gray skate, Raja batis
little skate, Raja erinacea
…

Stingray

Mantaray

Le, et al., Building high-level features using large-scale unsupervised learning. ICML 2012

ImageNet classification challenge

~~22,000 categories~~

1000 categories

14,000,000 images

1,200,000 images in train set

200,000 images in test set

Hand-engineered features (SIFT, HOG, LBP), Spatial pyramid, SparseCoding/Compression

moothhound shark, Mustelus mustelus
n dogfish, Mustelus canis
Florida smoothhound, Mustelus norrisi
nodon obseus
smooth hammerhead, Sphyrna zygaena
smalleye hammerhead, Sphyrna tudes
shovelhead, bonnethead, bonnet shark, Sphyrna tiburo
angel shark, angelfish, Squatina squatina, monkfish
electric ray, crampfish, numbfish, torpedo
smalltooth sawfish, Pristis pectinatus
guitarfish
roughtail stingray, Dasyatis centroura
butterfly ray
eagle ray
spotted eagle ray, spotted ray, Aetobatus narinari
cownose ray, cow-nosed ray, Rhinoptera bonasus
manta, manta ray, devilfish
Atlantic manta, Manta birostris
devil ray, Mobula hypostoma
grey skate, gray skate, Raja batis
little skate, Raja erinacea
…

Le, et al., Building high-level features using large-scale unsupervised learning. ICML 2012

ImageNet challenge: Top-5 classification error

(lower is better)

99.5%

Random guess

$$P(\text{true class label not in 5 guesses}) = \frac{\binom{999}{5}}{\binom{1000}{5}} = \frac{995}{1000}$$

ImageNet challenge: Top-5 classification error

(lower is better)

99.5%

Random guess

25.8%

Pre-Neural Networks

5.1%

Humans
(2014)

16.4%

GoogLe Net
(2015)

Russakovsky et al., ImageNet Large Scale Visual Recognition Challenge. IJCV 2015
Szegedy et al., Going Deeper With Convolutions. CVPR 2015
Hu et al., Squeeze-and-Excitation Networks. Preprint arXiV 2017

ImageNet challenge: Top-5 classification error

(lower is better)

99.5%
Random guess

25.8%
Pre-Neural Networks

5.1%
Humans
(2014)

16.4%
GoogLe Net
(2015)

2.25%
SENet
(2017)

Russakovsky et al., ImageNet Large Scale Visual Recognition Challenge. IJCV 2015
Szegedy et al., Going Deeper With Convolutions. CVPR 2015
Hu et al., Squeeze-and-Excitation Networks. Preprint arXiV 2017

GoogLeNet (2015)

Google Brain

1 Trillion Artificial Neurons

(btw human brains have 1 billion neurons)

Multiple,
Multi class output

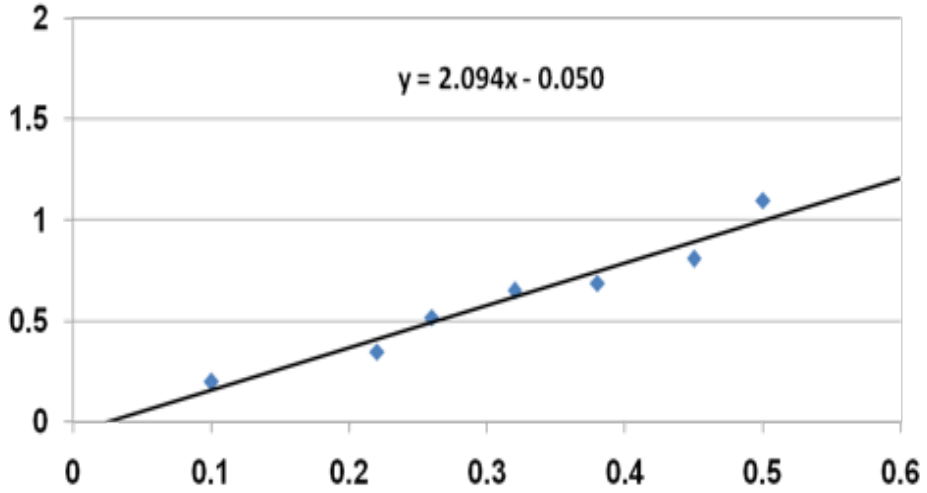
22 layers deep!

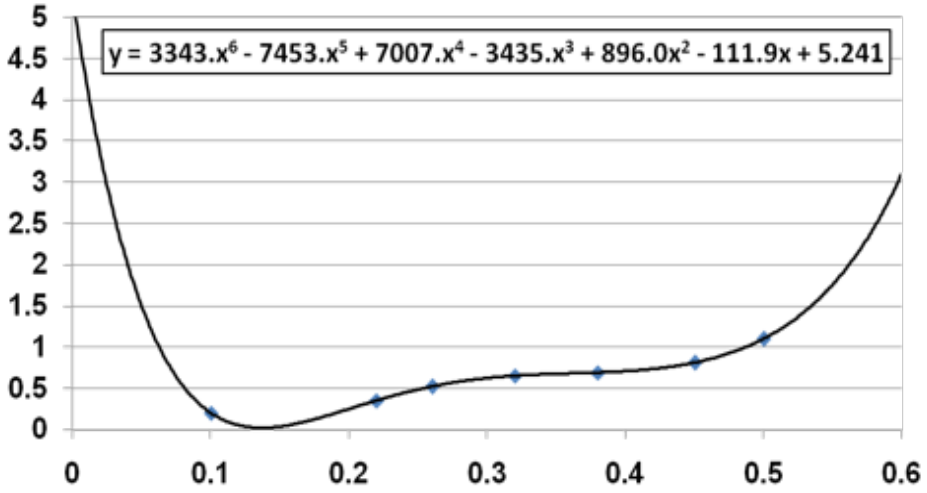
Szegedy et al., Going Deeper With Convolutions. CVPR 2015

Good ML = Generalization

Goal of machine learning: build models that *generalize* well to predicting new data

