

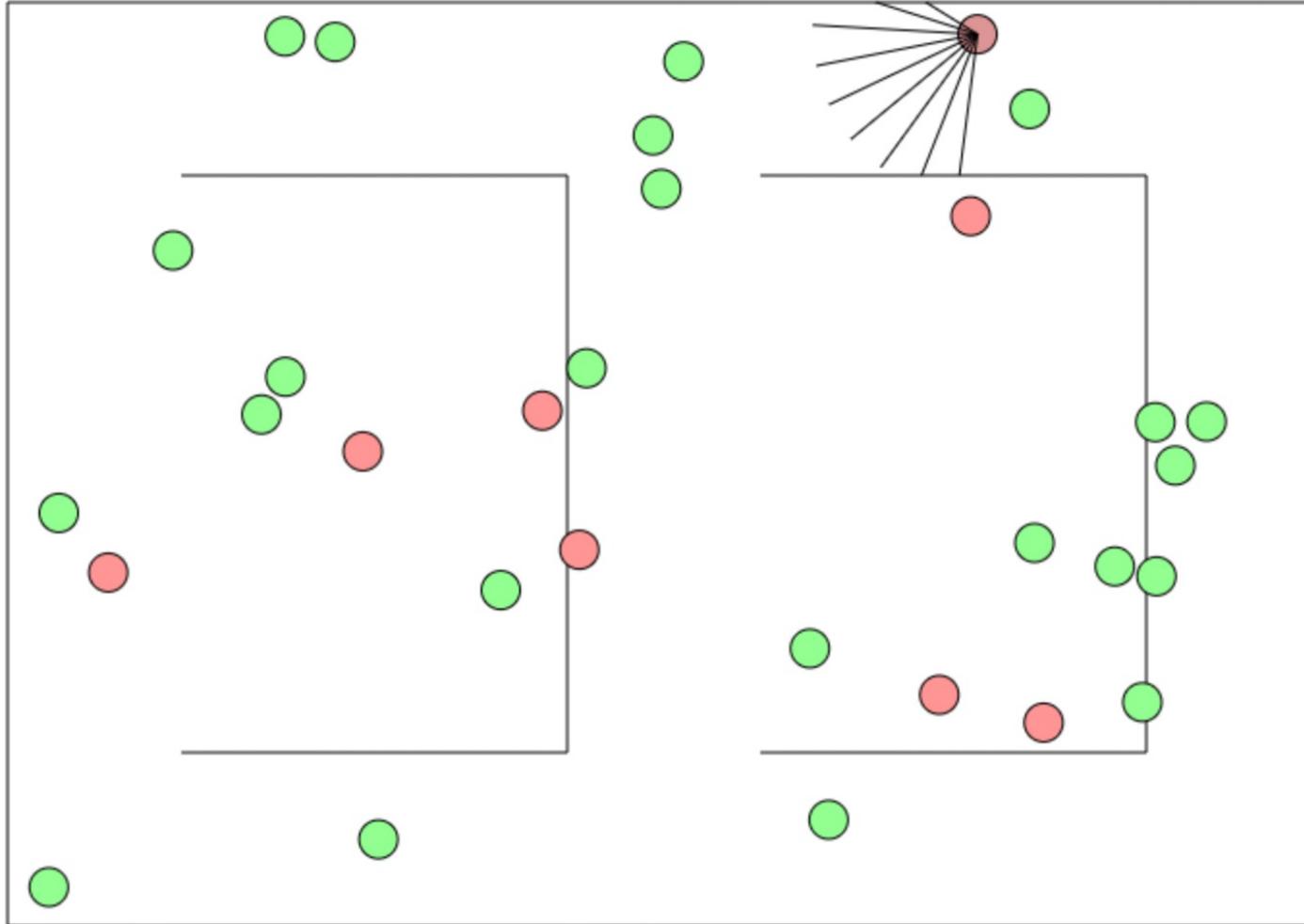


Deep Learning II

Chris Piech

CS109, Stanford University

Lets start training a Critter



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

Announcements (many)

1. Happy thanksgiving
2. Challenge is due tonight, 11:59pm AOE.
3. Wed is the last day of class
4. Last PSet due Wed, will accept late by Friday
5. Final Exam is next Thursday
 - a. 3 hours. Open notes (15 pages front/back)
 - b. Review session Dec 2nd 6-8p (recorded)
 - c. Practice exams posted by end of day today
 - d. Cumulative. Covers material up through last Friday's class. Emphasis on second half and psets. Section problems are great study tools!
6. Last sections and office hours this week.

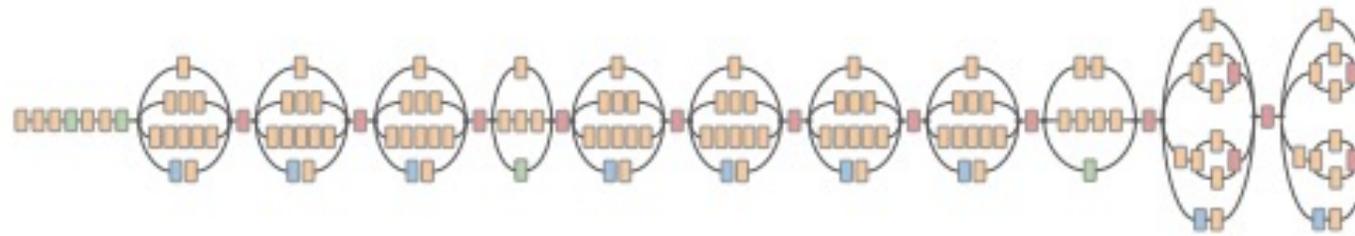
Review

Detecting skin cancer

Skin Lesion Image



Deep Convolutional Neural Network (Inception-v3)

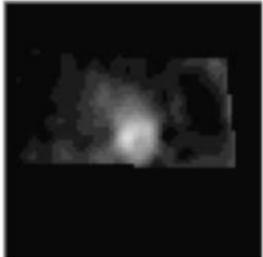


Training Classes (757)

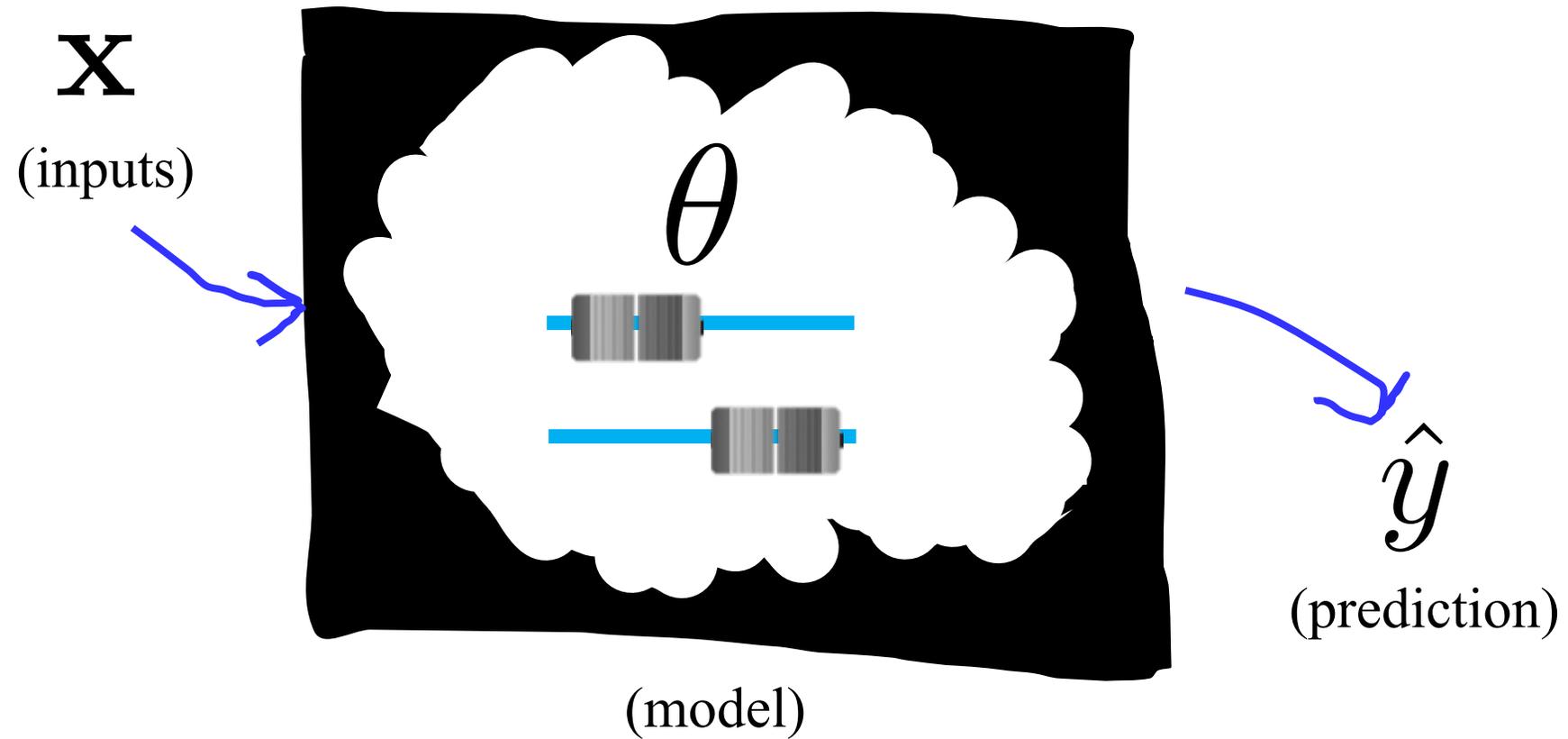
- Acral-lent. melanoma
- Amelanotic melanoma
- Lentigo melanoma
- ...
- Blue nevus
- Halo nevus
- Mongolian spot
- ...
-
-
-

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

Binary Classification Task

	ROI 1	ROI 2	...	ROI m	Output
			...		
Heart 1	0	1		1	0
Heart 2	1	1		1	0
			⋮		⋮
Heart n	0	0		0	1

