

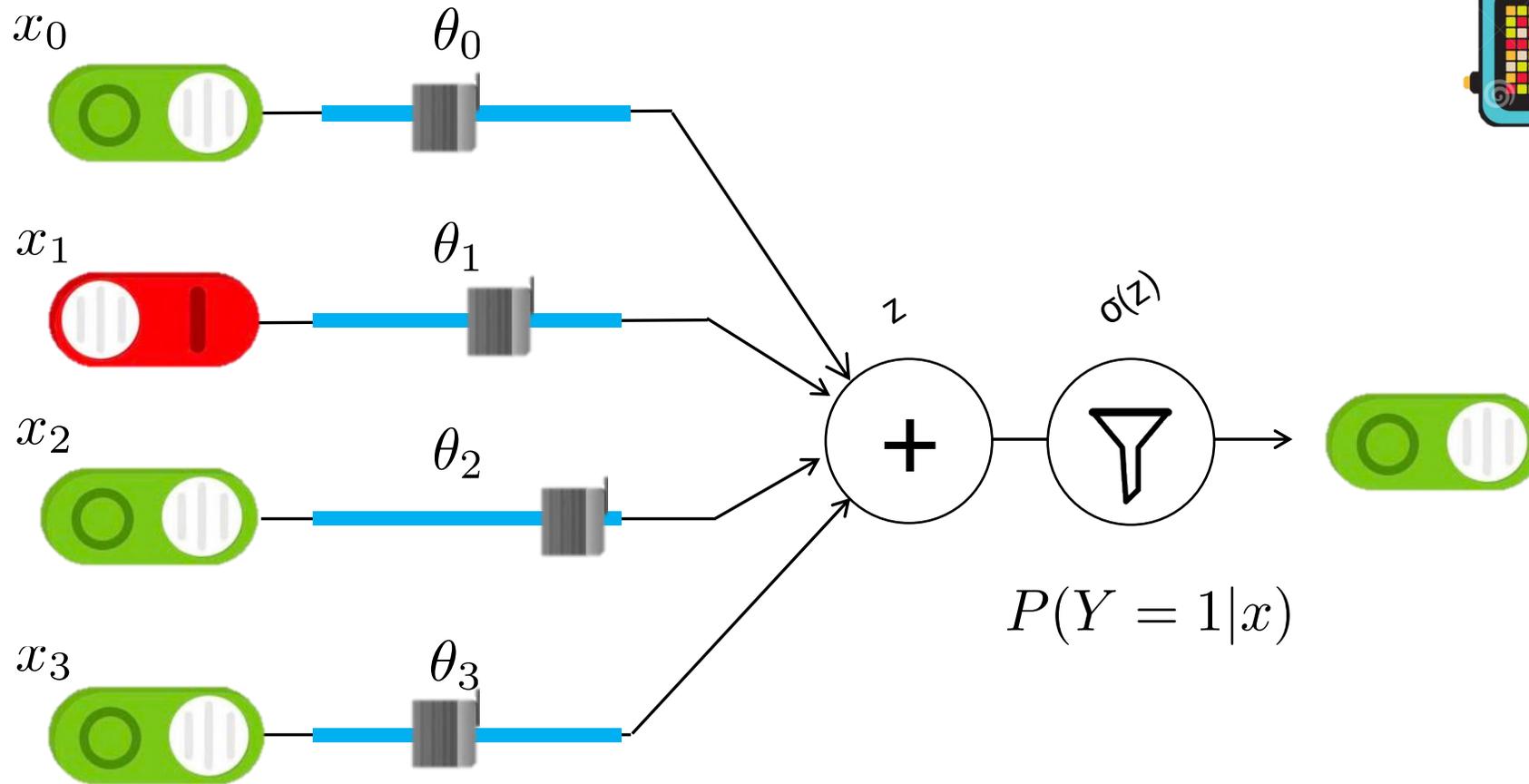


# Reinforcement 1

CS109, Stanford University

Review

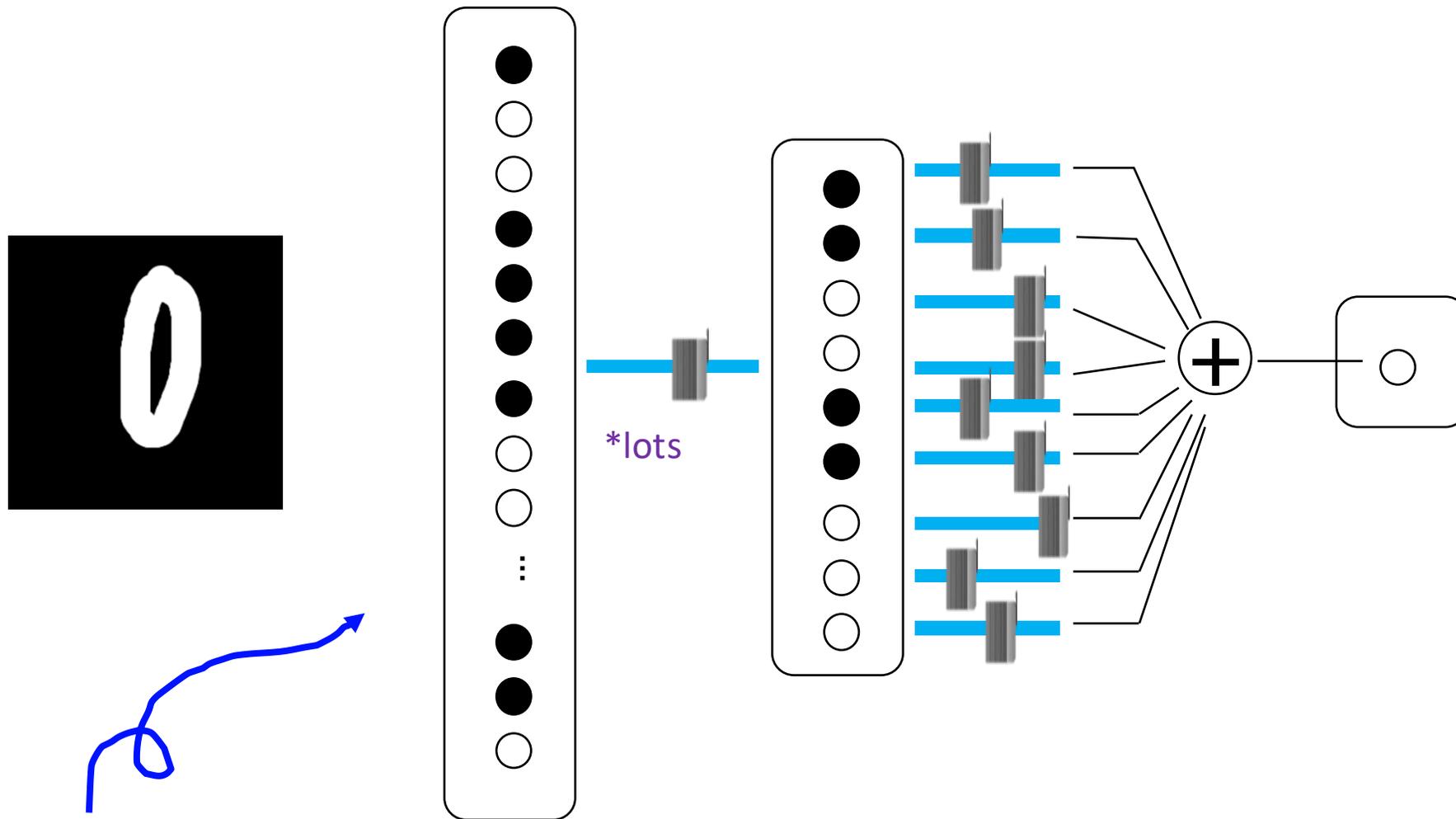
# Logistic Regression



$$P(Y = 1|\mathbf{X} = \mathbf{x}) = \sigma\left(\sum_i \theta_i x_i\right)$$

# Deep Learning

# We Can Put Neurons Together



These are the features for one training example.

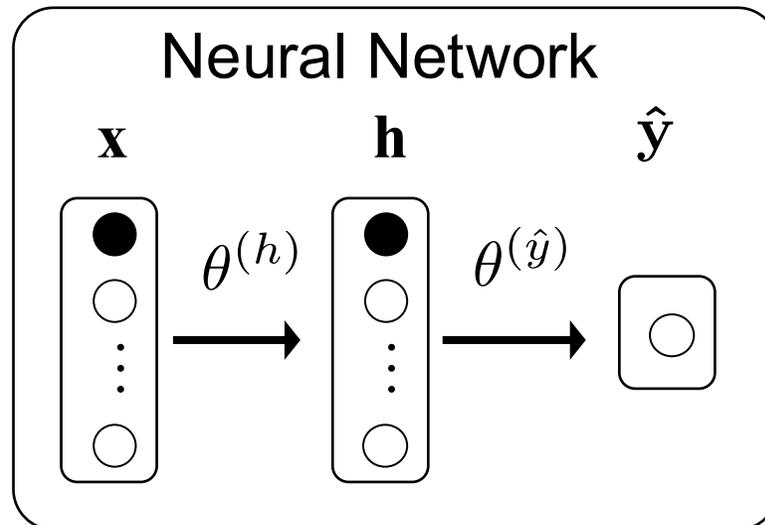
# Derivative Goals

Loss with respect to  
output layer params

$$\frac{\partial LL(\theta)}{\partial \theta_i^{(\hat{y})}}$$

Loss with respect to  
hidden layer params

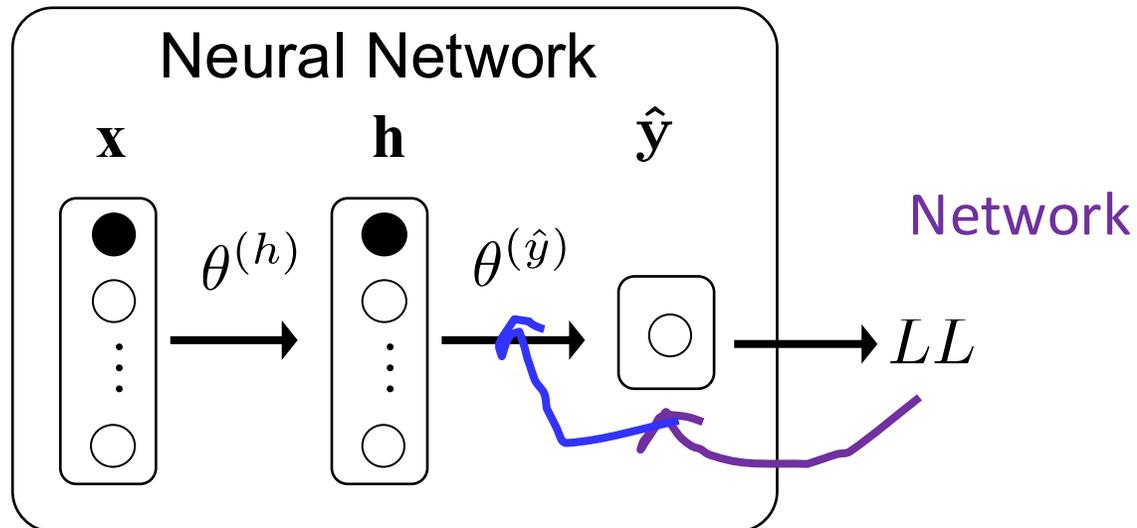
$$\frac{\partial LL(\theta)}{\partial \theta_{i,j}^{(h)}}$$