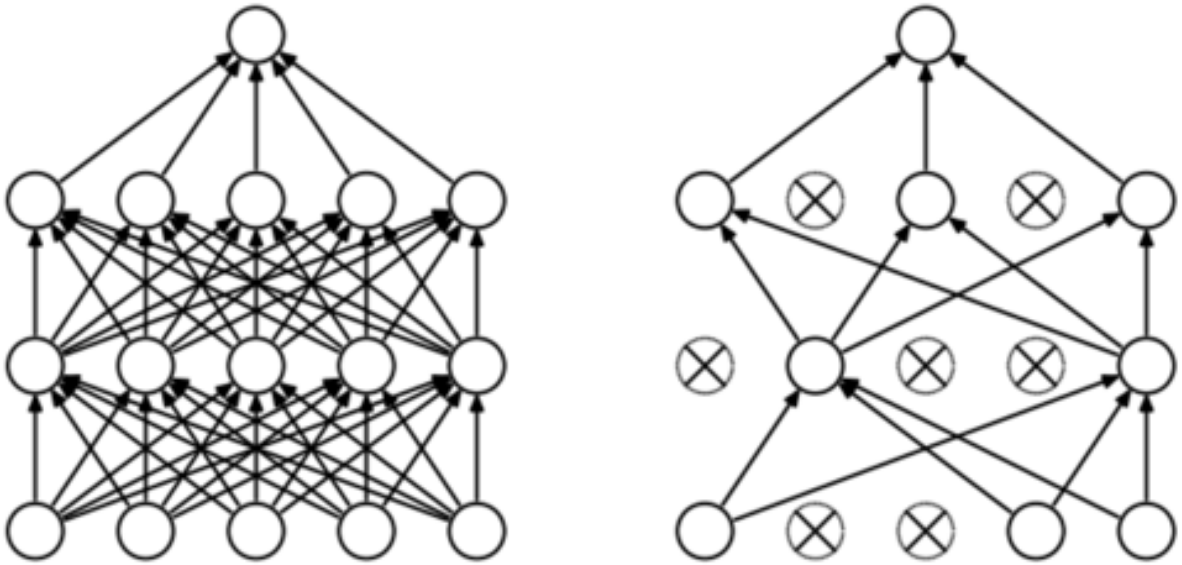
Overfitting: fitting the training data too well, so we lose generality of model





- Polynomial on the right fits training data perfectly!
- Which would you rather use to predict a new data point?

Prevent overfitting?



(a) Standard Neural Net (b) After applying dropout.

Dropout (training technique)

When your model is training, randomly turn off your neurons with probability 0.5.

It will make your network more robust.

Making decisions?

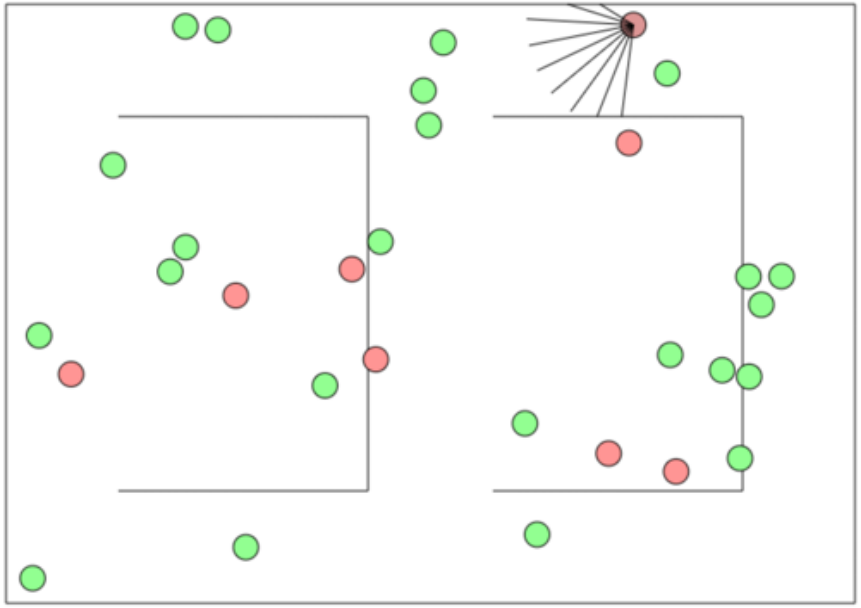
Not everything is classification.

Deep Reinforcement Learning

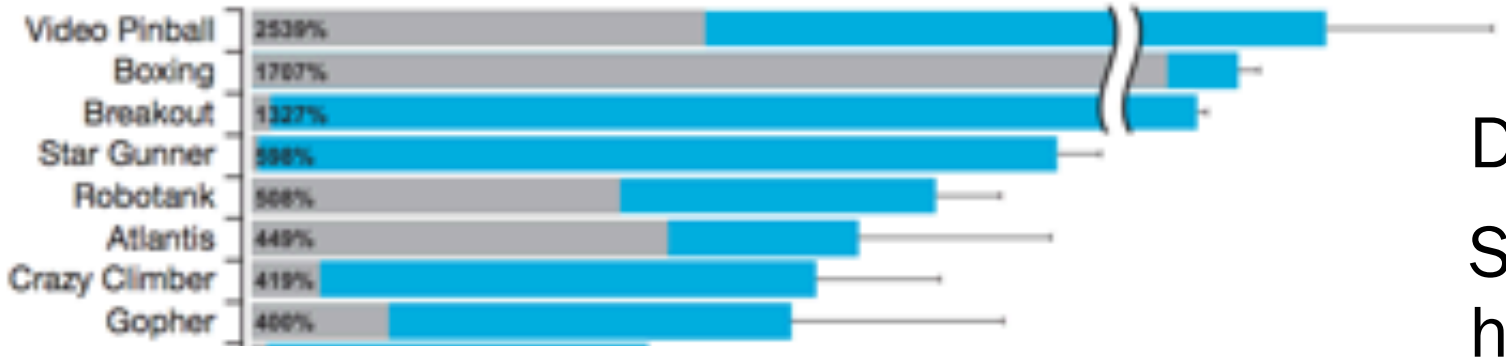
Instead of having the output of a model be a probability, you make output an expectation.