Machine Learning



The Training / Testing Paradigm

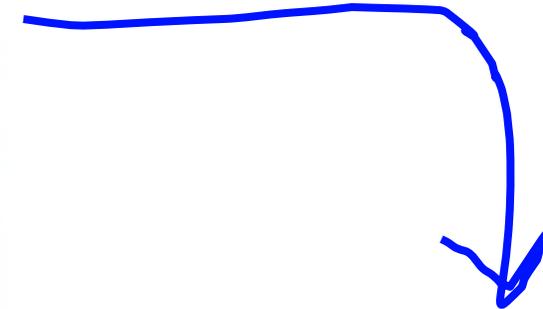


Learn your
parameters

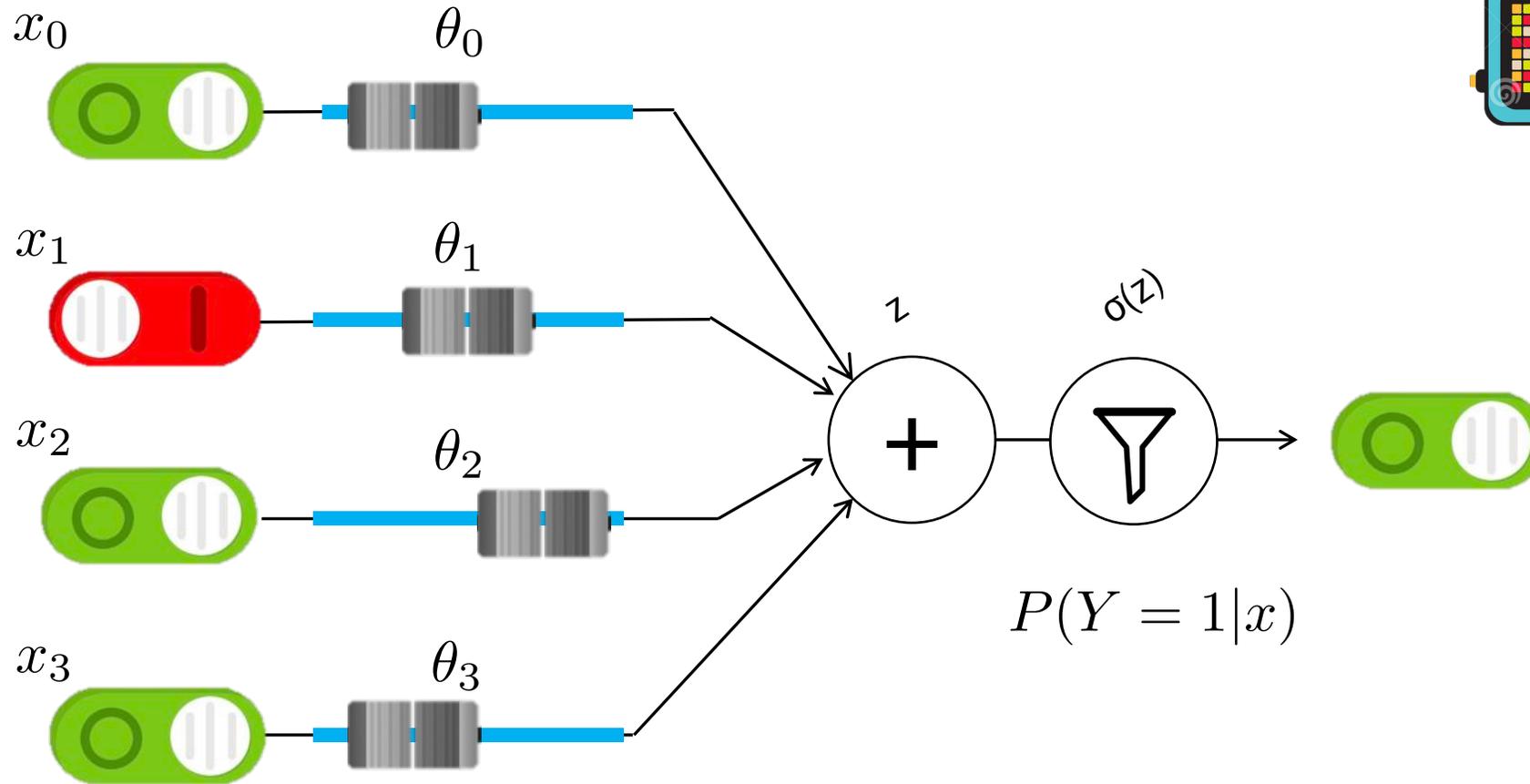


Make sure that
they work

If your model passes
testing...



Logistic Regression



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

Math for Logistic Regression

1

Make logistic regression assumption

$$P(Y = 1|X = \mathbf{x}) = \sigma(\theta^T \mathbf{x})$$

$$P(Y = 0|X = \mathbf{x}) = 1 - \sigma(\theta^T \mathbf{x})$$

Often call this

\hat{y}

2

Calculate the log likelihood for all data

$$LL(\theta) = \sum_{i=1}^n y^{(i)} \log \sigma(\theta^T \mathbf{x}^{(i)}) + (1 - y^{(i)}) \log[1 - \sigma(\theta^T \mathbf{x}^{(i)})]$$

3

Get derivative of log likelihood with respect to thetas

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

Logistic Regression Training

Initialize: $\theta_j = 0$ for all $0 \leq j \leq m$

Repeat many times:

gradient[j] = 0 for all $0 \leq j \leq m$

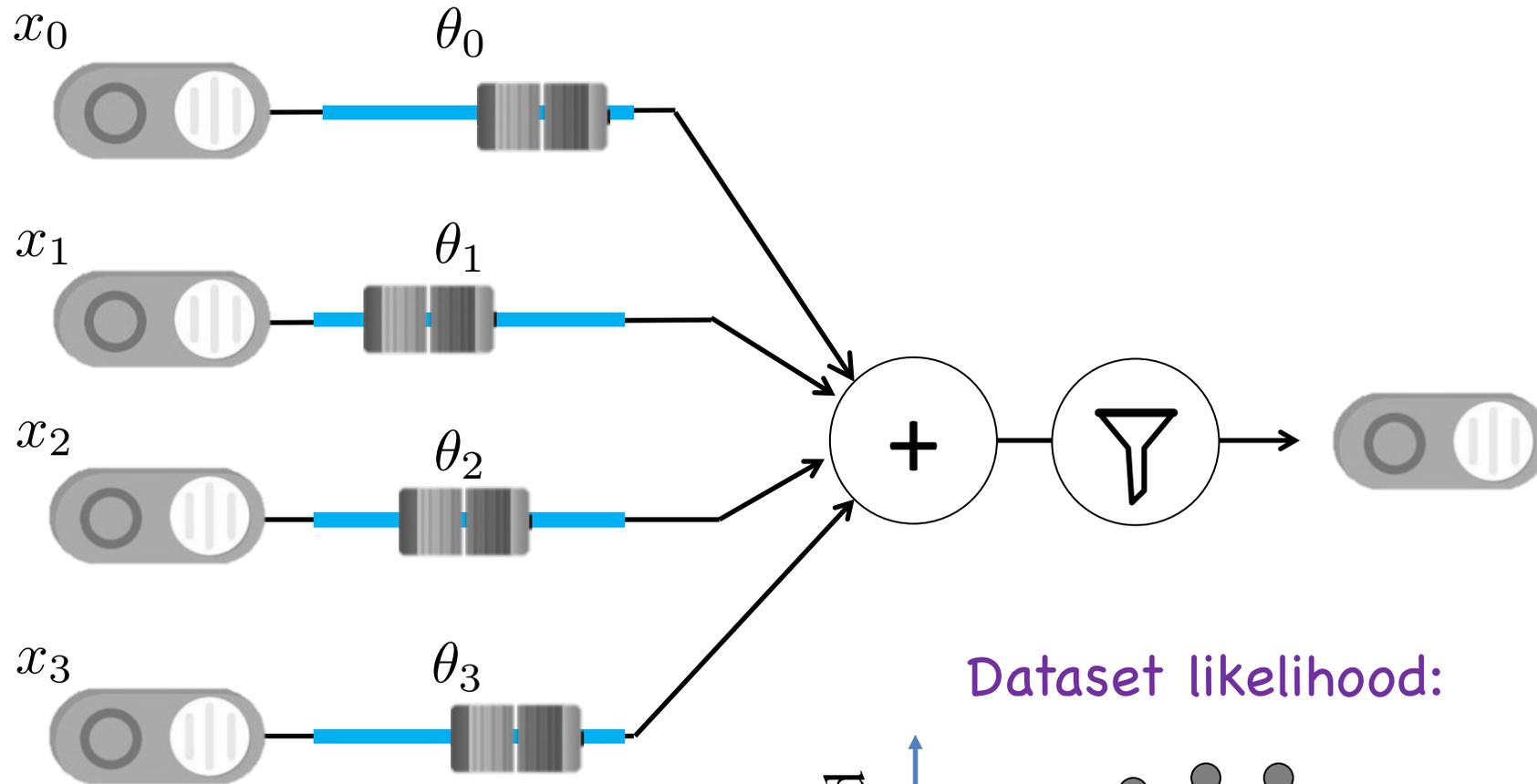
For each training example (\mathbf{x}, y) :

For each parameter j :

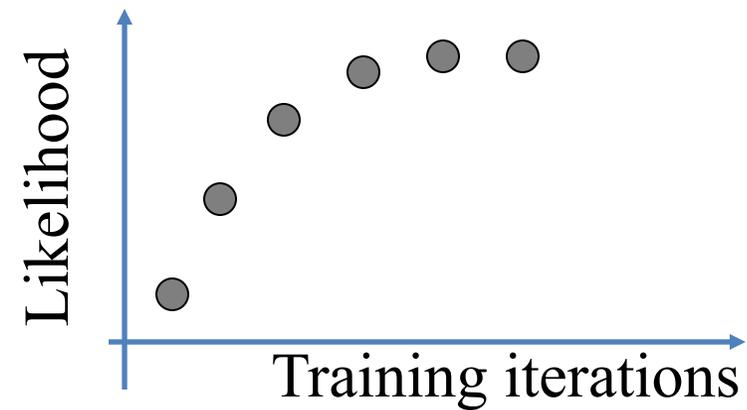
$$\text{gradient}[j] \quad += \quad x_j \left(y - \frac{1}{1 + e^{-\theta^T \mathbf{x}}} \right)$$