# Chain Rule Example 1

$$\frac{\partial LL(\theta)}{\partial \theta_i^{(\hat{y})}}$$

Goal



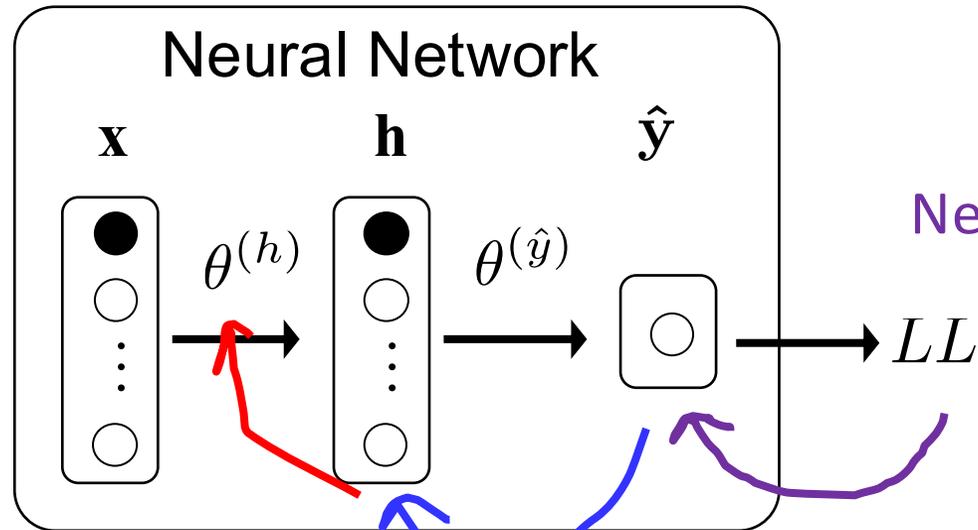
$$\frac{\partial LL(\theta)}{\partial \theta_i^{(\hat{y})}} = \frac{\partial LL}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial \theta_i^{(\hat{y})}}$$

Decomposition

# Chain Rule Example 2

$$\frac{\partial LL(\theta)}{\partial \theta_{i,j}^{(h)}}$$

Goal



Network

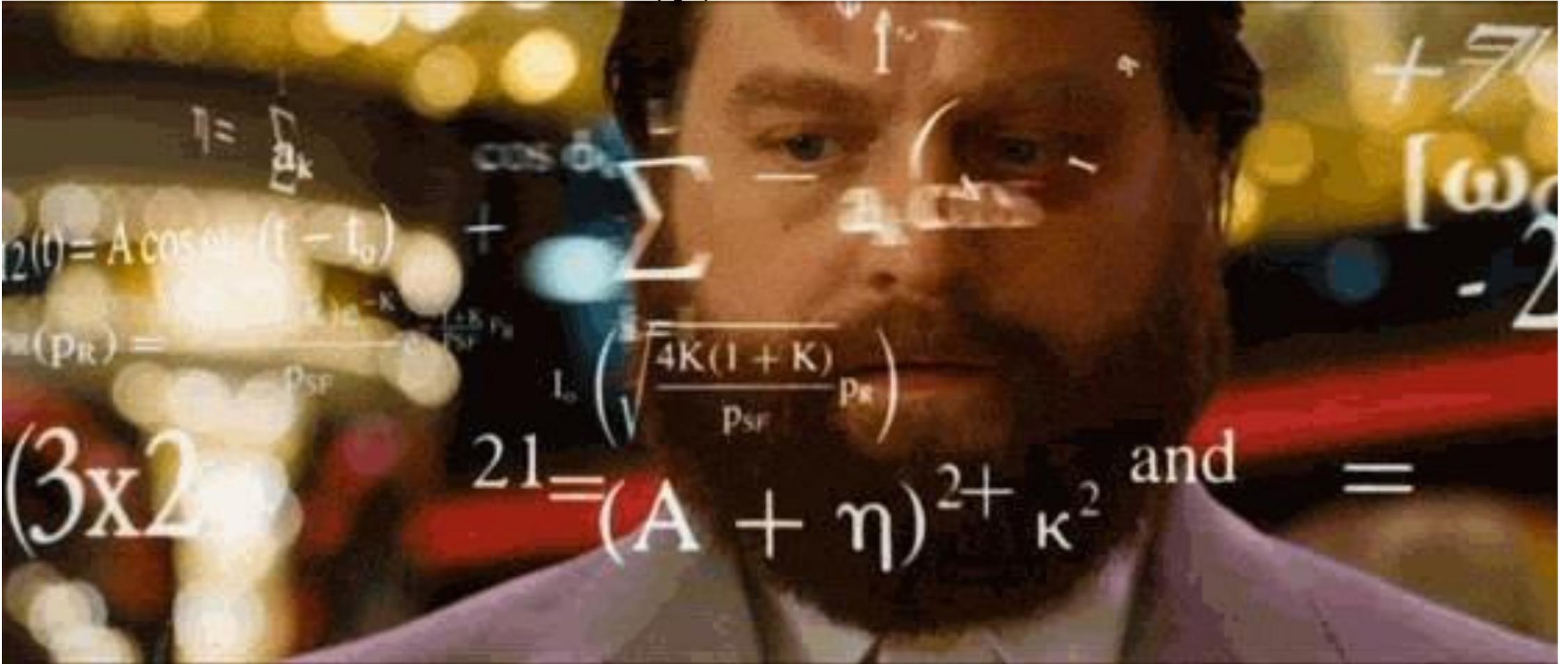
$$\frac{\partial LL(\theta)}{\partial \theta_{i,j}^{(h)}} = \frac{\partial LL}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial \mathbf{h}_j} \cdot \frac{\partial \mathbf{h}_j}{\partial \theta_{i,j}^{(h)}}$$

Decomposition

# Chain Rule Example 2

$$\frac{\partial LL(\theta)}{\partial \theta}$$

Goal



$$\frac{\partial \theta_{i,j}^{(h)}}{\partial \theta_{i,j}^{(h)}}$$

$$\frac{\partial \hat{y}}{\partial \hat{y}}$$

$$\frac{\partial \mathbf{h}_j}{\partial \mathbf{h}_j}$$

$$\frac{\partial \theta_{i,j}^{(h)}}{\partial \theta_{i,j}^{(h)}}$$

Decomposition

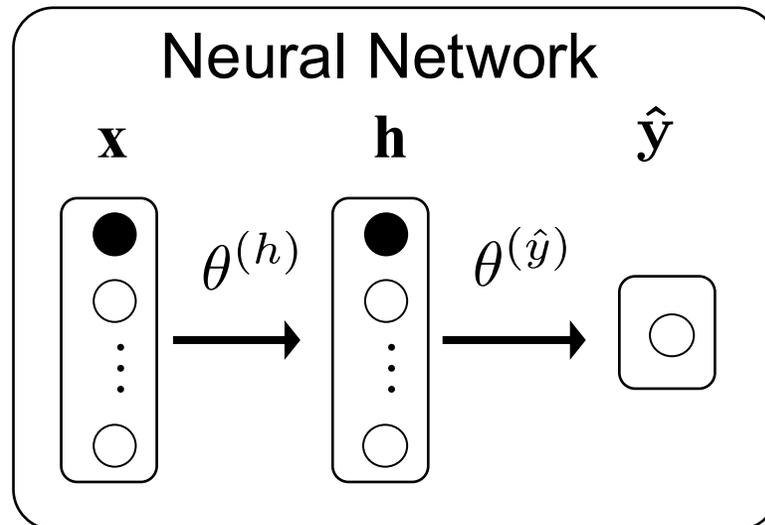
# Deep Learning

Loss with respect to  
output layer params

$$\frac{\partial LL(\theta)}{\partial \theta_i^{(\hat{y})}}$$

Loss with respect to  
hidden layer params

$$\frac{\partial LL(\theta)}{\partial \theta_{i,j}^{(h)}}$$



# Make it Simple

$$\frac{\partial LL(\theta)}{\partial \theta_i^{(\hat{y})}} = \text{[Yellow Box] - [Turtle]}$$

$$\text{[Yellow Box]} = \frac{y}{\hat{y}} - \frac{(1 - y)}{(1 - \hat{y})}$$

$$\text{[Turtle]} = \hat{y}[1 - \hat{y}] \cdot h_i$$

# Make it Simple

$$\frac{\partial LL(\theta)}{\partial \theta_{i,j}^{(h)}} = \begin{array}{|c|c|c|} \hline \img alt="Chest icon" data-bbox="444 171 526 317"/> & \img alt="Turtle icon" data-bbox="526 171 608 317"/> & \img alt="Croc icon" data-bbox="608 171 687 317"/> \\ \hline \end{array}$$

$$\begin{array}{|c|} \hline \img alt="Chest icon" data-bbox="344 354 426 506"/> \\ \hline \end{array} = \frac{y}{\hat{y}} - \frac{(1-y)}{(1-\hat{y})}$$

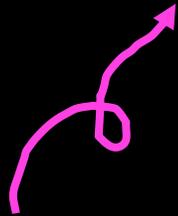
$$\begin{array}{|c|} \hline \img alt="Turtle icon" data-bbox="344 565 426 717"/> \\ \hline \end{array} = \hat{y}[1-\hat{y}]\theta_j^{(\hat{y})}$$

$$\begin{array}{|c|} \hline \img alt="Croc icon" data-bbox="351 754 433 908"/> \\ \hline \end{array} = \mathbf{h}_j[1-\mathbf{h}_j]\mathbf{x}_j$$



You will see this math in many future courses!

You will see this math in many future courses!



Backpropagation

Seeing and working through deep learning derivatives will help you better understand derivative for logistic regression.