http://cs.stanford.edu/people/karpathy/convnetjs/demo/rldemo.html

Deep Reinforcement Learning



http://cs.stanford.edu/people/karpathy/convnetjs/demo/rldemo.html



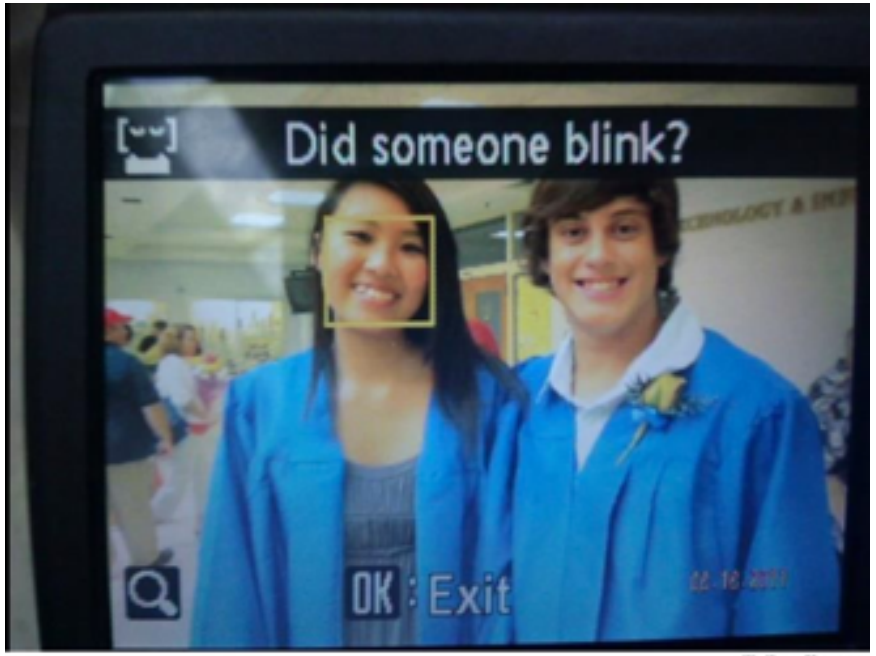
Deep Mind Atari Games

Score compared to best human

What’s missing?

Where is your data coming from?

Ethics and datasets



Sometimes machine learning feels universally unbiased.

We can even prove our estimators are “unbiased” (mathematically).

Google/Nikon/HP had biased datasets.

Should your data be unbiased?

Dataset: Google News

$$\overrightarrow{\text{man}} - \overrightarrow{\text{woman}} \approx \overrightarrow{\text{king}} - \overrightarrow{\text{queen}}$$

$$\overrightarrow{\text{man}} - \overrightarrow{\text{woman}} \approx \overrightarrow{\text{computer programmer}} - \overrightarrow{\text{homemaker}}.$$

Should our unbiased data collection reflect society's systemic bias?

Bolukbasi et al., Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. NIPS 2016

How can we explain decisions?

If your task is image classification, reasoning about high-level features is relatively easy.

Everything can be visualized.

What if you are trying to classify social outcomes?

- Criminal recidivism
- Job performance
- Policing
- Terrorist risk
- At-risk kids

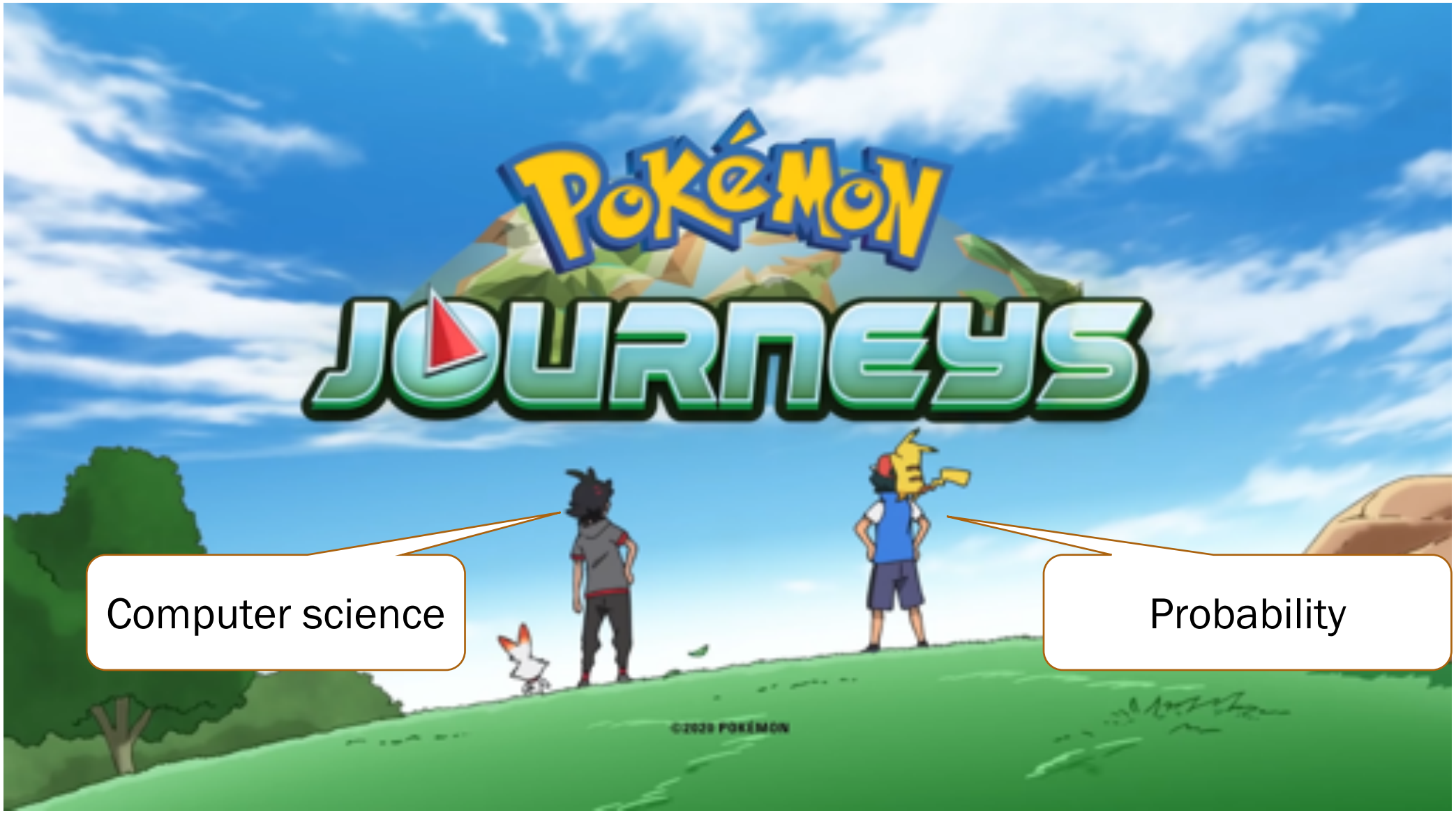
Ethics in Machine Learning is a whole new field. 🙂