$\theta_j += \eta * \text{gradient}[j]$ for all $0 \leq j \leq m$

Training



Dataset likelihood:



Speeding up Gradient Descent

```
initialize  $\theta_j = 0$  for  $0 \leq j \leq m$   
repeat many times:
```

```
  gradient[j] = 0 for  $0 \leq j \leq m$ 
```

```
  for each training example  $(x, y)$ :
```

```
    for each  $0 \leq j \leq m$ :
```

```
      compute gradient
```

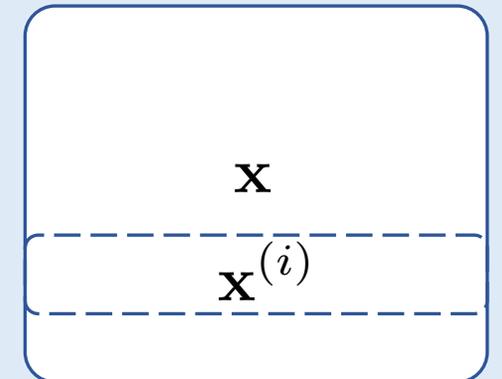
```
   $\theta_j -= \eta * \text{gradient}[j]$  for all  $0 \leq j \leq m$ 
```

1. What if we have 1,200,000 datapoints in our training set?
2. How can we speed up the update?

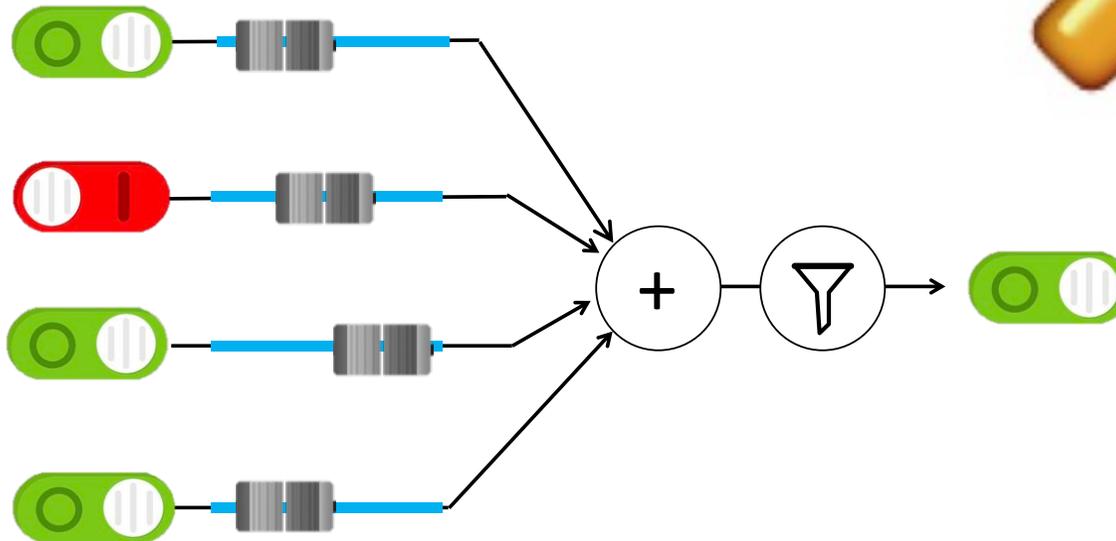
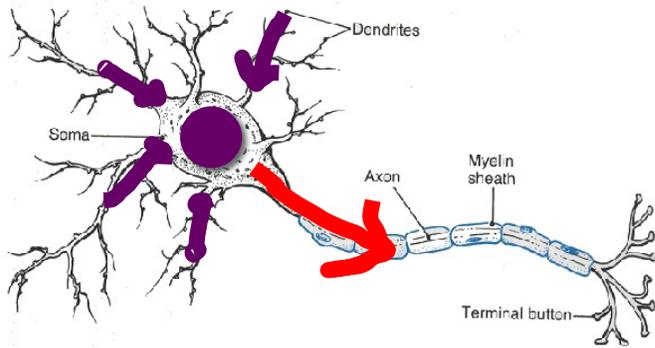
$$\frac{\partial LL(\theta)}{\partial \theta_j} = \sum_{i=1}^n \left[y^{(i)} - \sigma(\theta^T \mathbf{x}^{(i)}) \right] x_j^{(i)}$$

Start with this:

$$\theta^T \mathbf{x}^{(i)}$$

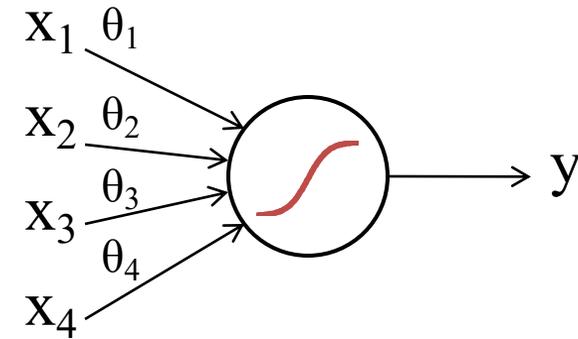
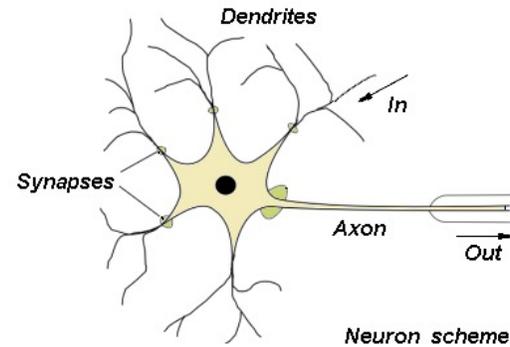


Artificial Neurons

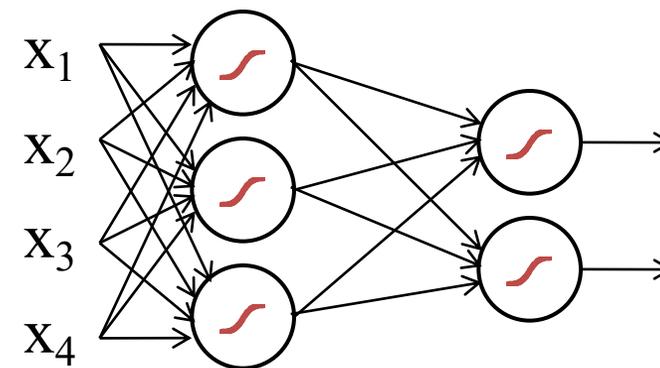
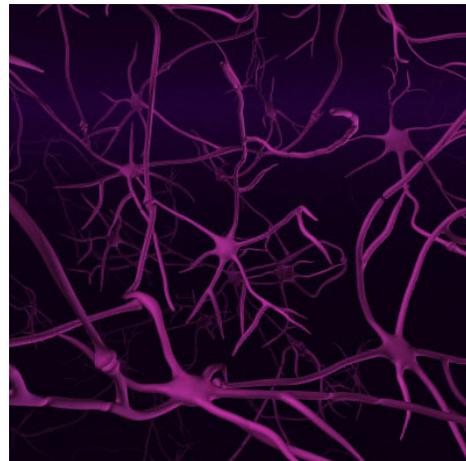