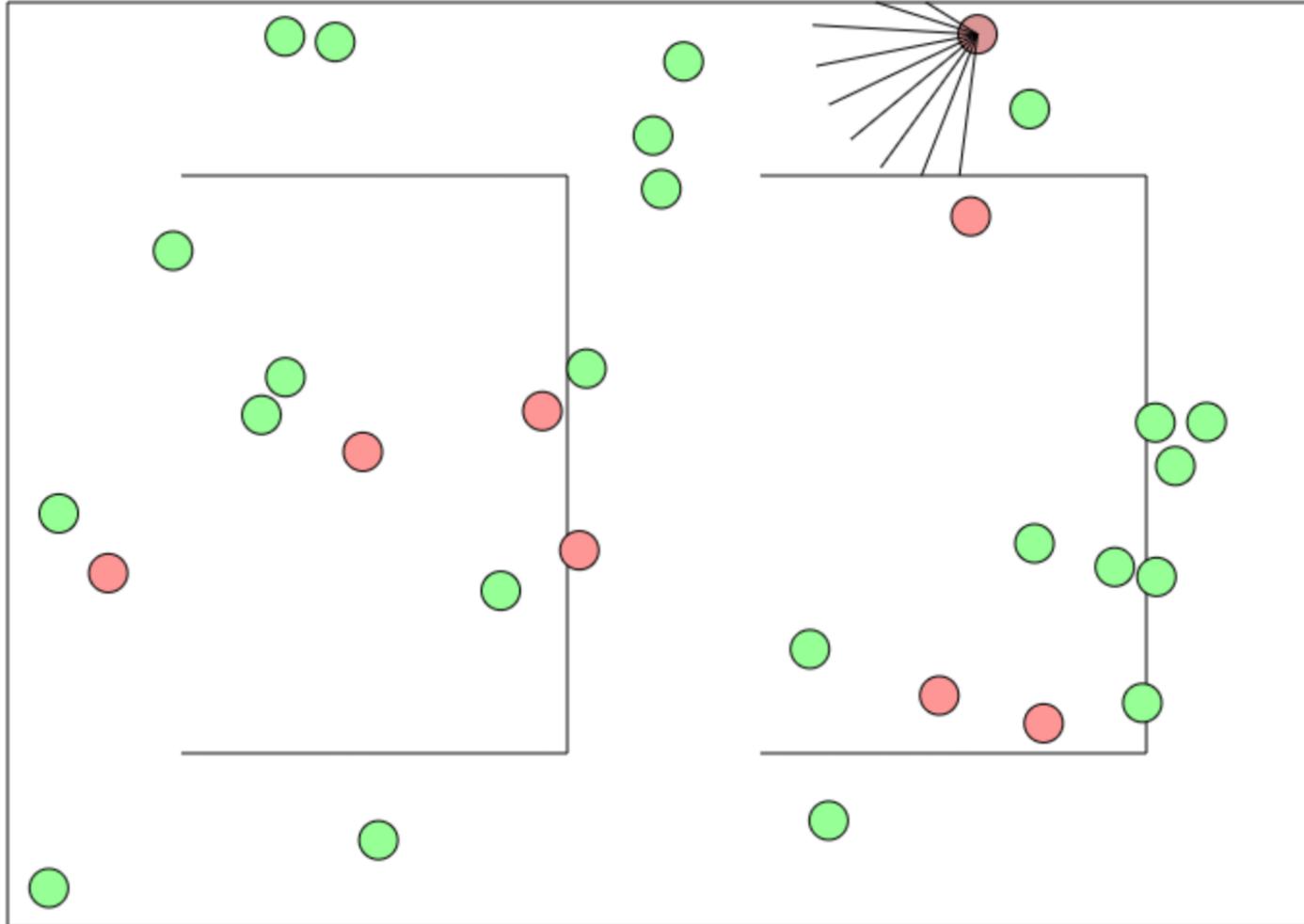
Need to know for CS109



But wait....

Is all of ML Classification?

# Lets start training a Critter



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

# Types of Machine Learning Tasks

---

Multi-Class  
Classification

Regression

Reinforcement  
Learning

Generation

# Types of Machine Learning Tasks

---

Multi-Class  
Classification

Regression

Reinforcement  
Learning

Generation

# Beyond Binary Classification

# Multiple Outputs

Draw your number here



0 1 2 3 4 5 6 7 8 9



X P Erase

Downsampled drawing:

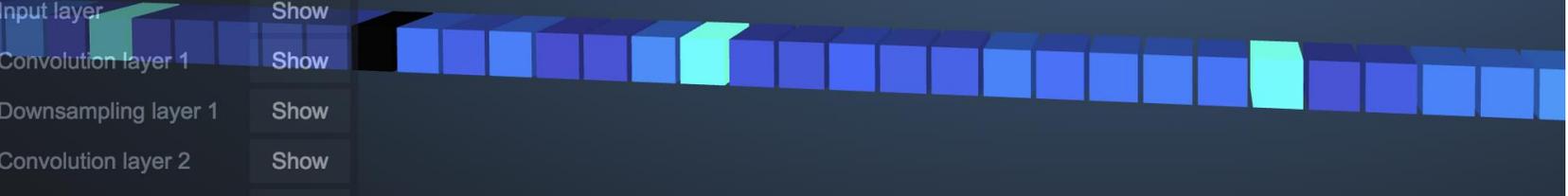
First guess: 3

Second guess: 3

8

Layer visibility

- Input layer Show
- Convolution layer 1 Show
- Downsampling layer 1 Show
- Convolution layer 2 Show



# The Categorical

Binary results (eg coin)

More than two outcomes (eg dice)

One experiment

Bernoulli

???

Many experiments

Binomial

Multinomial

# The Categorical

Binary results (eg coin)

More than two outcomes (eg dice)

One experiment

**Bernoulli**

**Categorical**

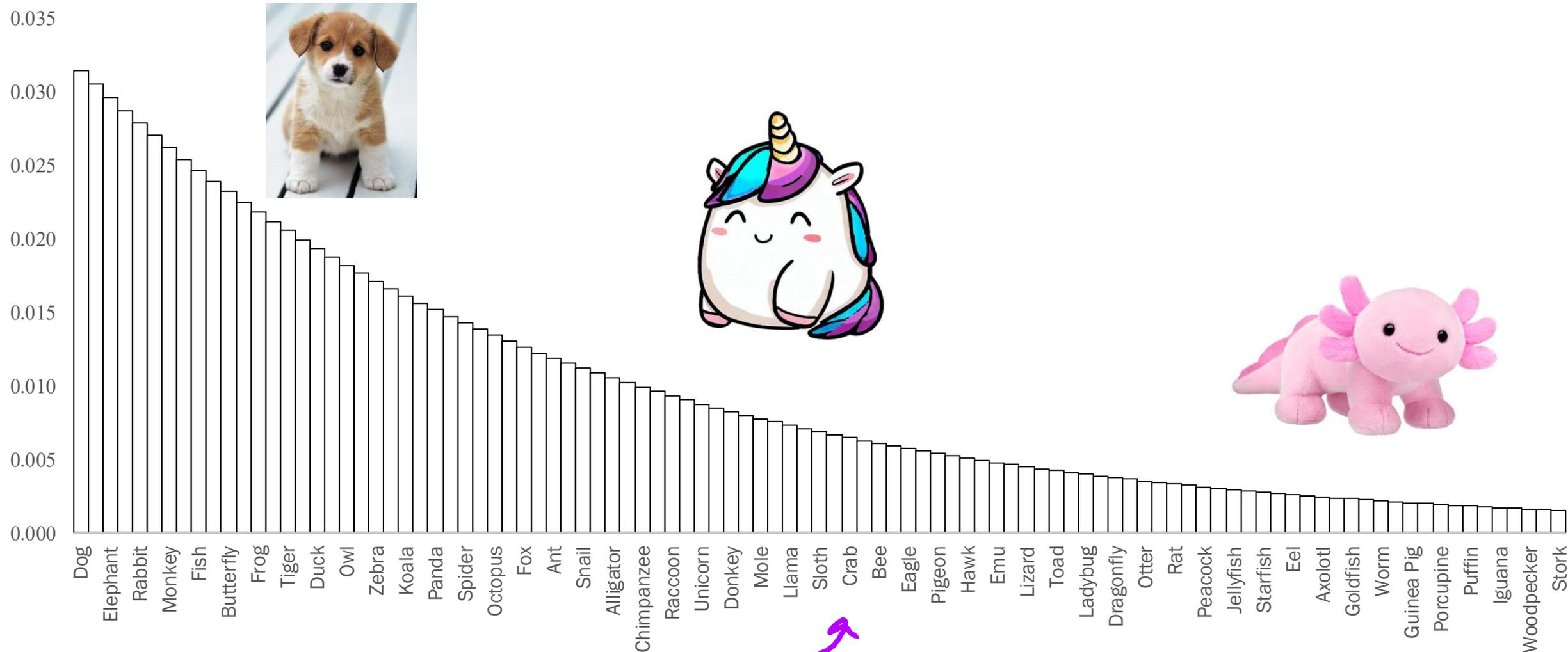
(or just Random Variable)

Many experiments

**Binomial**

**Multinomial**

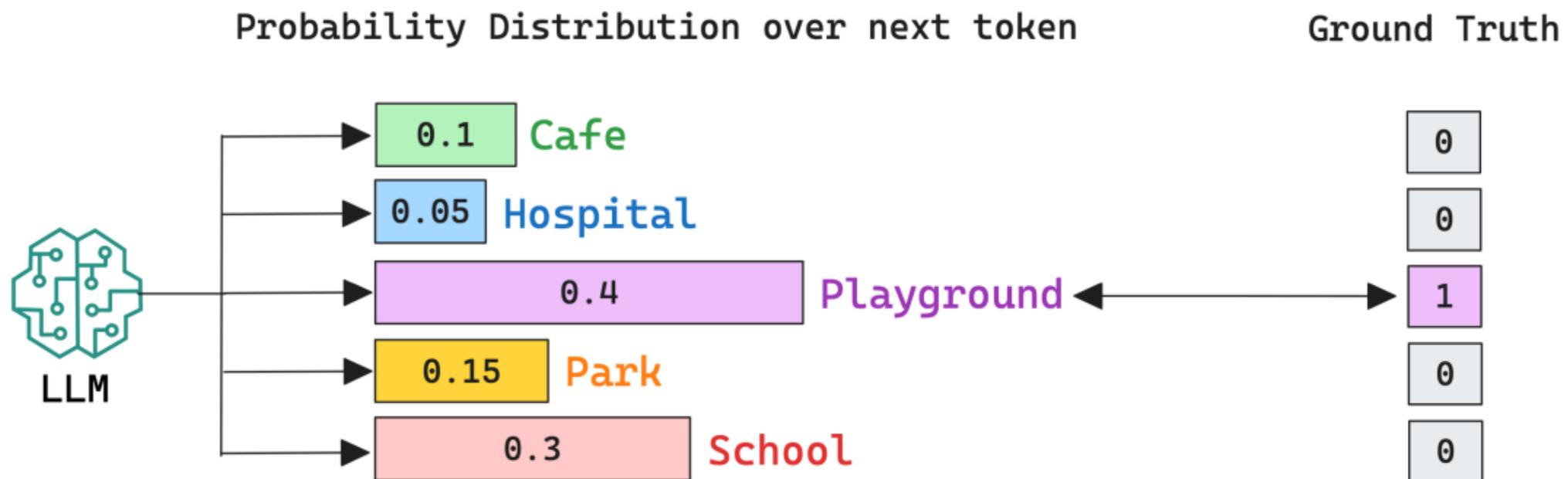
# Categorical Example from Class: Thinking of an Animal