LIVE

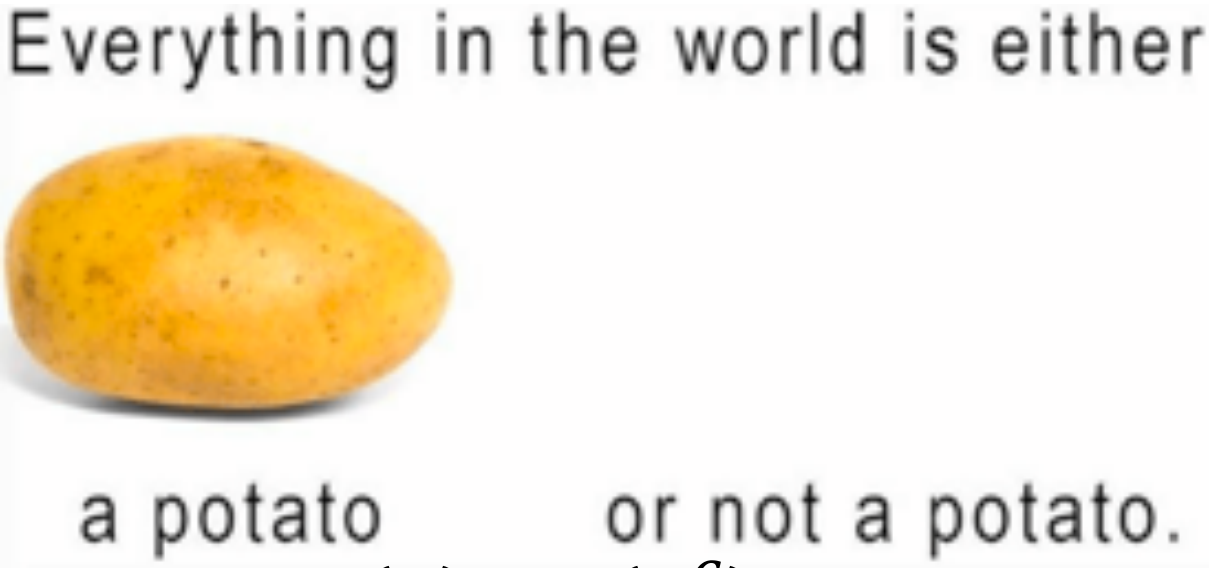
CS109 Wrap-Up

What have we learned in CS109?

A wild journey



From combinatorics to probability…

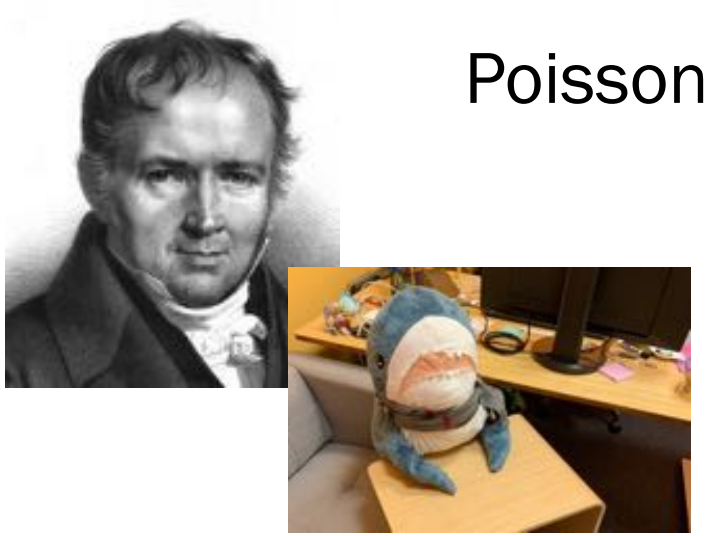


$$P(E) + P(E^C) = 1$$

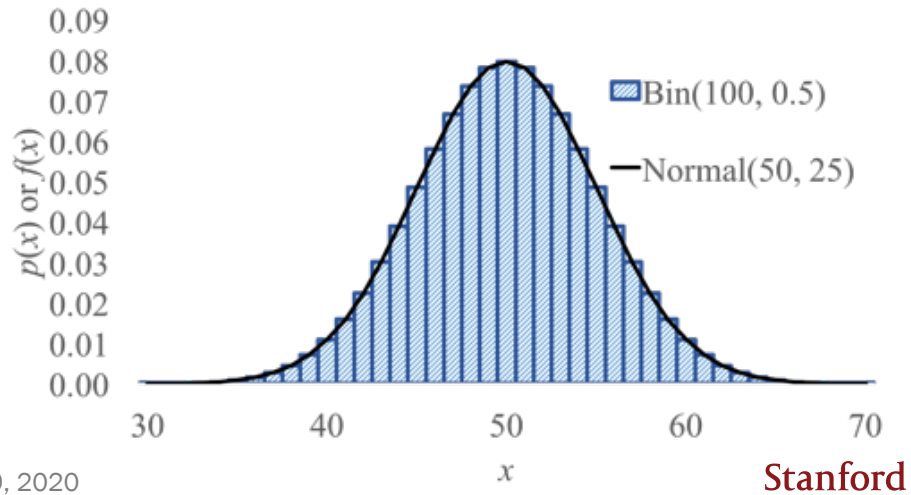
...to random variables and the Central Limit Theorem...