Biological Basis for Neural Networks

A neuron

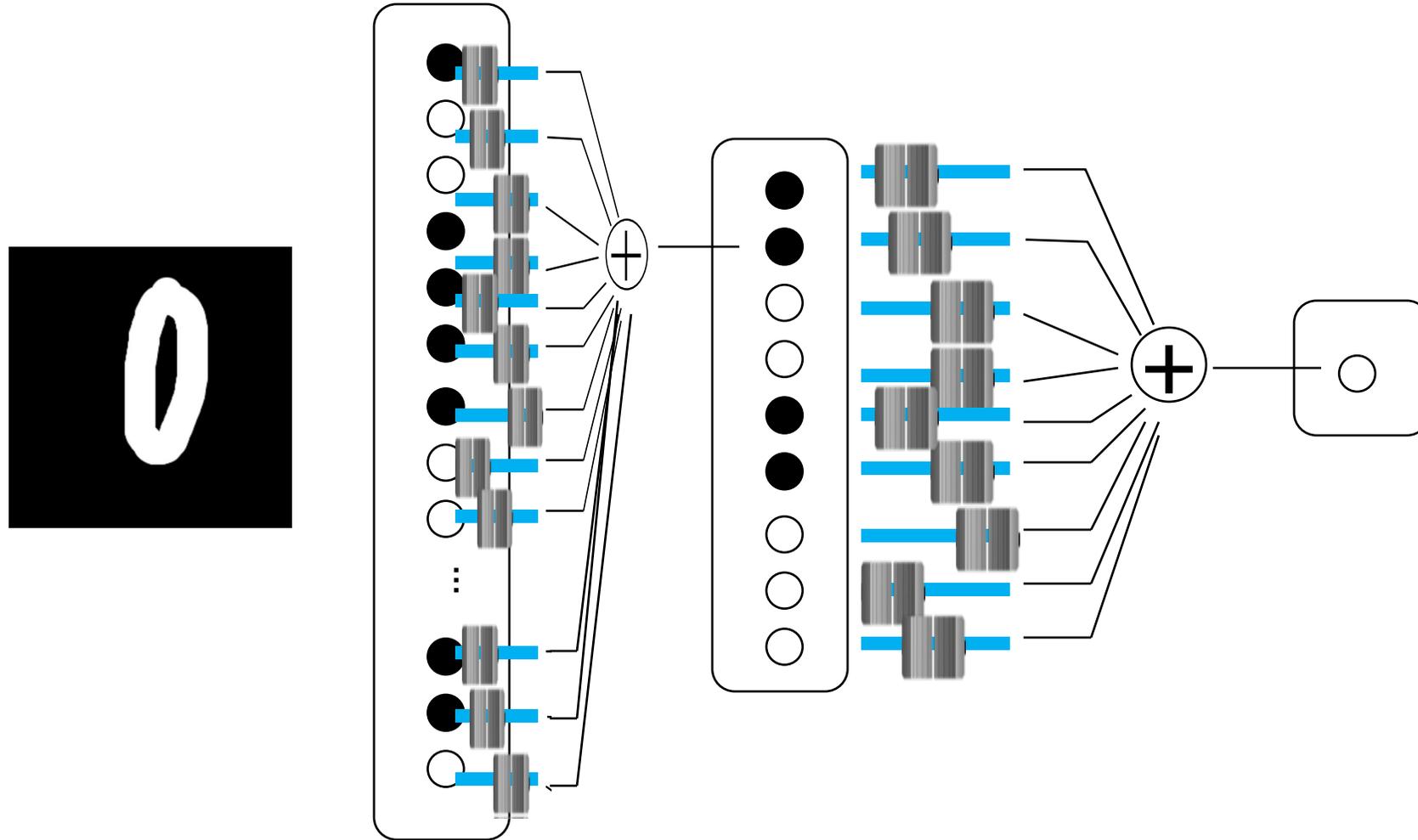


Your brain

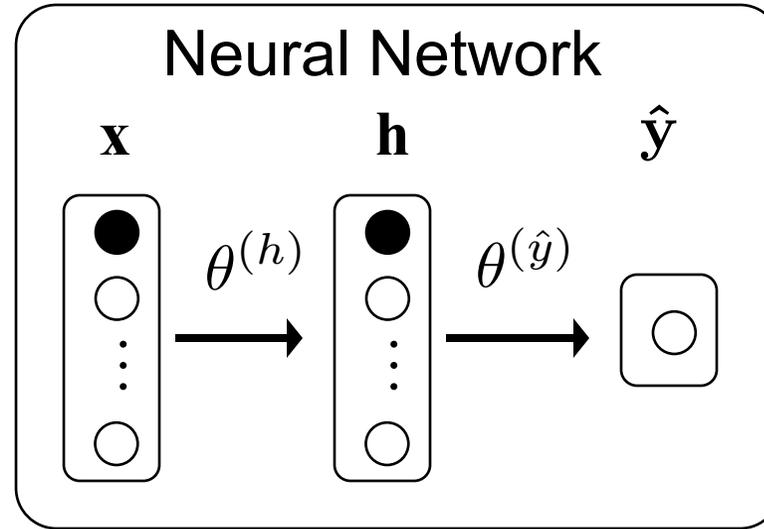


Actually, it's probably someone else's brain

We Can Put Neurons Together



Many many Parameters



$$|\mathbf{x}| = 40$$

$$|\mathbf{h}| = 20$$

How many parameters in total?

a) 800

b) 20

c) 820

d) 16000

Only Have to Do Three Things

- 1 Make deep learning assumption

$$P(Y = 1|X = \mathbf{x}) = \hat{y}$$

$$P(Y = 0|X = \mathbf{x}) = 1 - \hat{y}$$

$$\hat{y} = \sigma \left(\sum_{j=0}^{m_h} \mathbf{h}_j \theta_j^{(\hat{y})} \right)$$

- 2 Calculate the log probability for all data

$$LL(\theta) = \sum_{i=0}^n y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log[1 - \hat{y}^{(i)}]$$

- 3 Get partial derivative of log likelihood with respect to each theta

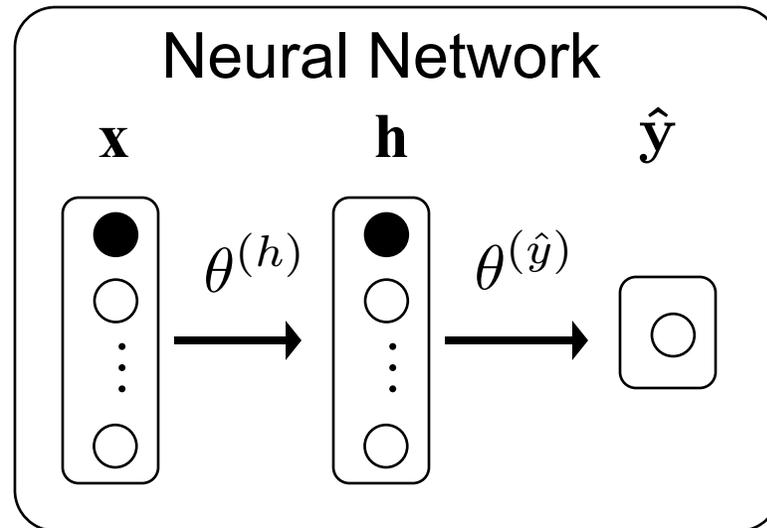
Summary: Simple Calculations For

Loss with respect to
output layer params

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

Loss with respect to
hidden layer params

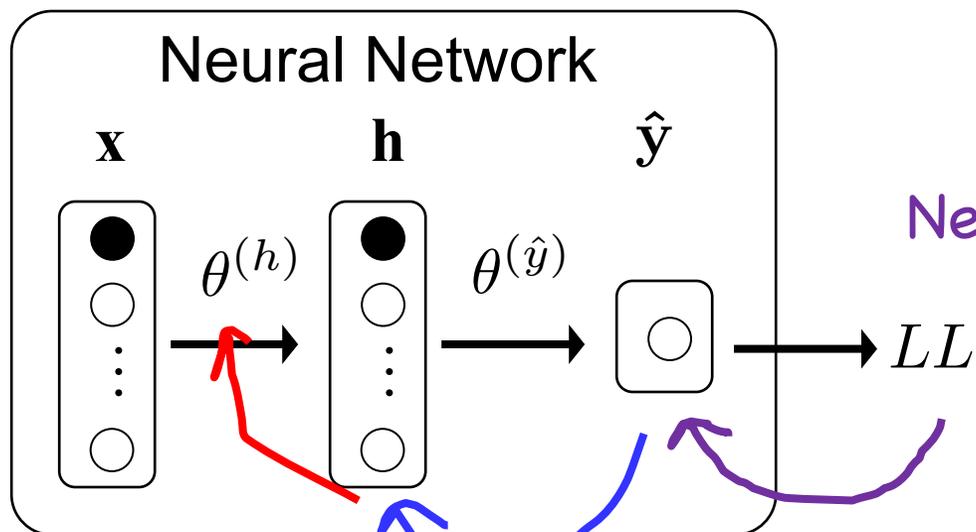
$$\frac{\partial LL(\theta)}{\partial \theta_{i,j}^{(h)}}$$



Chain Rule Strategy

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

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

Calculate the Derivative

$$\frac{\partial LL(\theta)}{\partial \theta_{i,j}^{(h)}} = \begin{array}{|c|c|c|} \hline \img alt="Chest icon" data-bbox="443 171 525 316"/> & \img alt="Turtle icon" data-bbox="525 171 607 316"/> & \img alt="Dinosaur icon" data-bbox="607 171 689 316"/> \\ \hline \end{array}$$

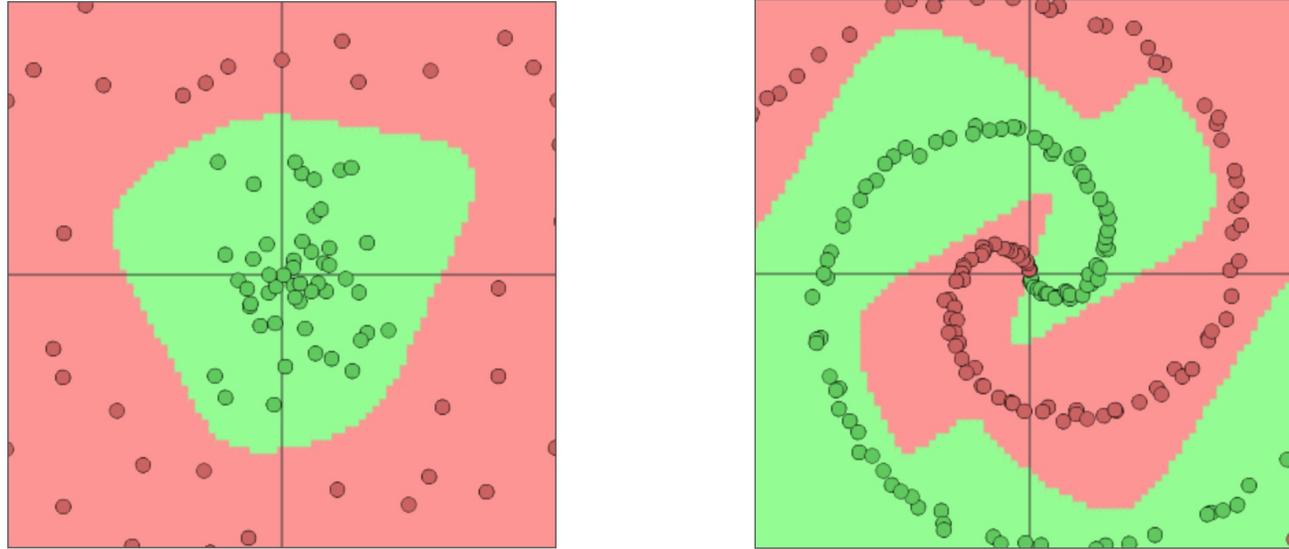
$$\begin{array}{|c|} \hline \img alt="Chest icon" data-bbox="344 358 426 503"/> \\ \hline \end{array} = \frac{y}{\hat{y}} - \frac{(1-y)}{(1-\hat{y})}$$

$$\begin{array}{|c|} \hline \img alt="Turtle icon" data-bbox="344 569 426 714"/> \\ \hline \end{array} = \hat{y}[1-\hat{y}]\theta_j^{(\hat{y})}$$

$$\begin{array}{|c|} \hline \img alt="Dinosaur icon" data-bbox="348 757 430 902"/> \\ \hline \end{array} = \mathbf{h}_j[1-\mathbf{h}_j]\mathbf{x}_j$$

Neural Networks Can Learn Complex Functions

- Some data sets/functions are not linearly separable



- These are classifiers learned by neural networks

<http://cs.stanford.edu/people/karpathy/convnetjs/demo/classify2d.html>

End Review

Chain rule:
Game changer for
artificial intelligence

Beyond Binary Classification

Multiple Outputs

Draw your number here



X Pencil Eraser

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

0 1 2 3 4 5 6 7 8 9



The Categorical

Binary results (eg coin)

More than two outcomes (eg dice)

One experiment
Many experiments

Bernoulli	???
Binomial	Multinomial

The Categorical

Binary results (eg coin)

More than two outcomes (eg dice)

One experiment
Many experiments

Bernoulli	Categorical
Binomial	Multinomial

Categorical Classification?