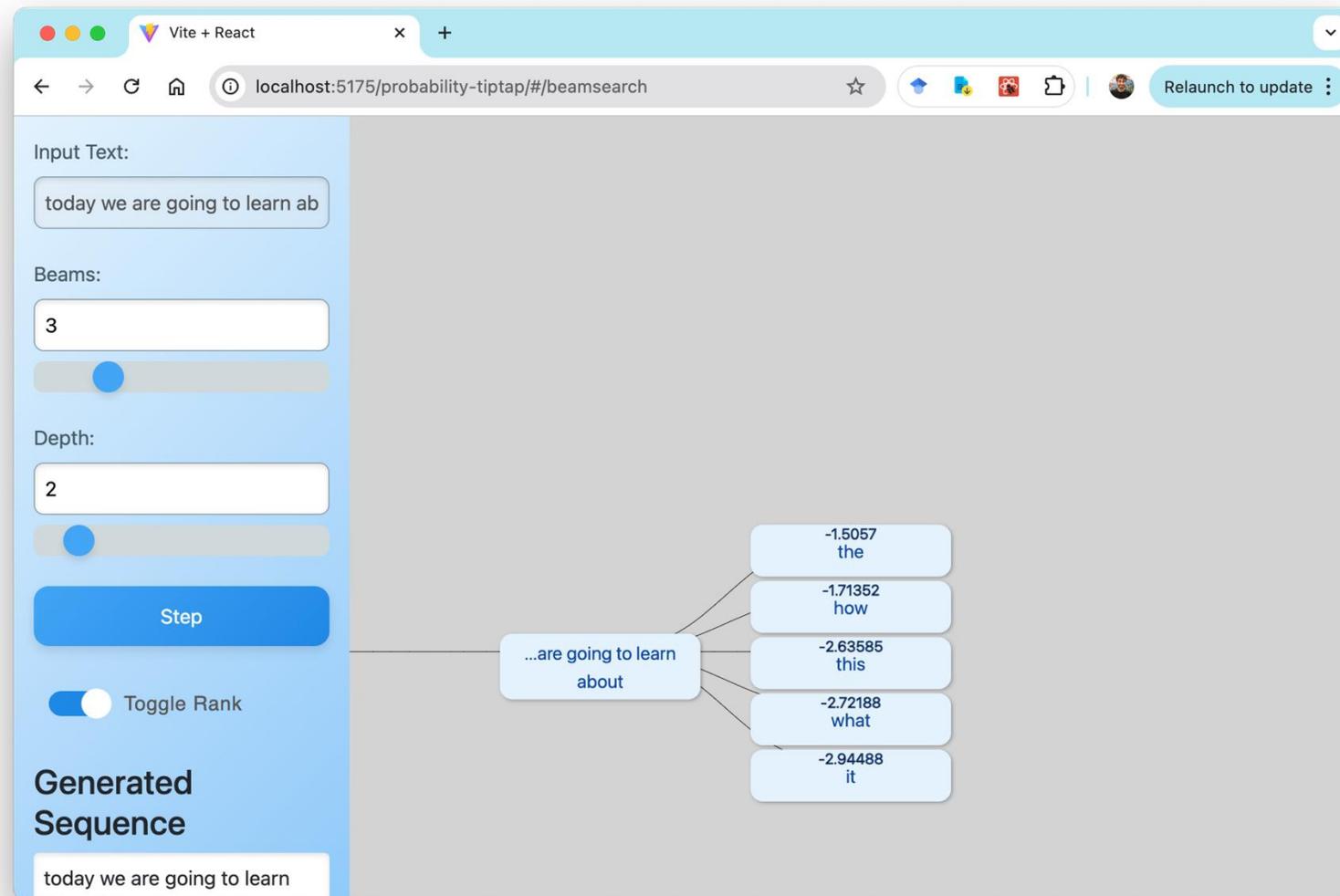
Notice that these are not numbers



# Output of an LLM is a Categorical for Next Token

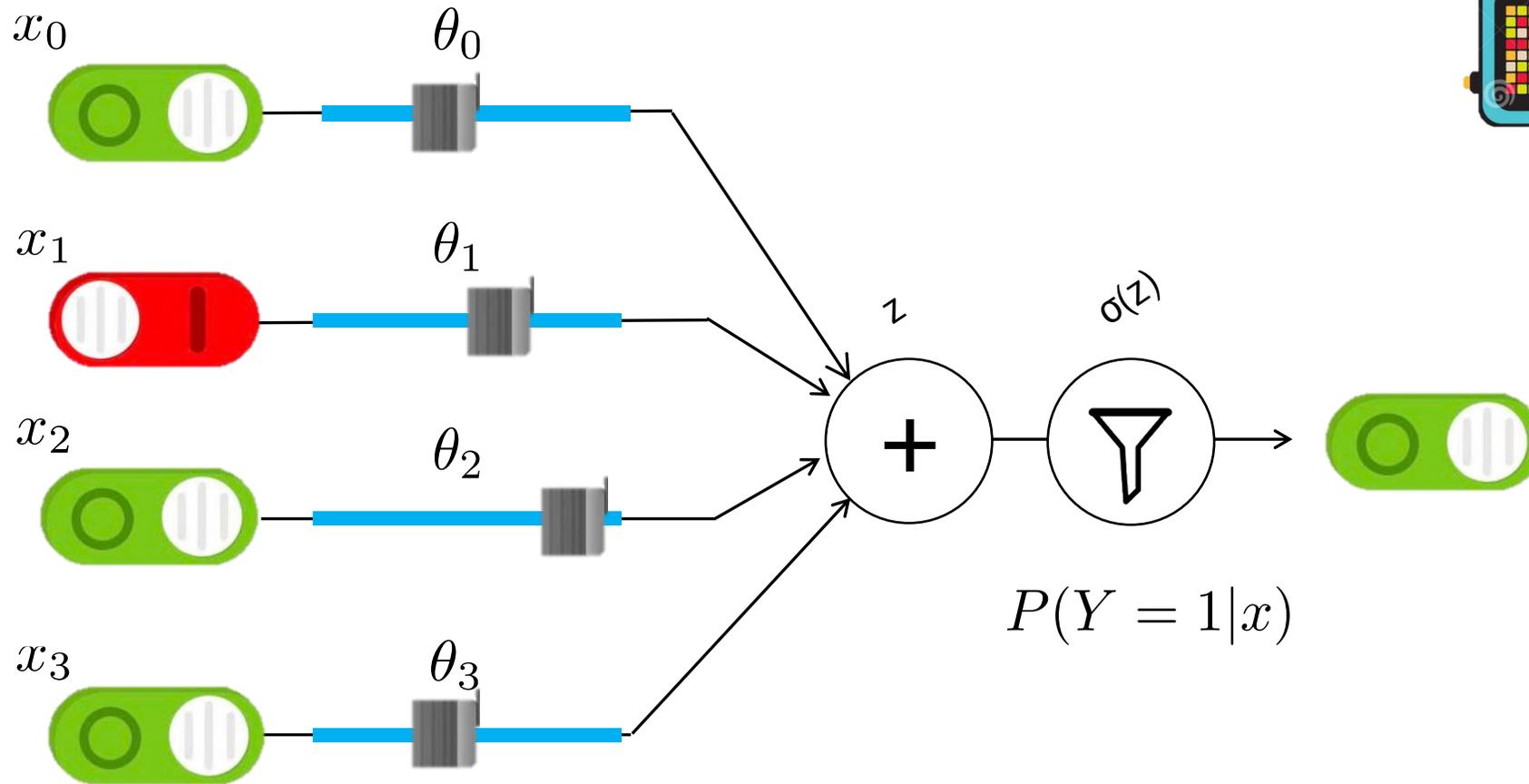
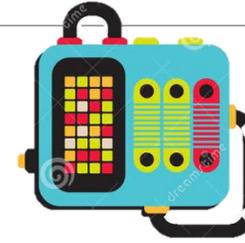


# Output of an LLM is a Categorical for Next Token



Visualization is thanks to Justin Blumencranz and Adam Boswell

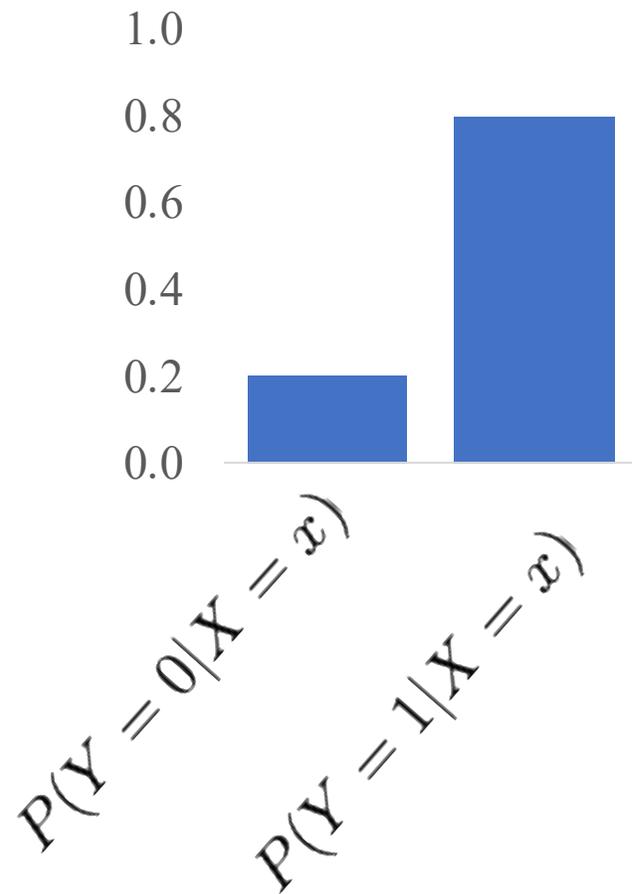
# Logistic Regression to Predict a Categorical?