Bernoulli

Gaussian

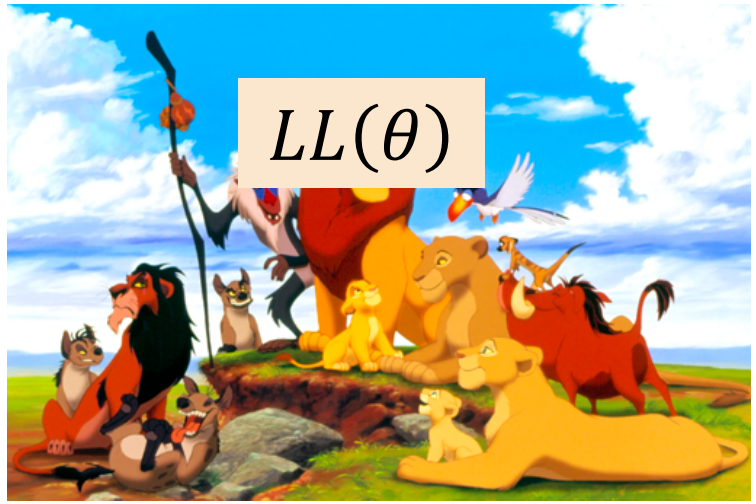


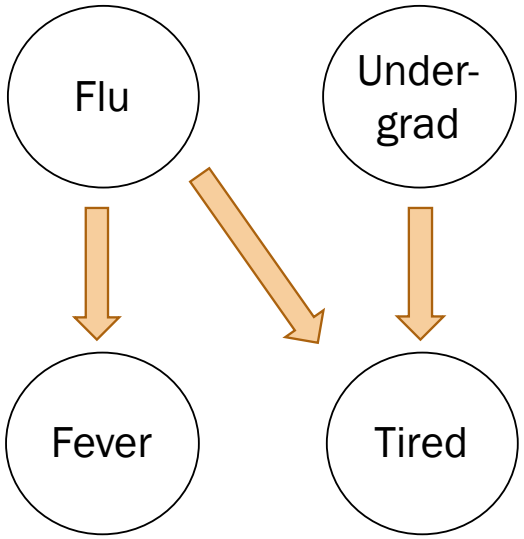
Poisson



...to statistics, parameter estimation, and machine learning

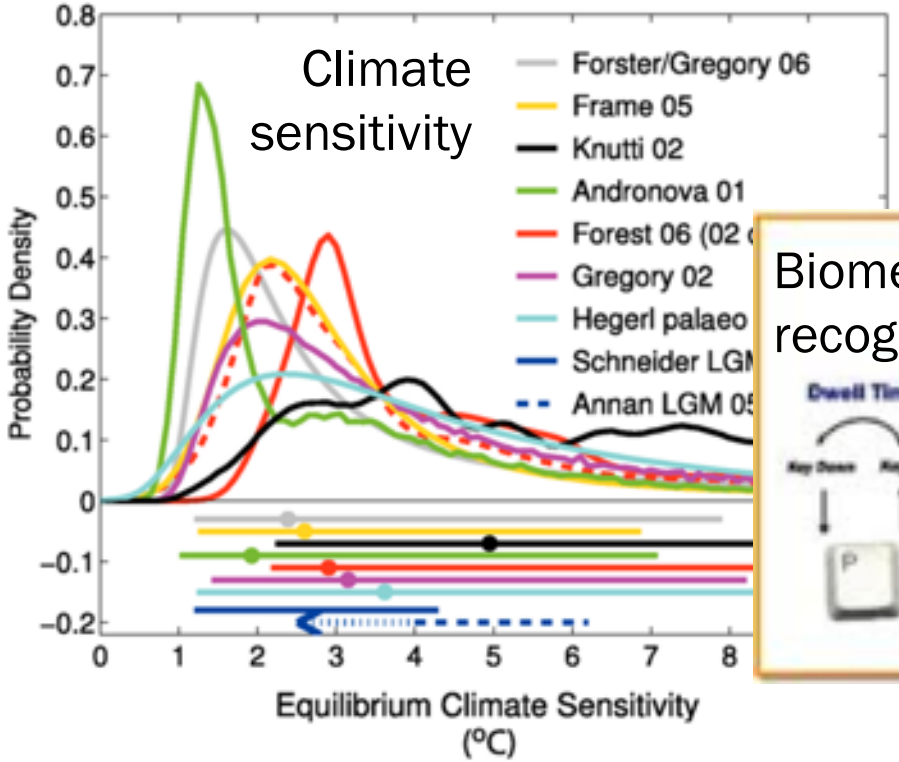
A happy
Bhutanese person

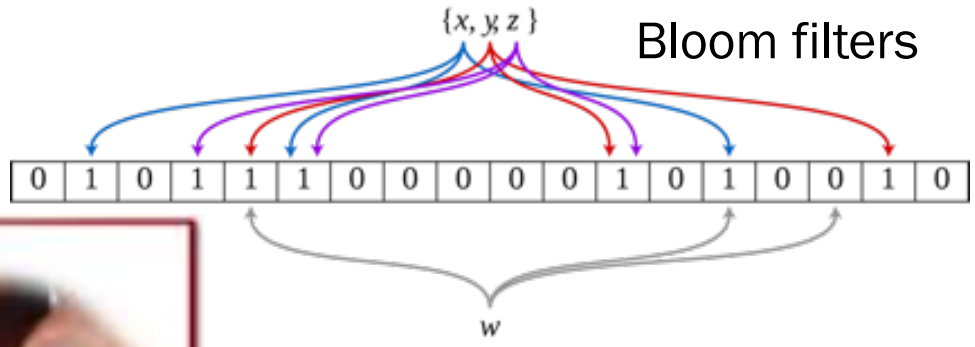


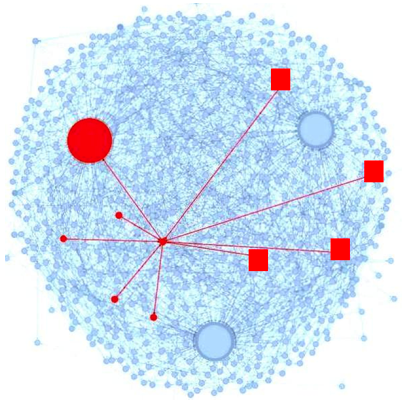


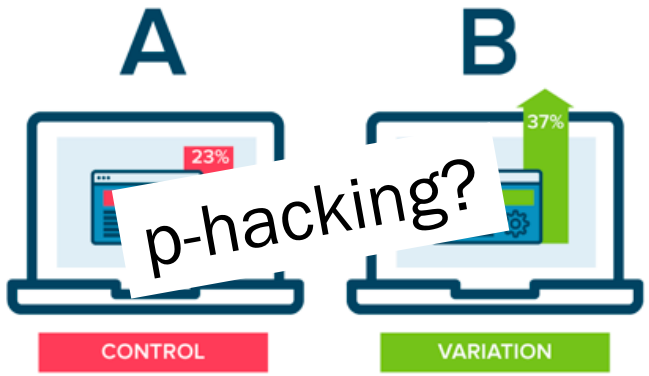
NETFLIX
and Learn

Lots and lots of analysis

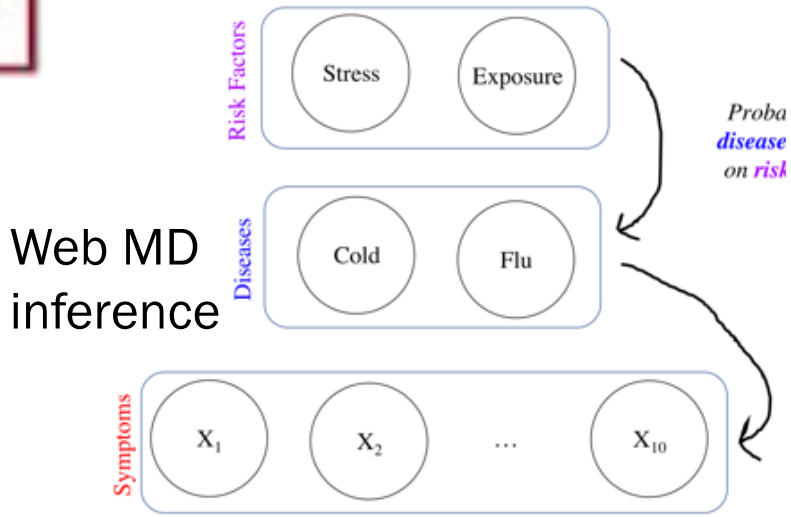






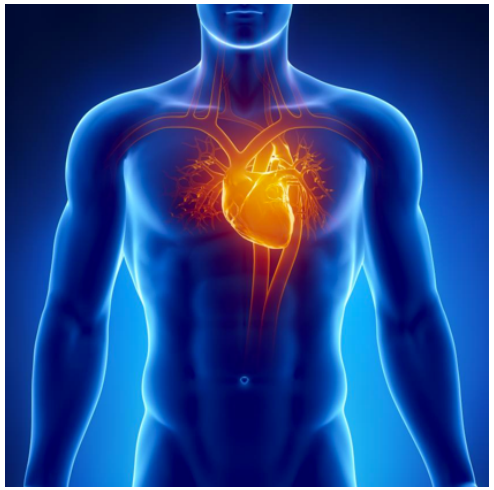


Coursera A/B testing



Lots and lots of analysis

Heart