Softmax is a generalization of the sigmoid function that squashes a K -dimensional vector \mathbf{z} of arbitrary real values to a K -dimensional vector $\text{softmax}(\mathbf{z})$ of real values in the range $[0, 1]$ that **add up to 1**.

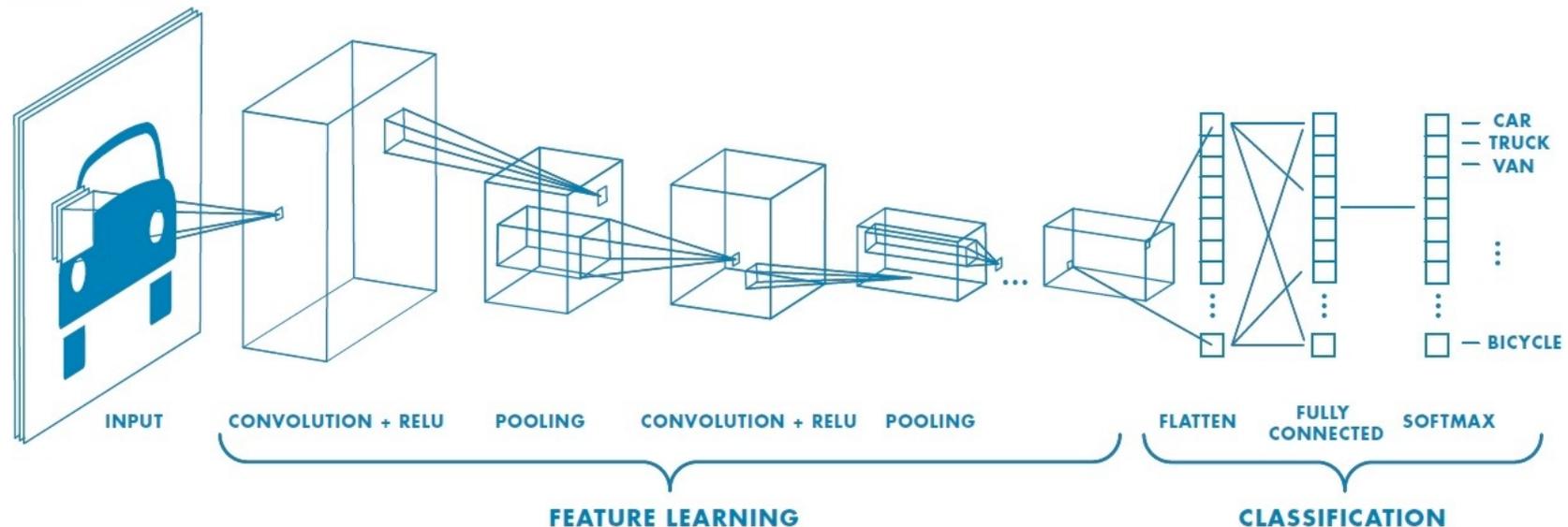
$$P(Y = j | \mathbf{X} = \mathbf{x}) = \text{softmax}(f(\mathbf{x}))_j$$

Sigmoid is to Bernoulli as **Softmax is to Categorical**

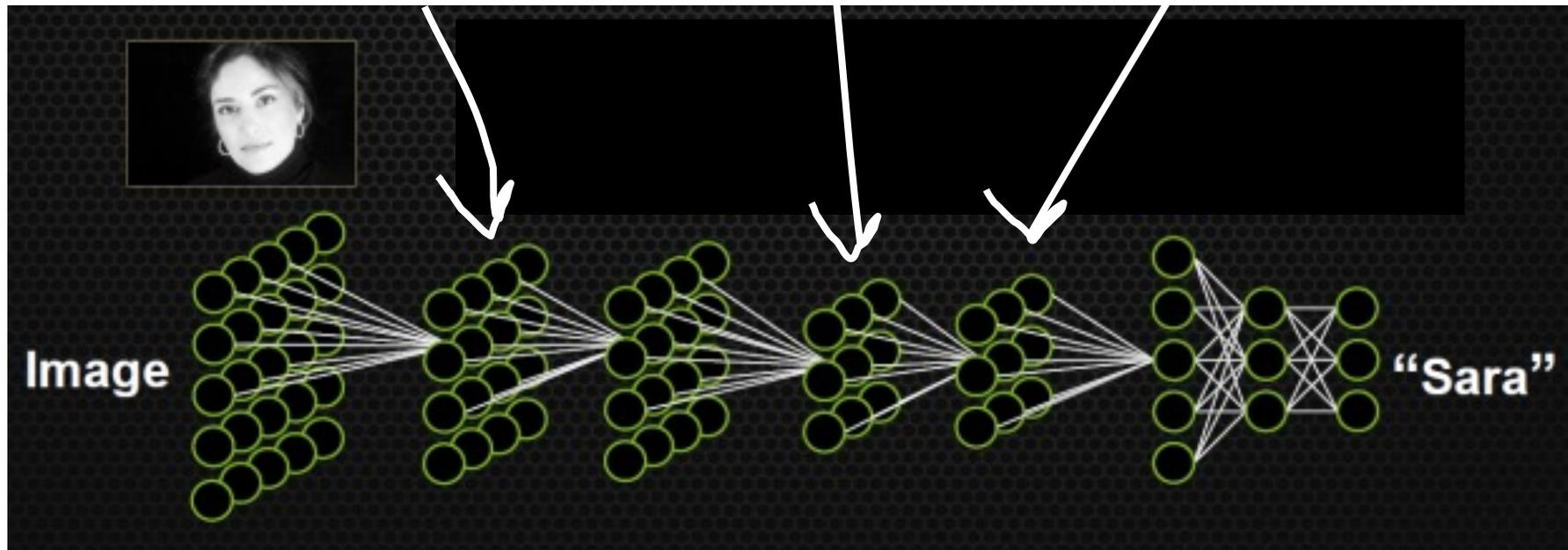
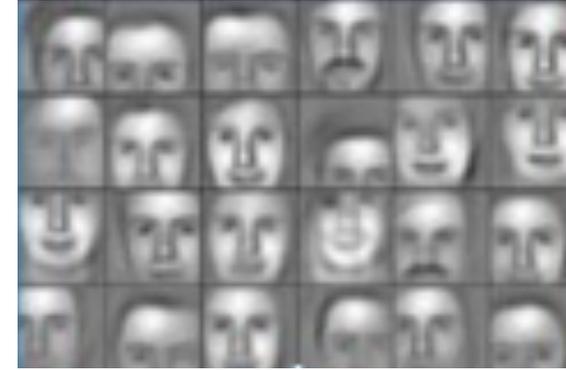
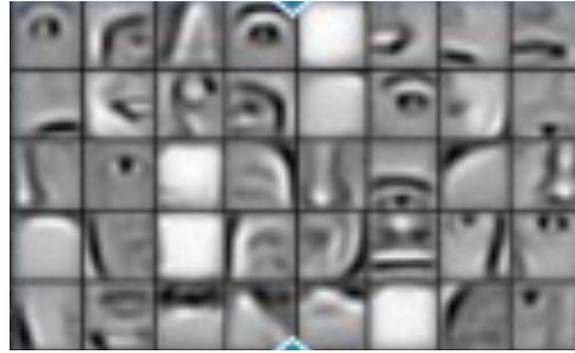
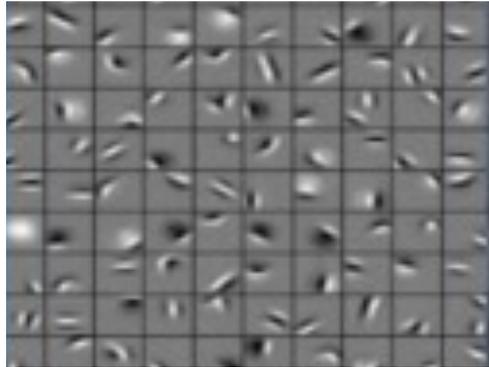
Shared Weights?



Convolution it turns out if you want to force some of your weights to be shared for different neurons, the math isn't that much harder. This is used a lot for vision (CNN).