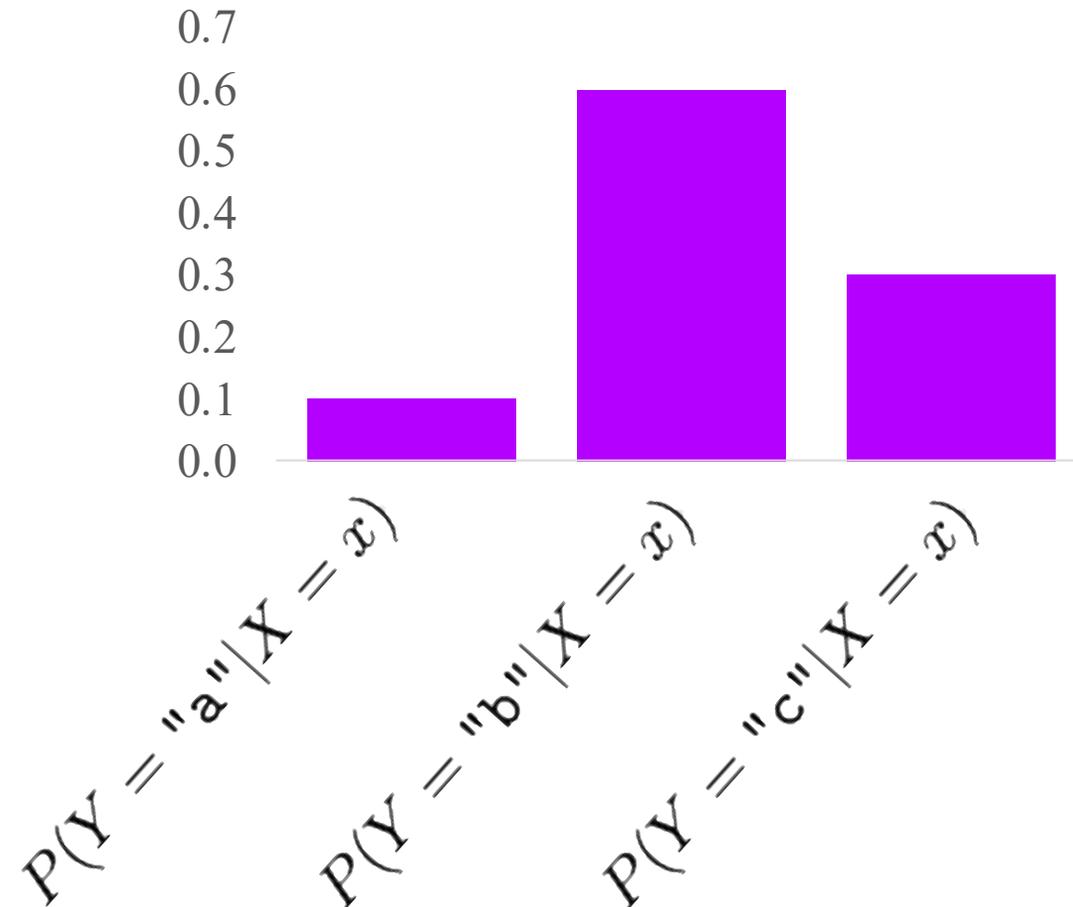
$$P(Y = 1 | \mathbf{X} = \mathbf{x}) = \sigma\left(\sum_i \theta_i x_i\right)$$

# Logistic Regression to Predict a Categorical?

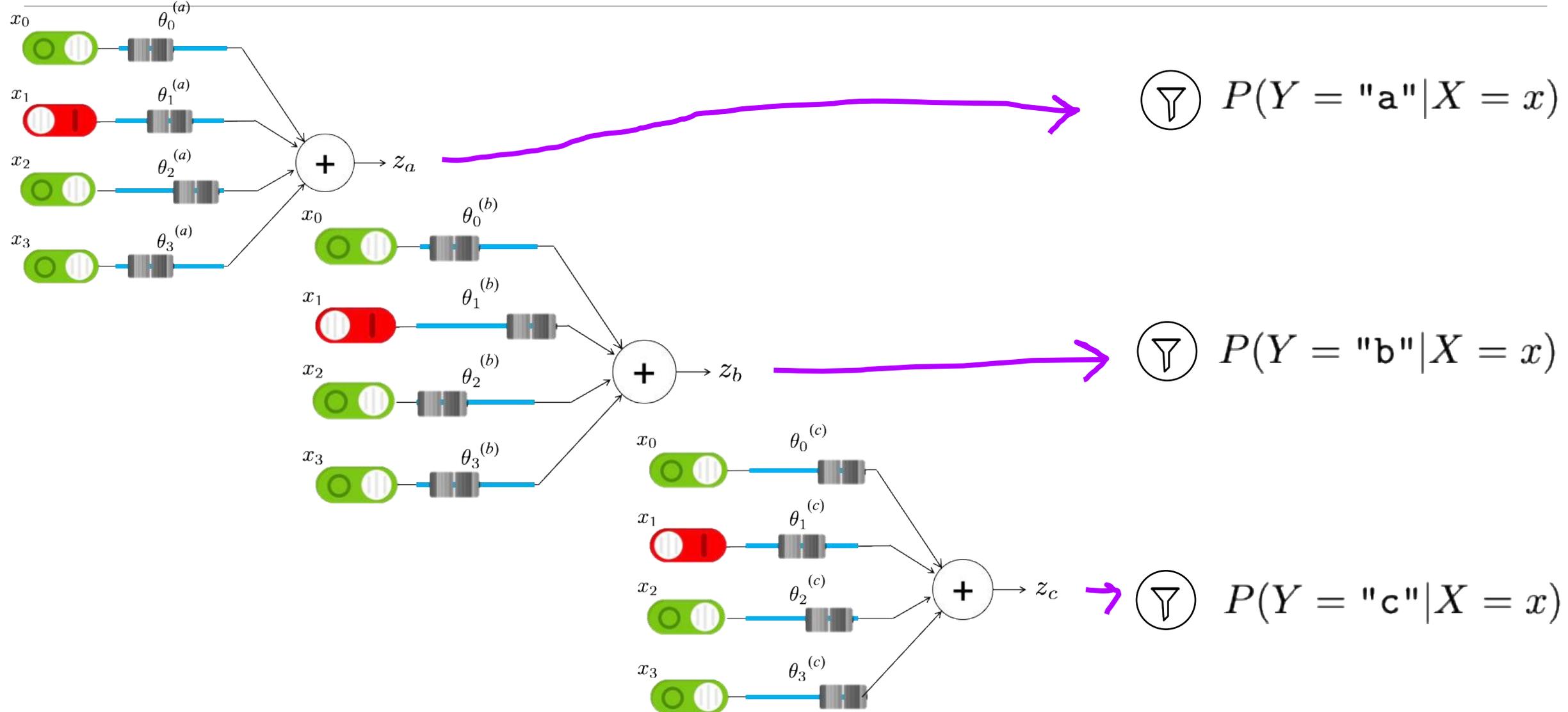
Standard Logistic Regression



Multi-Class Logistic Regression



# Logistic Regression to Predict a Categorical



# Types of Machine Learning Tasks

---

Multi-Class  
Classification

Regression

Today

Reinforcement  
Learning

Generation

# Regression: Predicting Real Numbers

	Opposing team ELO	Points in last game	At Home?	Output
				 # Points
Game 1	84	105	1	120
Game 2	90	102	0	95
		⋮		⋮
Game $n$	74	120	0	115

# Same Notation for Training Data

Training Data: assignments all random variables  $\mathbf{X}$  and  $Y$

Assume IID data:

*n training datapoints*

$$(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \dots, (\mathbf{x}^{(n)}, y^{(n)})$$