Ancestry

23andMe

NETFLIX

Netflix

After CS109

Theory

CS161 – Algorithmic analysis
Stats 217 – Stochastic Processes
CS238 – Decision Making Under Uncertainty
CS228 – Probabilistic Graphical Models

Statistics

Stats 200 – Statistical Inference
Stats 208 – Intro to the Bootstrap
Stats 209 – Group Methods/Causal Inference

After CS109

AI

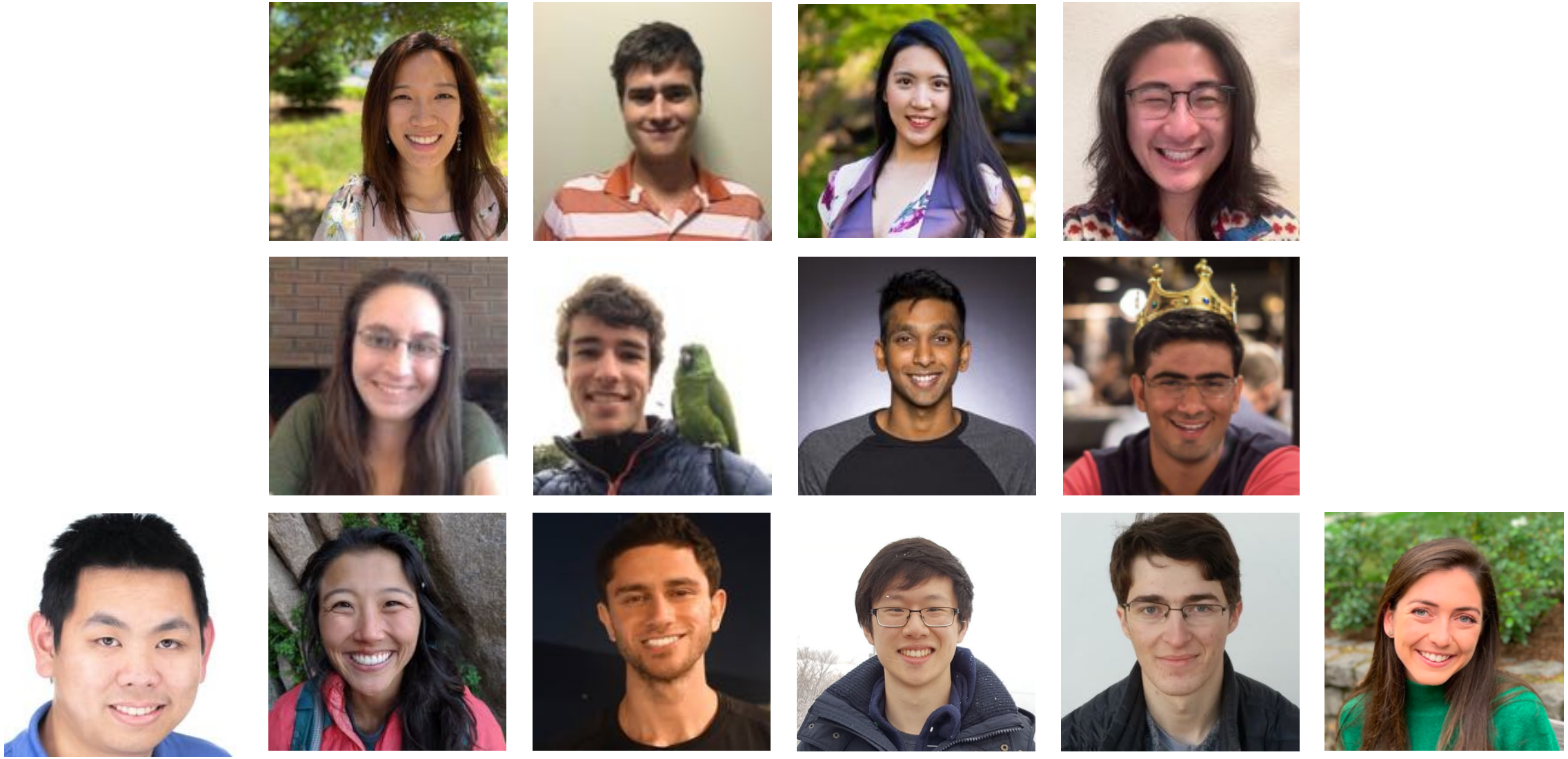
CS 221 – Intro to AI
CS 229 – Machine Learning
CS 230 – Deep Learning
CS 224N – Natural Language Processing
CS 231N – Conv Neural Nets for Visual Recognition
CS 234 – Reinforcement Learning