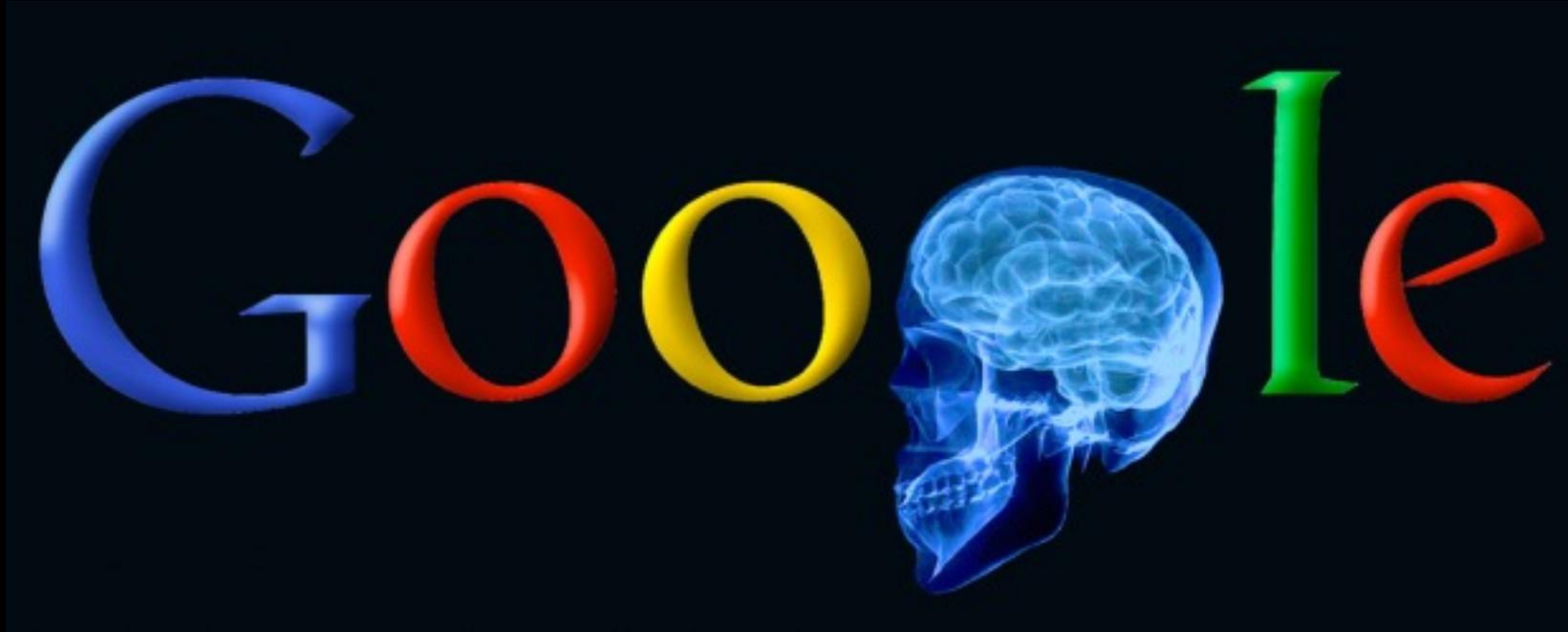


Works for any number of layers



\hat{y}

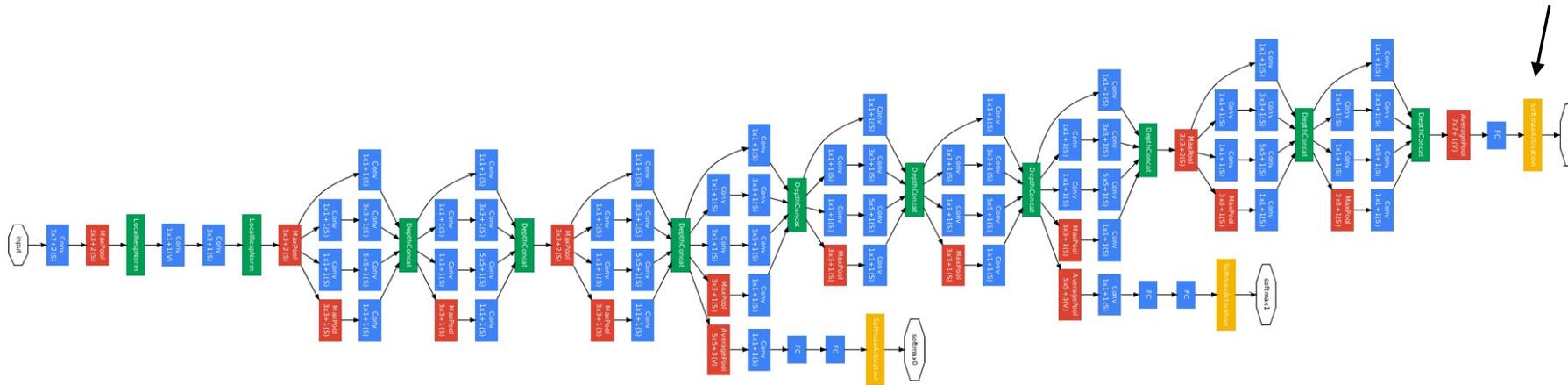
GoogLeNet Brain



1 Trillion Artificial Neurons

GoogLeNet Brain

Multiple,
Multi class output



22 layers deep

The Cat Neuron



Top stimuli from the test set



Optimal stimulus
by numerical optimization

Hire the smartest people in the world



Invent cat detector

Best Neuron Stimuli

Neuron 1



Neuron 2



Neuron 3



Neuron 4



Neuron 5



Best Neuron Stimuli

Neuron 6



Neuron 7



Neuron 8



Neuron 9



Best Neuron Stimuli

Neuron 10



Neuron 11



Neuron 12



Neuron 13



ImageNet Classification

22,000 categories

14,000,000 images

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

22,000 is a lot!

...

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, *Sphyrna tiburo*

angel shark, angelfish, *Squatina squatina*, monkfish

electric ray, crampfish, numbfish, torpedo

smalltooth sawfish, *Pristis pectinatus*

guitarfish

rougtail 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



0.005%

Random guess

1.5%

Pre Neural Networks

?

GoogLeNet

0.005%

Random guess

1.5%

Pre Neural Networks

43.9%

GoogLeNet

0.005%

Random guess

1.5%

Pre Neural Networks

95.1%

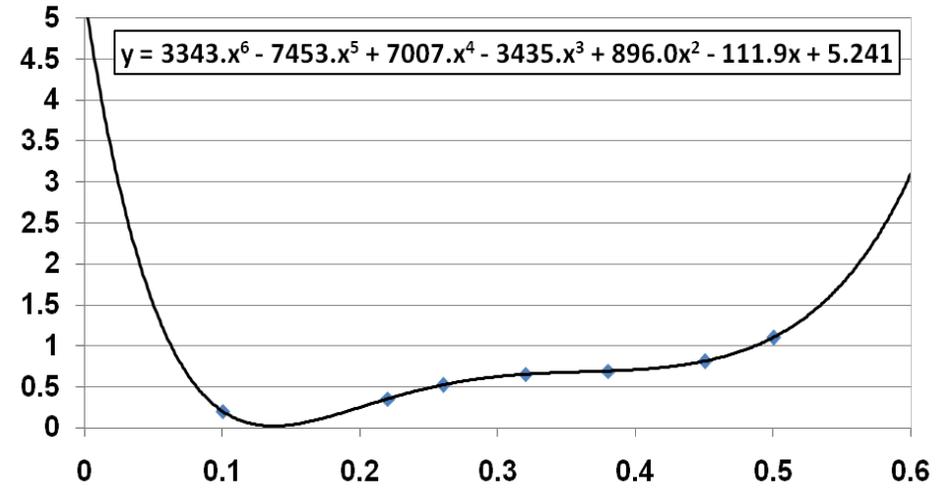
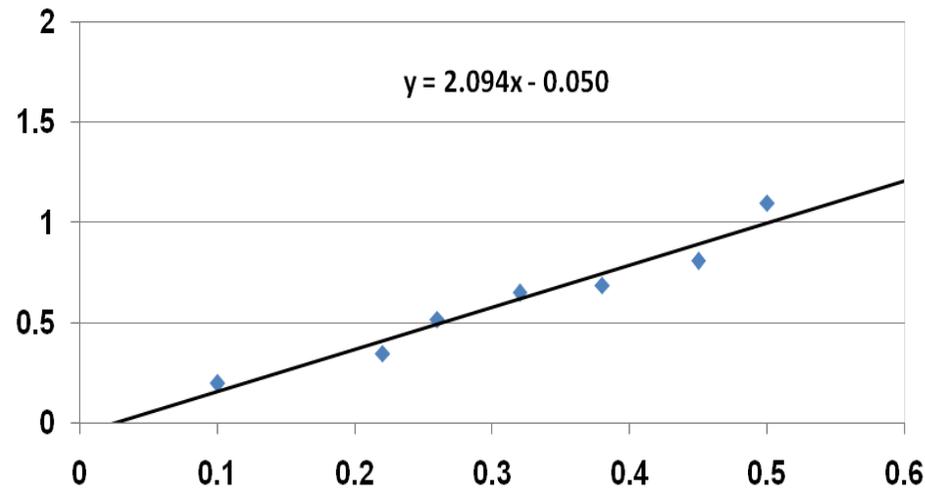
SE-ResNet

How many parameters
is too many?

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

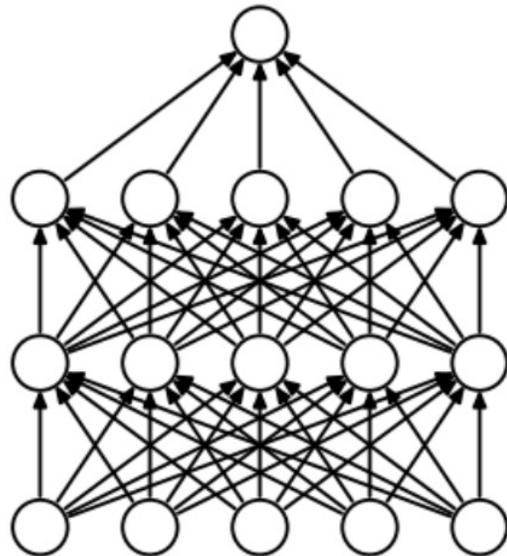


- Polynomial on the right fits training data perfectly!
- Which would you rather use to predict a new data point?