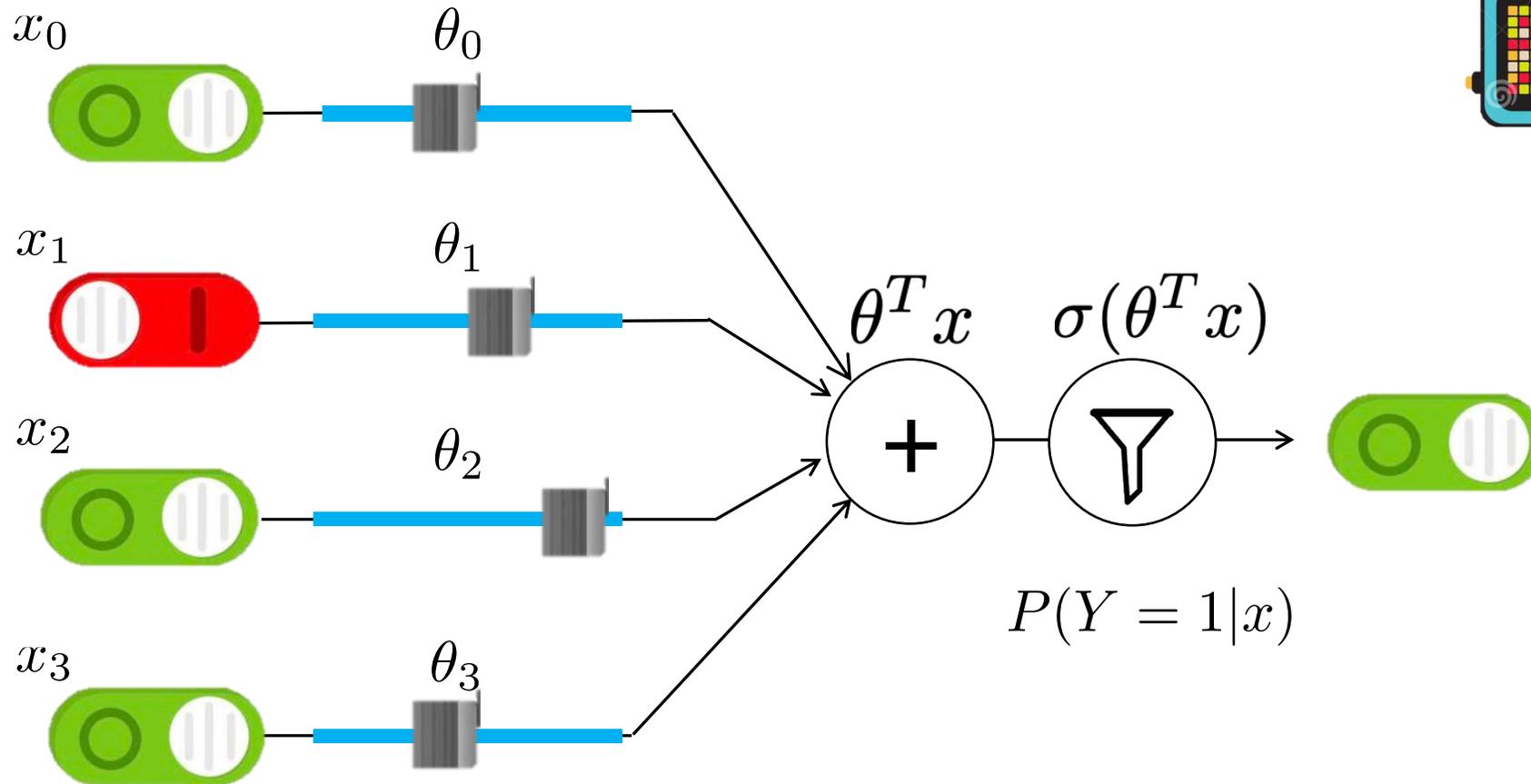
$$m = |\mathbf{x}^{(i)}|$$

Each datapoint has  $m$  features and a single output

# Regression: Predicting Real Numbers

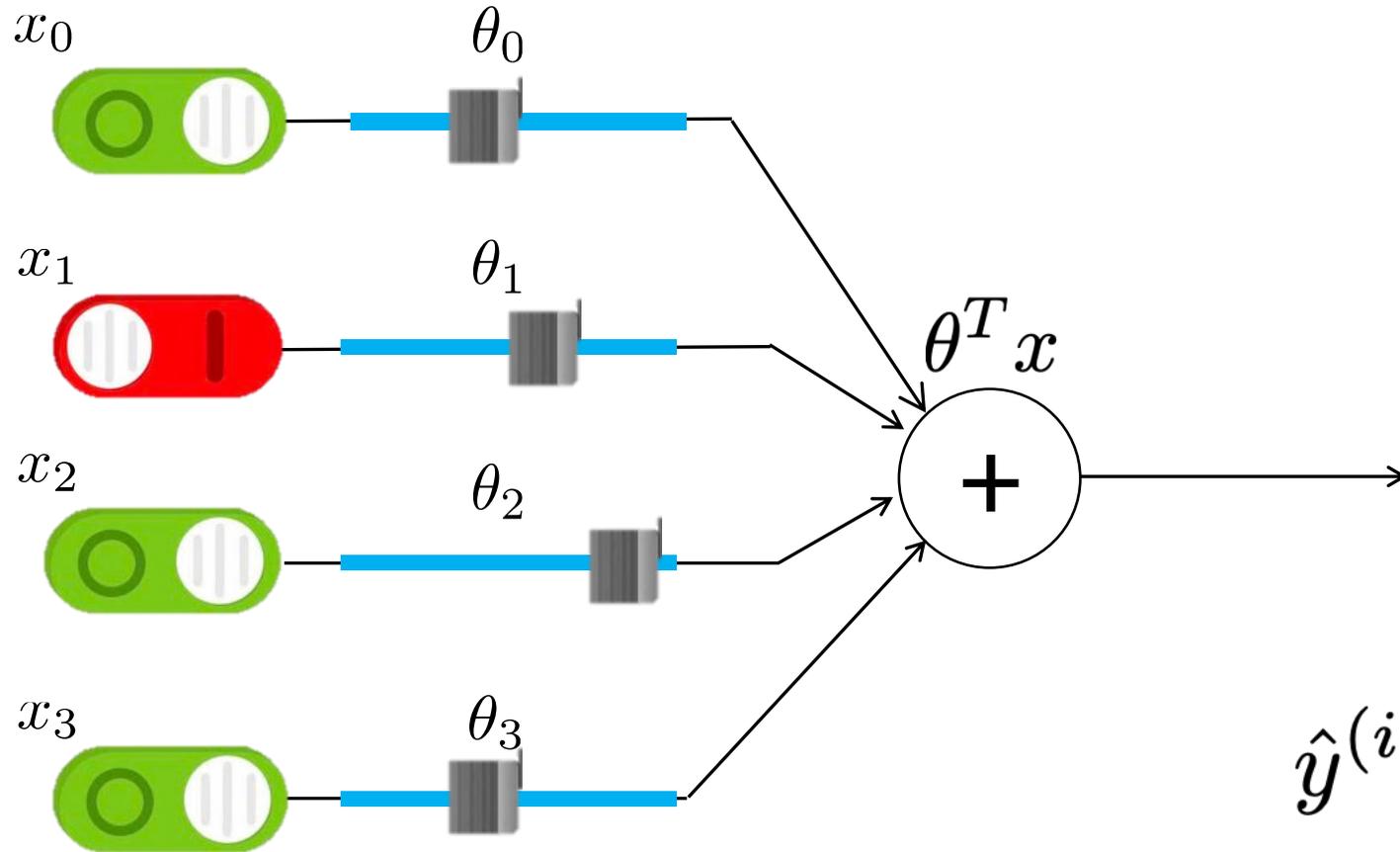
	Opposing team ELO	Points in last game	At Home?	Output
				 # Points
Game 1	84	105	1	120
Game 2	90	102	0	95
		⋮		⋮
Game $n$	74	120	0	115

# Logistic Regression



$$P(Y = 1 | \mathbf{X} = \mathbf{x}) = \sigma\left(\sum_i \theta_i x_i\right)$$

# Linear Regression



$$\hat{y}^{(i)} = \theta^T x^{(i)} + Z$$

$$Z \sim N(0, \sigma^2)$$

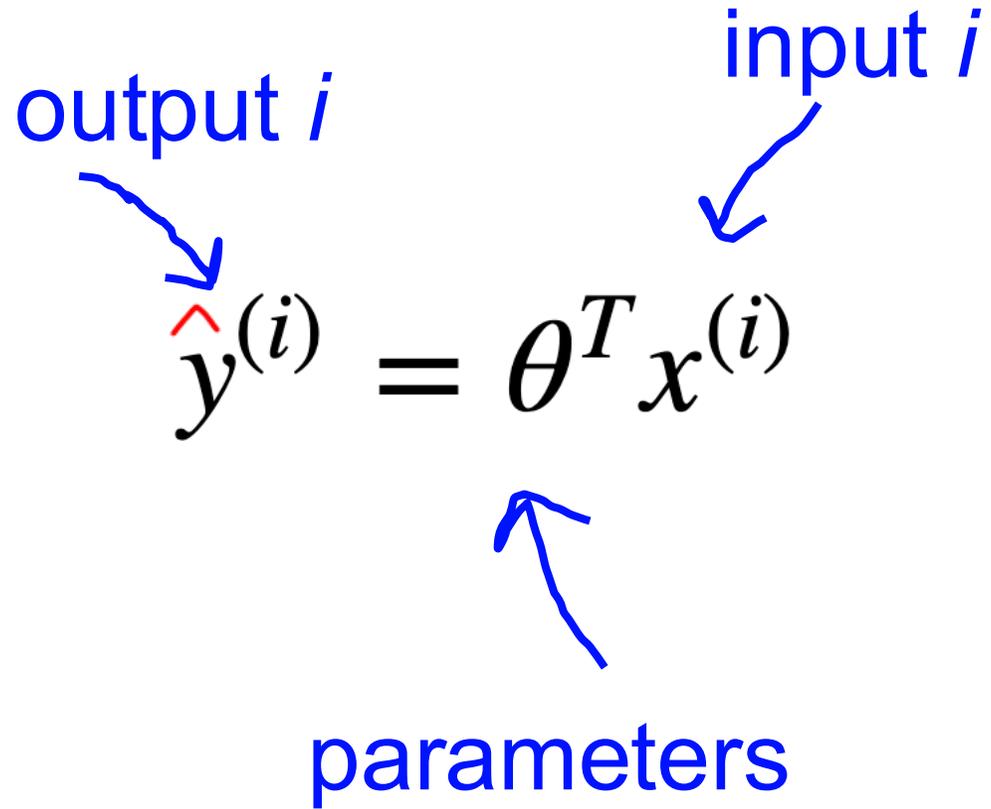
# Linear Regression Model

output  $i$

input  $i$

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

parameters

The diagram shows the linear regression model equation  $\hat{y}^{(i)} = \theta^T x^{(i)}$ . The predicted output  $\hat{y}^{(i)}$  is labeled as 'output  $i$ ' with a blue arrow pointing to the red-hatted  $y$ . The input vector  $x^{(i)}$  is labeled as 'input  $i$ ' with a blue arrow pointing to the  $x$ . The parameter vector  $\theta$  is labeled as 'parameters' with a blue arrow pointing to the  $\theta$ .

# Linear Regression Model

The diagram illustrates the linear regression model equation  $\hat{y}^{(i)} = \theta^T x^{(i)} + Z$ . The equation is centered on the page. Four blue arrows point from text labels to parts of the equation: one from 'output *i*' to the predicted output  $\hat{y}^{(i)}$ , one from 'input *i*' to the input vector  $x^{(i)}$ , one from 'parameters' to the parameter vector  $\theta$ , and one from 'random noise' to the noise term  $Z$ . The predicted output  $\hat{y}^{(i)}$  has a red hat symbol above the  $y$ .

$$\hat{y}^{(i)} = \theta^T x^{(i)} + Z$$

output *i*

input *i*

parameters

random noise

# Linear Regression Model

output  $i$

input  $i$

Noise is  $N$  with mean 0

$$\hat{y}^{(i)} = \theta^T x^{(i)} + Z$$

$Z \sim N(0, \sigma^2)$

parameters

random noise

The diagram shows the linear regression model equation  $\hat{y}^{(i)} = \theta^T x^{(i)} + Z$  with several annotations. A blue arrow points from the text 'output i' to the predicted output  $\hat{y}^{(i)}$ . Another blue arrow points from 'input i' to the input vector  $x^{(i)}$ . A third blue arrow points from 'parameters' to the parameter vector  $\theta$ . A fourth blue arrow points from 'random noise' to the noise term  $Z$ . To the right of the equation, the text 'Noise is N with mean 0' is written in blue, followed by the mathematical expression  $Z \sim N(0, \sigma^2)$ .