Applications

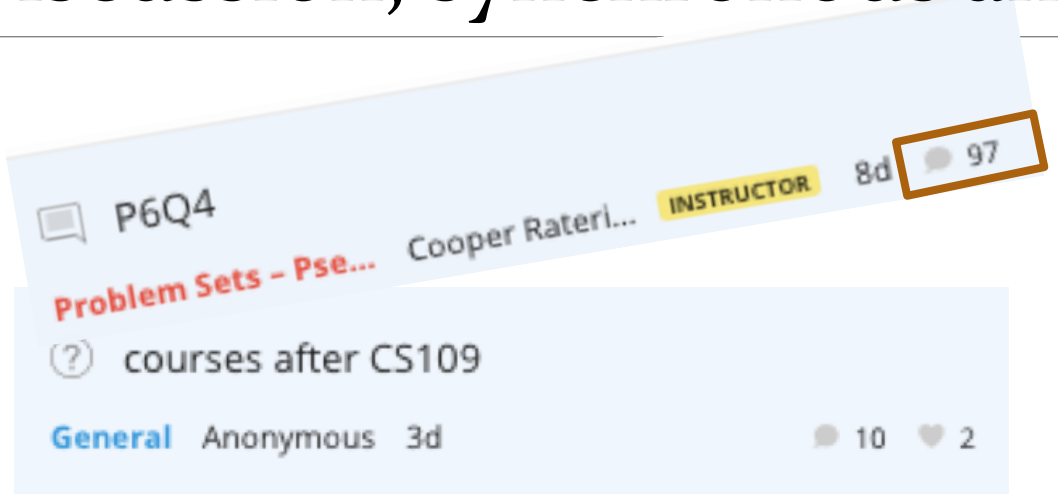
CS 279 – Bio Computation
Literally any class with numbers in it

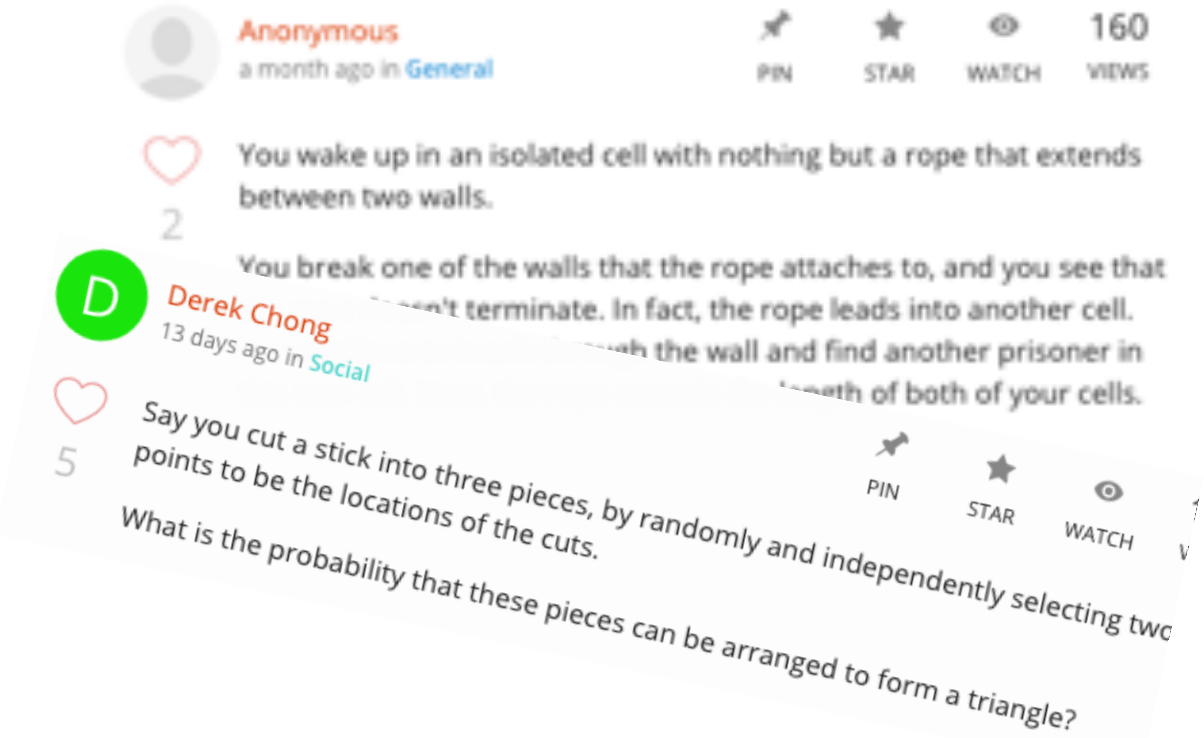
What have we done together this quarter?

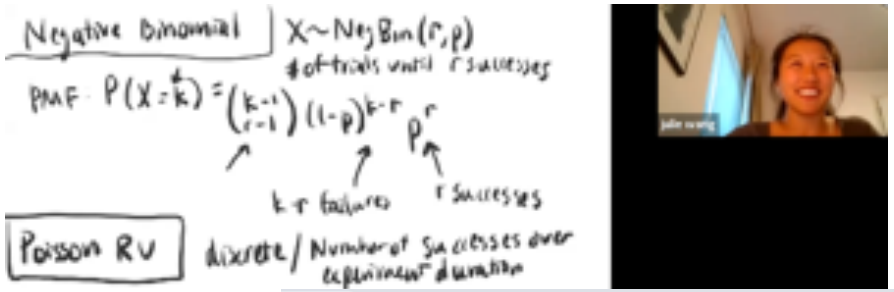
The CS109 teaching team



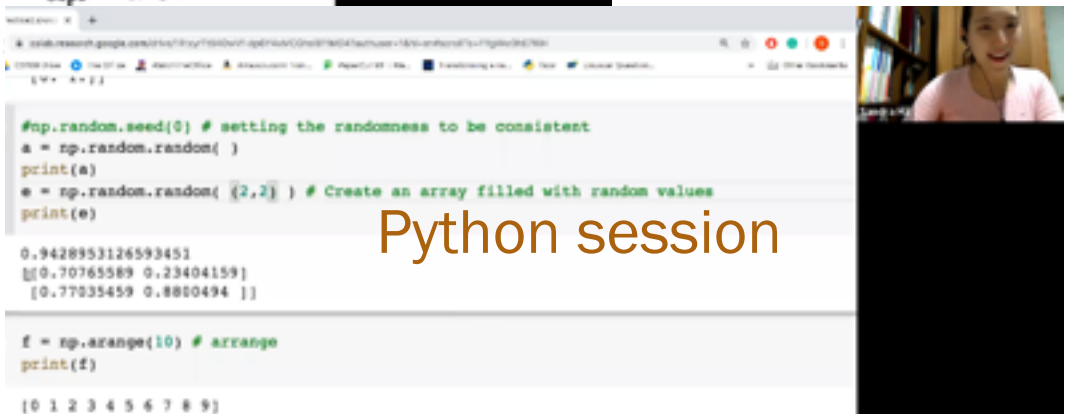
Discussion, synchronous and asynchronous