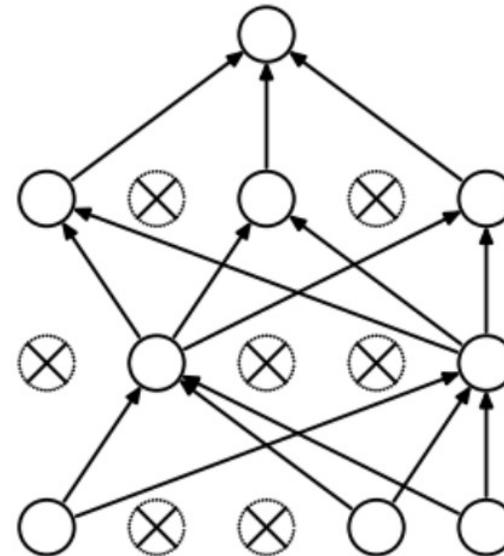
Prevent Overfitting?



Dropout when your model is training, randomly turn off your neurons with probability 0.5. It will make your network more robust.



(a) Standard Neural Net



(b) After applying dropout.

Not everything is classification

Types of Neural Network Tasks

Classification

Regression

Reinforcement
Learning

Generation

First lesson in CS229

Types of Neural Network Tasks

Classification

Regression

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 X and Y

Assume IID data:

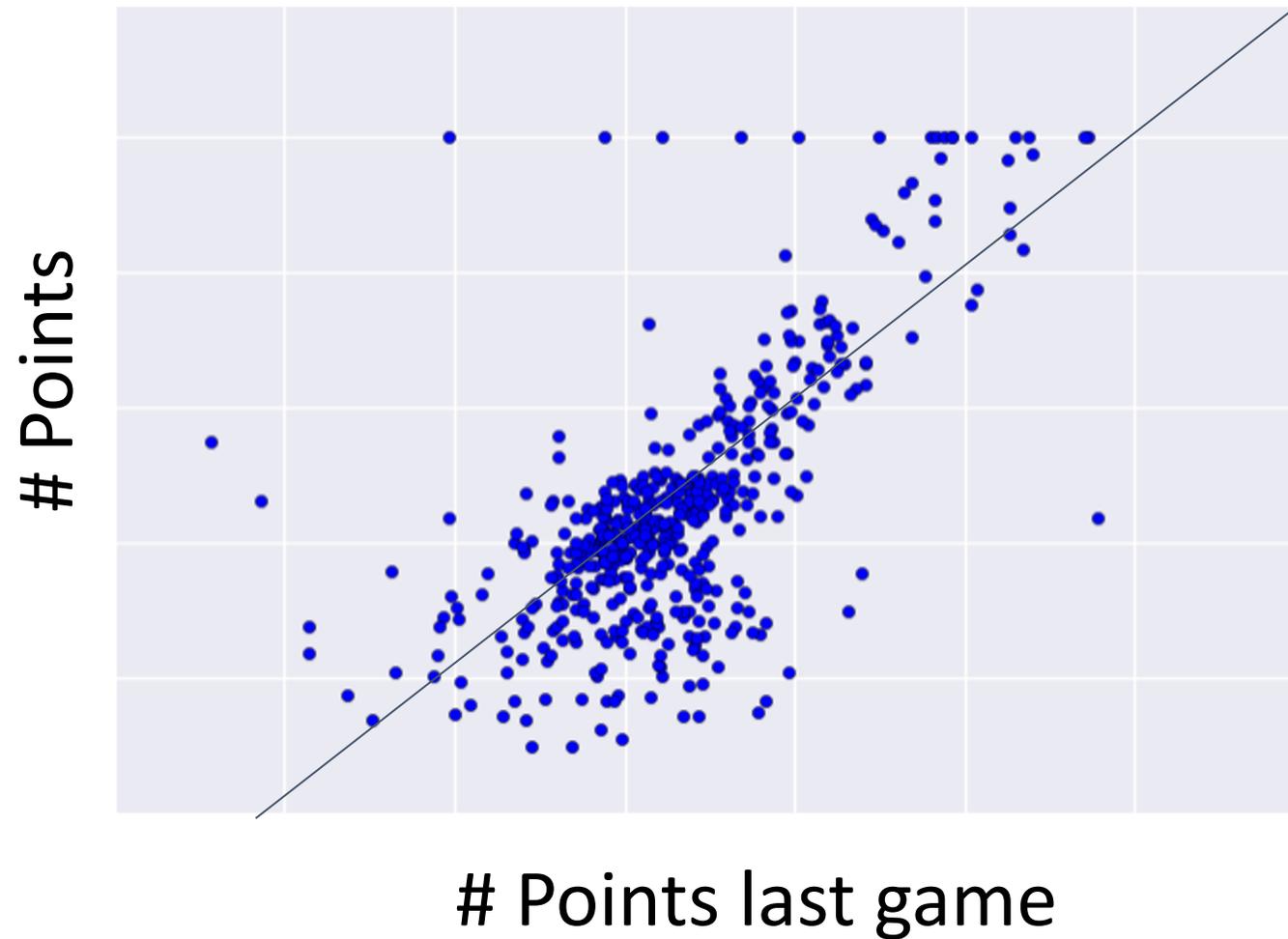
n training datapoints

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

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

Each datapoint has m features and a single output

Linear Regression with one Input



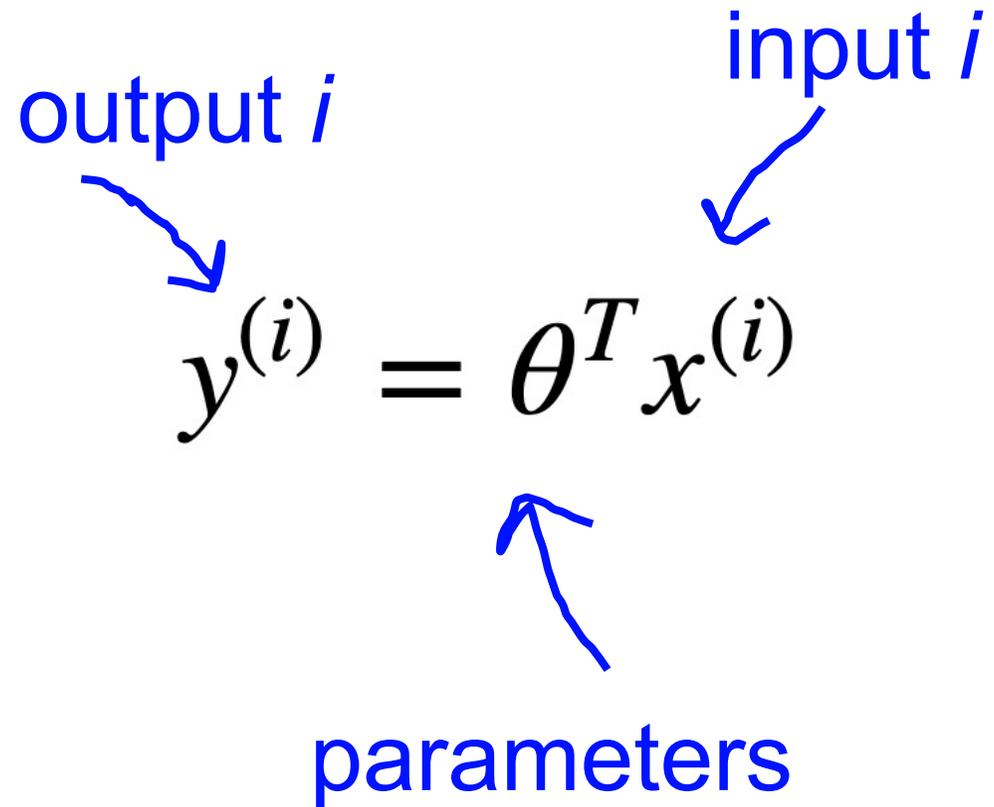
Linear Regression Model

output i

input i

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

parameters

The diagram shows the equation $y^{(i)} = \theta^T x^{(i)}$ centered on the page. Three blue arrows point from text labels to parts of the equation: one from 'output i ' to $y^{(i)}$, one from 'input i ' to $x^{(i)}$, and one from 'parameters' to θ^T .

Linear Regression Model

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

output i input i

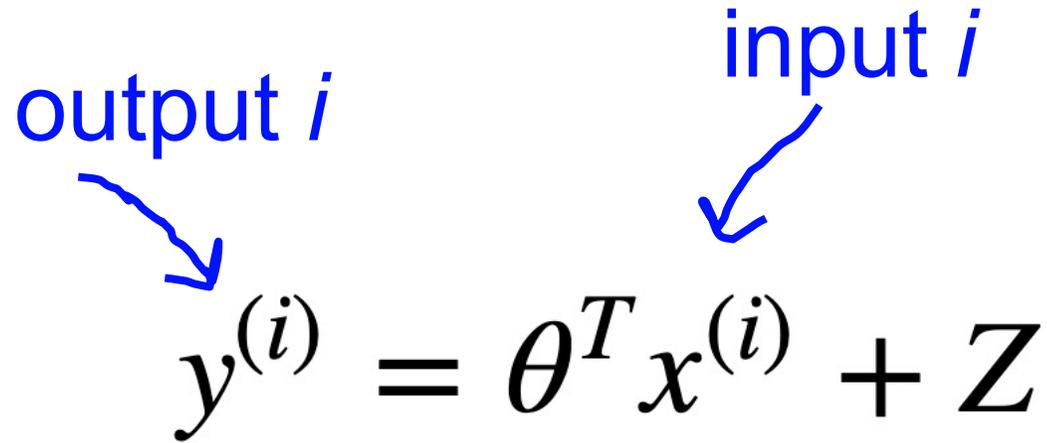
parameters random noise

The diagram shows the linear regression model equation $y^{(i)} = \theta^T x^{(i)} + Z$. Hand-drawn blue arrows point from the labels to the corresponding terms in the equation: 'output i ' points to $y^{(i)}$, 'input i ' points to $x^{(i)}$, 'parameters' points to θ^T , and 'random noise' points to Z .