# Linear Regression Model

output  $i$

noise

noise is  $N$  with mean 0

$$\hat{y}^{(i)} = \theta^T x^{(i)} + Z \quad Z \sim N(0, \sigma^2)$$

## 1. Linear Transform

If  $X$  is a Normal such that  $X \sim N(\mu, \sigma^2)$  and  $Y$  is a linear transform of  $X$  such that  $Y = aX + b$  then  $Y$  is also a Normal where:

$$Y \sim N(a\mu + b, a^2\sigma^2)$$

# Linear Regression Model

output  $i$

noise

noise is  $N$  with mean 0

$$\hat{y}^{(i)} = \underline{\theta}^T x^{(i)} + Z \quad Z \sim N(0, \sigma^2)$$

Output is normal too:

$$\hat{y}^{(i)} \sim N(\theta^T x^{(i)}, \underline{\sigma}^2)$$

# Log Likelihood

Assume:  $\hat{y}^{(i)} \sim N(\theta^T x^{(i)}, \sigma^2)$

Data:  $(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \dots, (\mathbf{x}^{(n)}, y^{(n)})$

$$LL(\theta) = \sum_{i=1}^n \log [f(y^{(i)})]$$

$$= \sum_{i=1}^n \log \left[ \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2} \left( \frac{y^{(i)} - \mu}{\sigma} \right)^2} \right]$$

Normal distribution PDF

$$= \sum_{i=1}^n \log \left[ \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2} \left( \frac{y^{(i)} - \theta^T x^{(i)}}{\sigma} \right)^2} \right]$$

Substitute in the mean

$$= \sum_{i=1}^n \log \left[ \frac{1}{\sigma\sqrt{2\pi}} \right] - \frac{1}{2} \left( \frac{y^{(i)} - \theta^T x^{(i)}}{\sigma} \right)^2$$

Apply the log

# Optimization

Log likelihood:  $LL(\theta) = \sum_{i=1}^n \log \left[ \frac{1}{\sigma\sqrt{2\pi}} \right] - \frac{1}{2} \left( \frac{y^{(i)} - \theta^T x^{(i)}}{\sigma} \right)^2$

$$\operatorname{argmax}_{\theta} LL(\theta) = \operatorname{argmax}_{\theta} \sum_{i=1}^n \log \left[ \frac{1}{\sigma\sqrt{2\pi}} \right] - \frac{1}{2} \left( \frac{y^{(i)} - \theta^T x^{(i)}}{\sigma} \right)^2$$

$$= \operatorname{argmax}_{\theta} - \sum_{i=1}^n \frac{1}{2} \left( \frac{y^{(i)} - \theta^T x^{(i)}}{\sigma} \right)^2$$

Simplify

$$= \operatorname{argmax}_{\theta} - \sum_{i=1}^n \left( y^{(i)} - \theta^T x^{(i)} \right)^2$$

Simplify

Hey it's the sum of squared errors!

# Linear Regression with one Input



# Derivative is Necessary for Gradient Ascent

---

$$\frac{\partial LL(\theta)}{\partial \theta_j} = - \sum_{i=1}^n \frac{\partial}{\partial \theta_j} \left( y^{(i)} - \theta^T x^{(i)} \right)^2$$