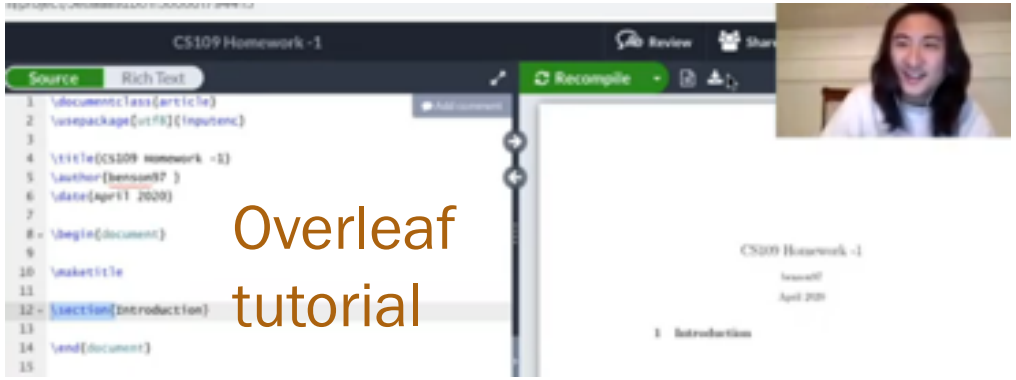




Section

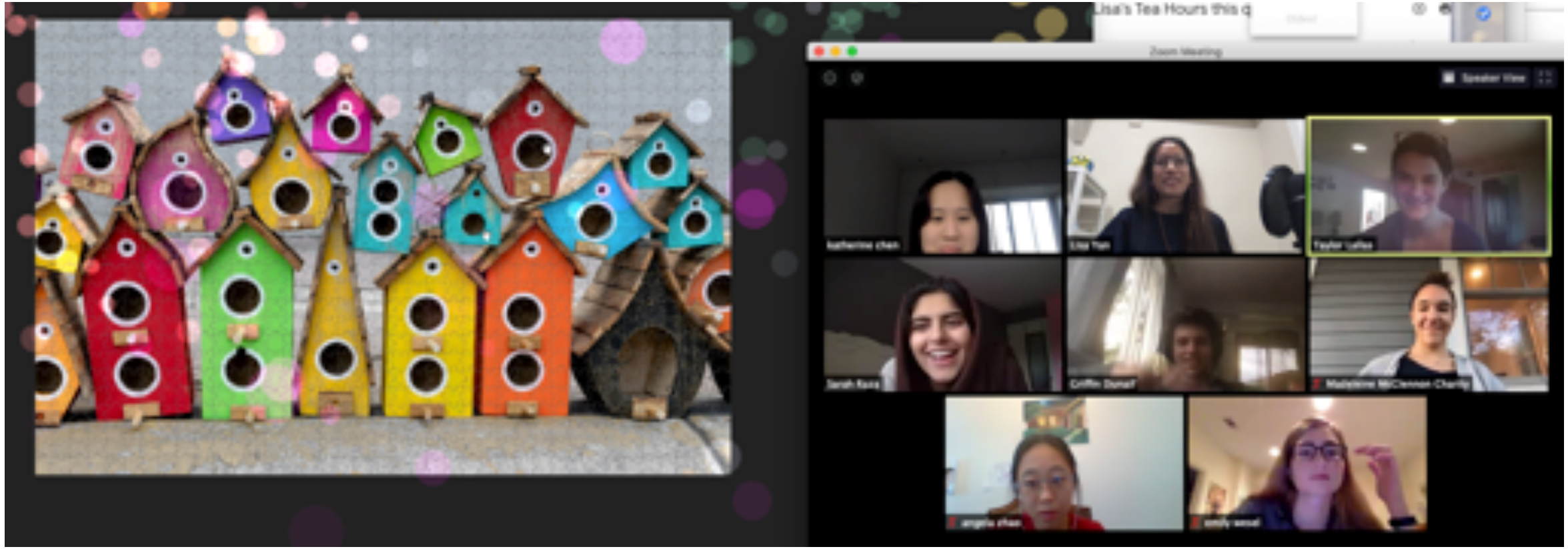


Python session



Overleaf tutorial

Tea hours



Turns out jigsaw puzzles aren't probabilistic!

What have you learned from CS109?

What do you want to remember in 5 years?

What do you want to remember in the next 5 years?

Probability has two school of thoughts, frequentist and Bayesian

Bayes Theorem

How to count (lol who knew I'd learn to count during college)

To be honest, counting.

Counting!

How to use random variables, bootstrapping, and random sampling to analyze data.

LOTUS because it helped me realize how I use stats to reason about the world in my everyday life

Always switch the door when playing Monty Hall!

This machine learning stuff! So cool!

Statistics can be p-hacked to misinterpret results

Machine Learning algorithms and Bayesian statistics! (also "cancellation city")

I want to be able to take data and be able to structure and model it in a way that makes it useful to me and the world!

That I was finally able to code a Naive Bayes Classifier

What do you want to remember in the next 5 years?

bay-gulls

Poisson!! the shark... but hopefully also the distribution

poisson shark

seagulls in the bay area are actually bagels!

Interlude for jokes LOL.

Lisa's jokes

the jokes!

Lisa's jokes, of course! A close second is bootstrapping! I think it is the coolest

the fun times!

What do you want to remember in the next 5 years?

The Probability in the Real World segments

I want to remember how probability is so relevant to many different things in the world

I learned a lot.

I want to remember the fundamentals of probability and the commitment of the teaching staff to making this tough quarter as smooth as possible.

How much I struggled.

I never knew how much probability was around me!

I want to remember how I learned about COVID-19 tracking through researching articles for CS109.

I'll want to remember how much learning helped me deal with the emotions of the quarter.

I'll remember feeling excited about stats

Thank you so much CS109! You will be missed.

Why study probability + CS?

Why study probability + CS?

Fastest growing occupations: 20 occupations with the highest percent change of employment between 2018-28.

Click on an occupation name to see the full occupational profile.

OCCUPATION	GROWTH RATE, 2018-28	2018 MEDIAN PAY
Physician assistants	31%	$108,610 per year
Nurse practitioners	28%	$107,030 per year
Software developers, applications	26%	$103,620 per year
Mathematicians	26%	$101,900 per year
Information security analysts	32%	$98,350 per year
Health specialties teachers, postsecondary	23%	$97,370 per year
Statisticians	31%	$87,780 per year
Operations research analysts	26%	$83,390 per year
Genetic counselors	27%	$80,370 per year

Source: US Bureau of Labor Statistics

Why study probability + CS?

Interdisciplinary

Closest thing to magic

Why study probability + CS?

Everyone is welcome!

Technology magnifies. What do we want magnified?

You are all one step closer to improving the world.

(all of you!)

The end

See you soon… ☺