Derivative of a sum

$$= - \sum_{i=1}^n 2 \left( y^{(i)} - \theta^T x^{(i)} \right) (-x_j^{(i)})$$

Chain rule

$$= \sum_{i=1}^n 2 \left( y^{(i)} - \theta^T x^{(i)} \right) \cdot x_j^{(i)}$$

Simplify

# Types of Machine Learning Tasks

---

Multi-Class  
Classification

Regression

Reinforcement  
Learning

Generation

# Types of Machine Learning Tasks

---

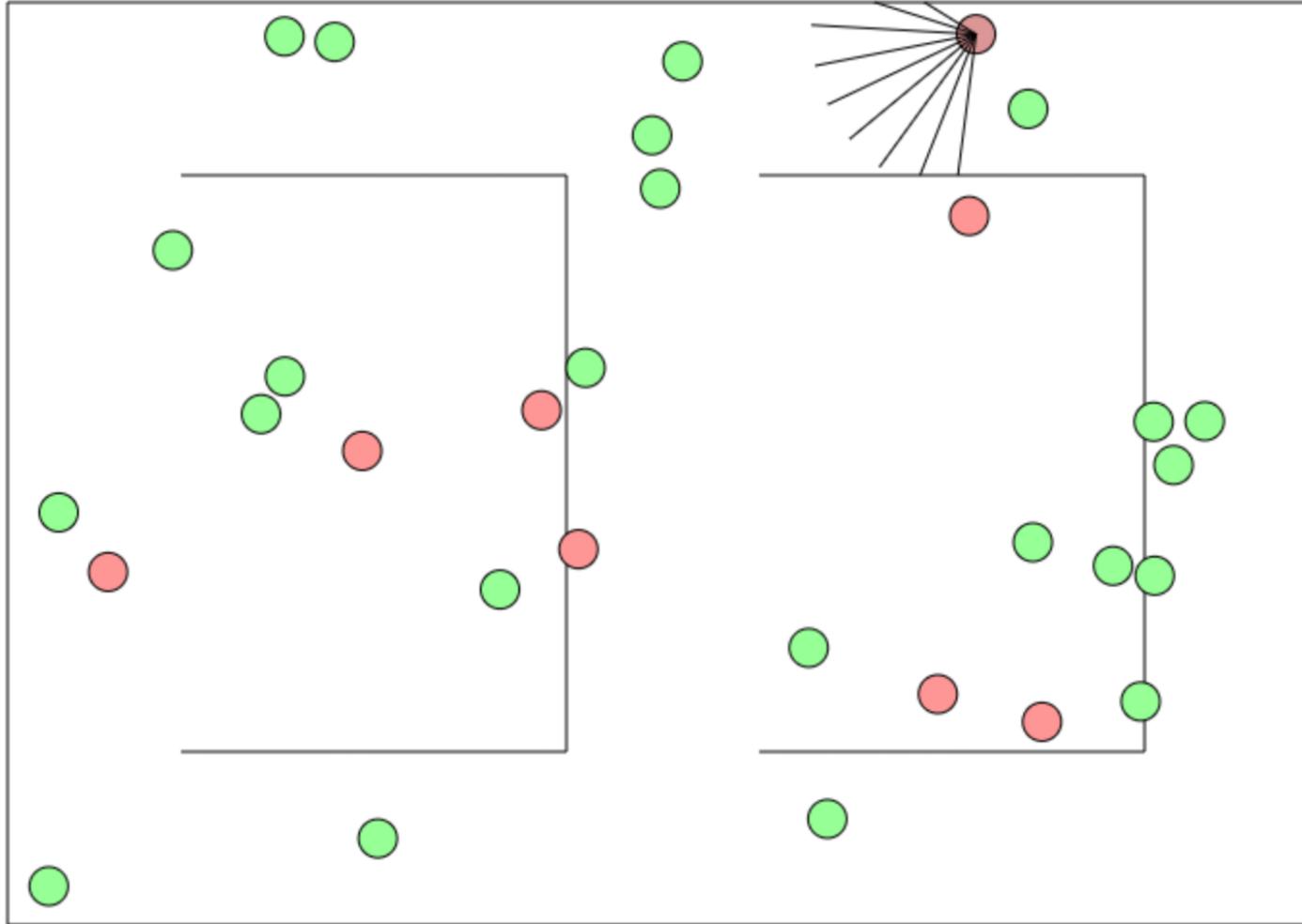
Multi-Class  
Classification

Regression

Reinforcement  
Learning

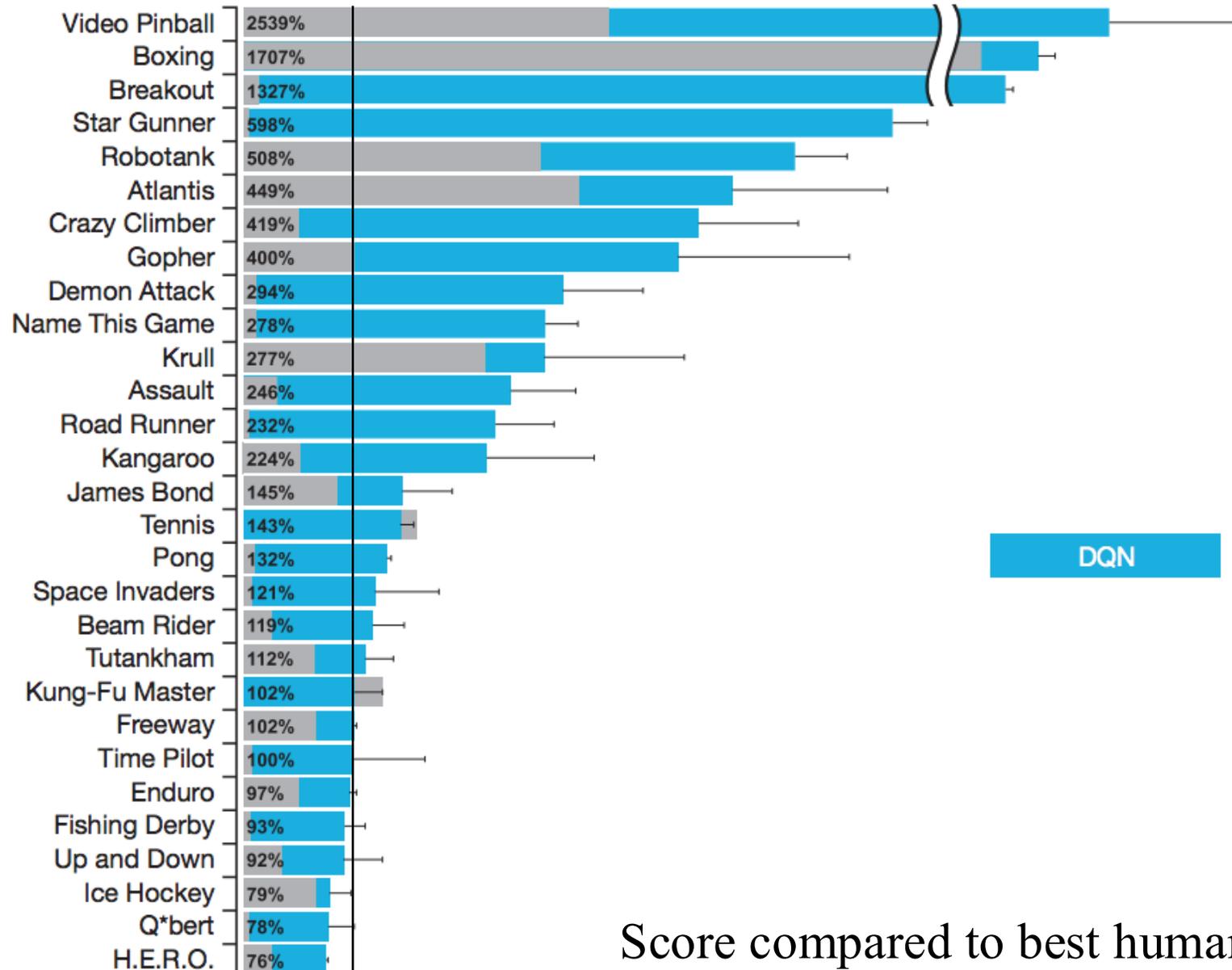
Generation

# Deep Reinforcement Learning



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

# Deep Mind Atari Games

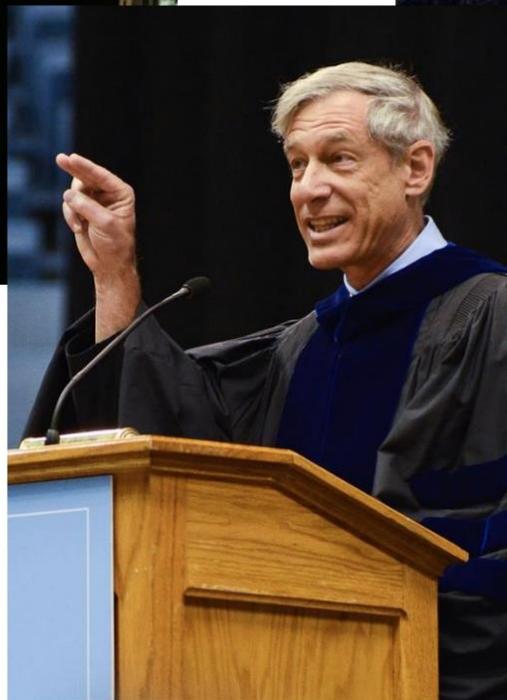


Score compared to best human

Review

Night Sight

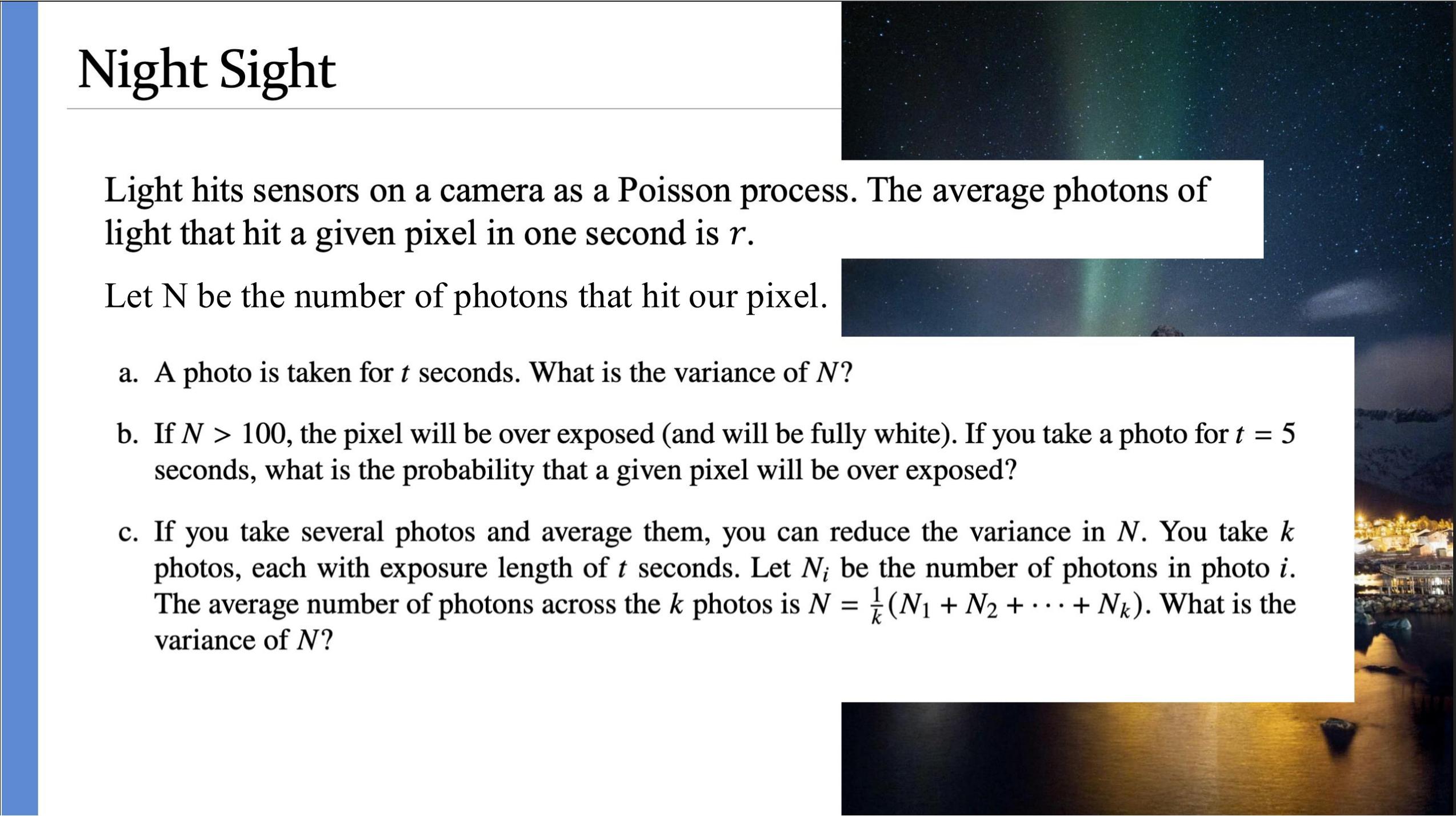
# Night Sight



Mark Levoy, Stanford Emeritus Professor

<https://static.googleusercontent.com/media/hdrplusdata.org/en//hdrplus.pdf>

# Night Sight



Light hits sensors on a camera as a Poisson process. The average photons of light that hit a given pixel in one second is  $r$ .

Let  $N$  be the number of photons that hit our pixel.

- A photo is taken for  $t$  seconds. What is the variance of  $N$ ?
- If  $N > 100$ , the pixel will be over exposed (and will be fully white). If you take a photo for  $t = 5$  seconds, what is the probability that a given pixel will be over exposed?
- If you take several photos and average them, you can reduce the variance in  $N$ . You take  $k$  photos, each with exposure length of  $t$  seconds. Let  $N_i$  be the number of photons in photo  $i$ . The average number of photons across the  $k$  photos is  $N = \frac{1}{k}(N_1 + N_2 + \dots + N_k)$ . What is the variance of  $N$ ?

Serendipity

Let it find you.

# SERENDIPITY

the effect by which one accidentally stumbles upon something truly wonderful, especially while looking for something entirely unrelated.





**WHEN YOU MEET YOUR BEST FRIEND**

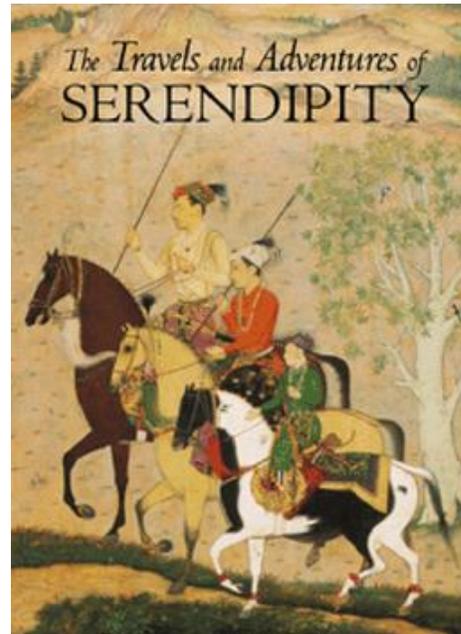
Somewhere you didn't expect to.



# Serendipity

---

- Say the population of Stanford is 17,000 people
  - You are friends with 100
  - Walk into a room, see 450 random people.
  - What is the probability that you see someone you know?
  - Assume you are equally likely to see each person at Stanford





Many times it is easier to  
calculate  $P(E^C)$  .



# Wisdom of the Crowds

# Wisdom of the Crowds

There are two answers for each audience member to choose from, a Correct answer and an Incorrect answer.

- 10% of the audience are knowledgeable about the problem (call them experts). An expert votes for the Correct answer with a probability of 0.7, otherwise they vote for the Incorrect answer.
- 90% of the audience are not knowledgeable (call them non-experts). A non-expert votes randomly with equal likelihood between the Correct answer and the Incorrect answer.



In 1999, what animal was taken off the U.S. Endangered species list after 29 years?

**A:**

**B:** Peregrine Falcon

**C:** Humpback Whale

**D:**

There are two answers for each audience member to choose from, a Correct answer and an Incorrect answer.

- 10% of the audience are knowledgeable about the problem (call them experts). An expert votes for the Correct answer with a probability of 0.7, otherwise they vote for the Incorrect answer.
  - 90% of the audience are not knowledgeable (call them non-experts). A non-expert votes randomly with equal likelihood between the Correct answer and the Incorrect answer.
- a. What is the probability that exactly  $k$  of the experts vote for the Correct answer? You may assume that  $k$  is a number between 0 and 20 inclusive.
  - b. If exactly  $k$  of the experts vote for the Correct answer, what is the probability that the Correct answer will get at least 101 votes? (hint: the Correct answer needs at least  $101 - k$  more votes from the non-experts).
  - c. Write an expression for the exact probability that the Correct answer will get at least 101 votes.
  - d. Use an approximation to estimate the probability that the Correct answer gets at least 101 votes. You may leave your answer in terms of roots and/or values that could be looked up from the  $\phi$  table. For full credit your approximation calculation should *not* include a summation or integral.

# Quant Interview

# Quant Interview

---



100 cupcakes.

Scenario 1: 51 are blue.

Scenario 2: 49 are blue.

a) You look at one cupcake. It is blue.

b) You look at three cupcakes (with replacement). Two are blue.

# Quant Interview - Bonus

---



100 cupcakes.

Scenario 1: 51 are blue.

Scenario 2: 49 are blue.

a) You look at one cupcake. It is blue.

b) You look at three cupcakes (with replacement). Two are blue.

c) You look at three cupcakes (without replacement). Two are blue.

# Quant Interview: Bonus

---

- c. A hypergeometric is a random variable for the number of successes if you remove items **without** replacement from a fixed population. If  $X \sim \text{Hypergeom}(t, k, n)$ , the PMF is

$$P(X = x) = \frac{\binom{k}{x} \binom{t-k}{n-x}}{\binom{t}{n}}$$

Where:

- $t$  is the population size,
- $k$  is the number of “success” items in the population,
- $n$  is the number of draws (without replacement),
- $X$  counts the number of successes observed in those  $n$  draws.

Three cupcakes are drawn without replacement and two are blue. What is the probability that the majority were blue?