Linear Regression Model

output i

input i

$$y^{(i)} = \theta^T x^{(i)} + Z$$
A diagram showing the linear regression equation $y^{(i)} = \theta^T x^{(i)} + Z$. Blue arrows point from the labels 'output i', 'input i', 'parameters', and 'random noise' to their respective variables in the equation: $y^{(i)}$, $x^{(i)}$, θ , and Z .

Noise is N with mean 0

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

parameters

random noise

Linear Regression Model

output i

noise

noise is N with mean 0

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

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

$y^{(i)} = \theta^T x^{(i)} + Z$ noise is N with mean 0

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

Output is normal too: $y^{(i)} \sim N(\theta^T x^{(i)}, \sigma^2)$

Log Likelihood

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

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!

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

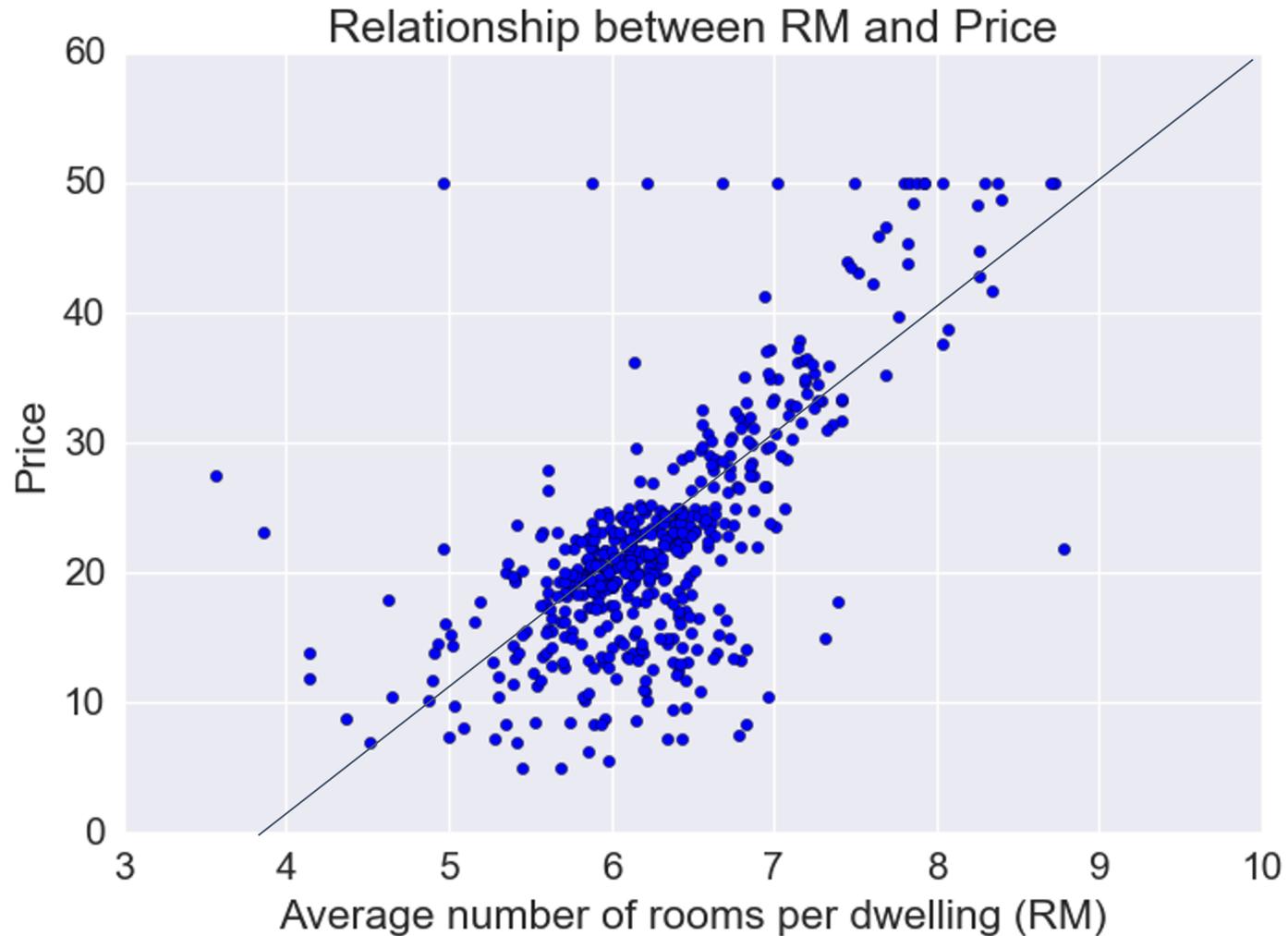
$$= - \sum_{i=1}^n 2 \left(y^{(i)} - \theta^T x^{(i)} \right) (-x_j^{(i)})$$

Chain rule

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

Simplify

Linear Regression with one Input



What does Five Thirty Eight Do?

Model Overfits Small Data-sample Per Player

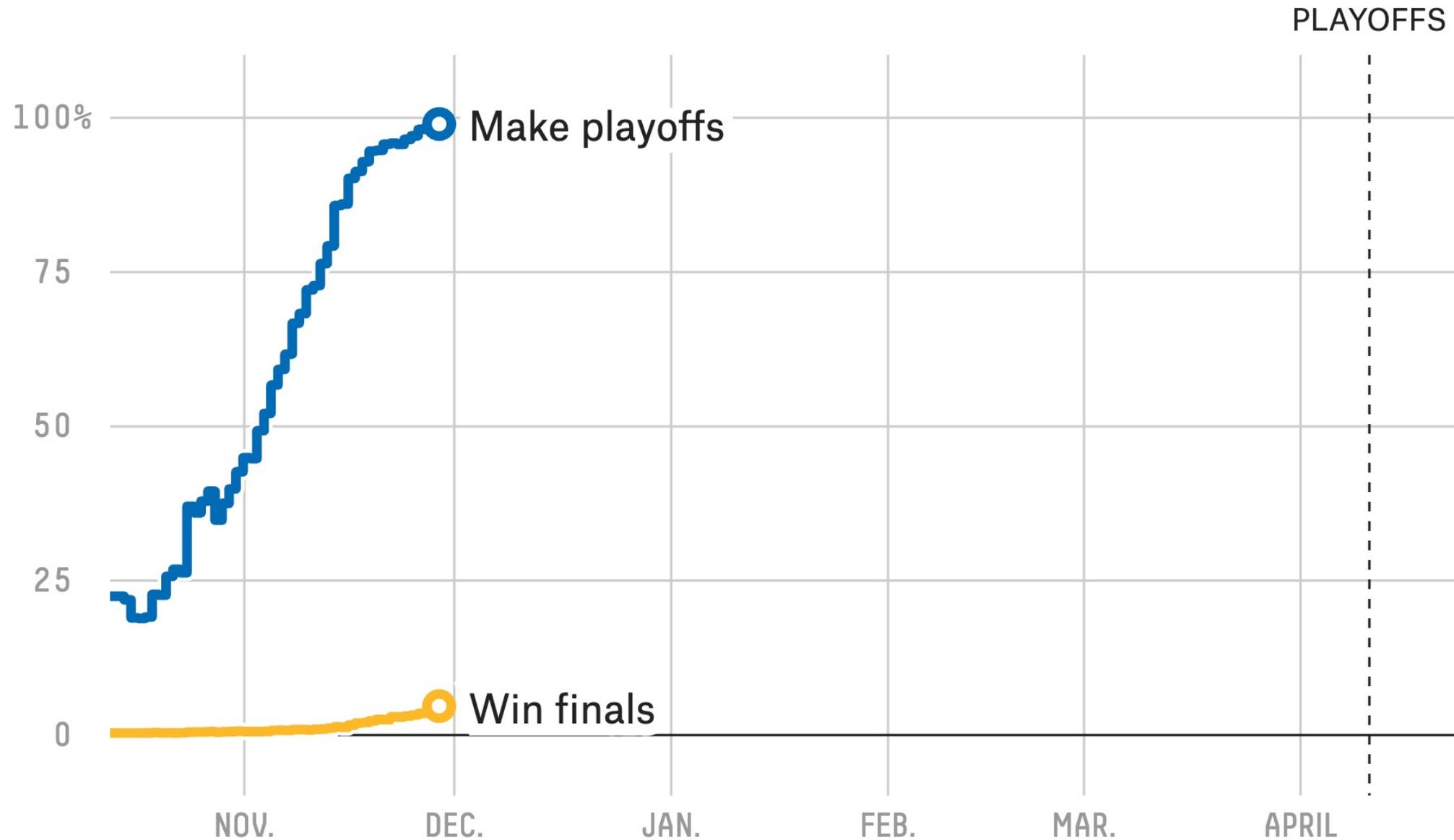
PLAYER	EXPECTED MINUTES PER GAME		PLAYER RATING	
	TOTAL	VS. FULL STRENGTH (TODAY)	OFF. +/-	DEF. +/-
Stephen Curry	36	+2	+7.8	+0.3
Jordan Poole	27	SAME	+0.5	-2.2
Draymond Green	29	SAME	-0.9	+2.5
Andrew Wiggins	31	+2	-0.3	0.0
Klay Thompson	0	-18	-0.6	-1.1
James Wiseman	0	-18	-4.3	-0.4
Damion Lee	21	+5	-0.5	+0.8
Andre Iguodala	0	-16	-1.2	+1.6
Kevon Looney	18	+8	-2.1	+1.6
Otto Porter Jr.	17	+6	+0.9	+0.3
Nemanja Bjelica	16	+6	-0.6	+0.2
Gary Payton II	14	+2	+0.6	+2.4
Juan Toscano-Anderson	16	+6	-1.4	+0.8
Moses Moody	5	+5	-1.9	-1.5
Jonathan Kuminga	3	+3	-2.3	-1.0
Chris Chiozza	7	+7	+1.4	-0.9
Jeff Dowtin	0	SAME	-1.7	-0.8

Rotation ratings

First, we take a minute-weighted average of the player ratings to get a measure of team efficiency per 100 possessions, relative to the league.

+3.3 +1.8

Five Thirty Eight Updates Probabilities



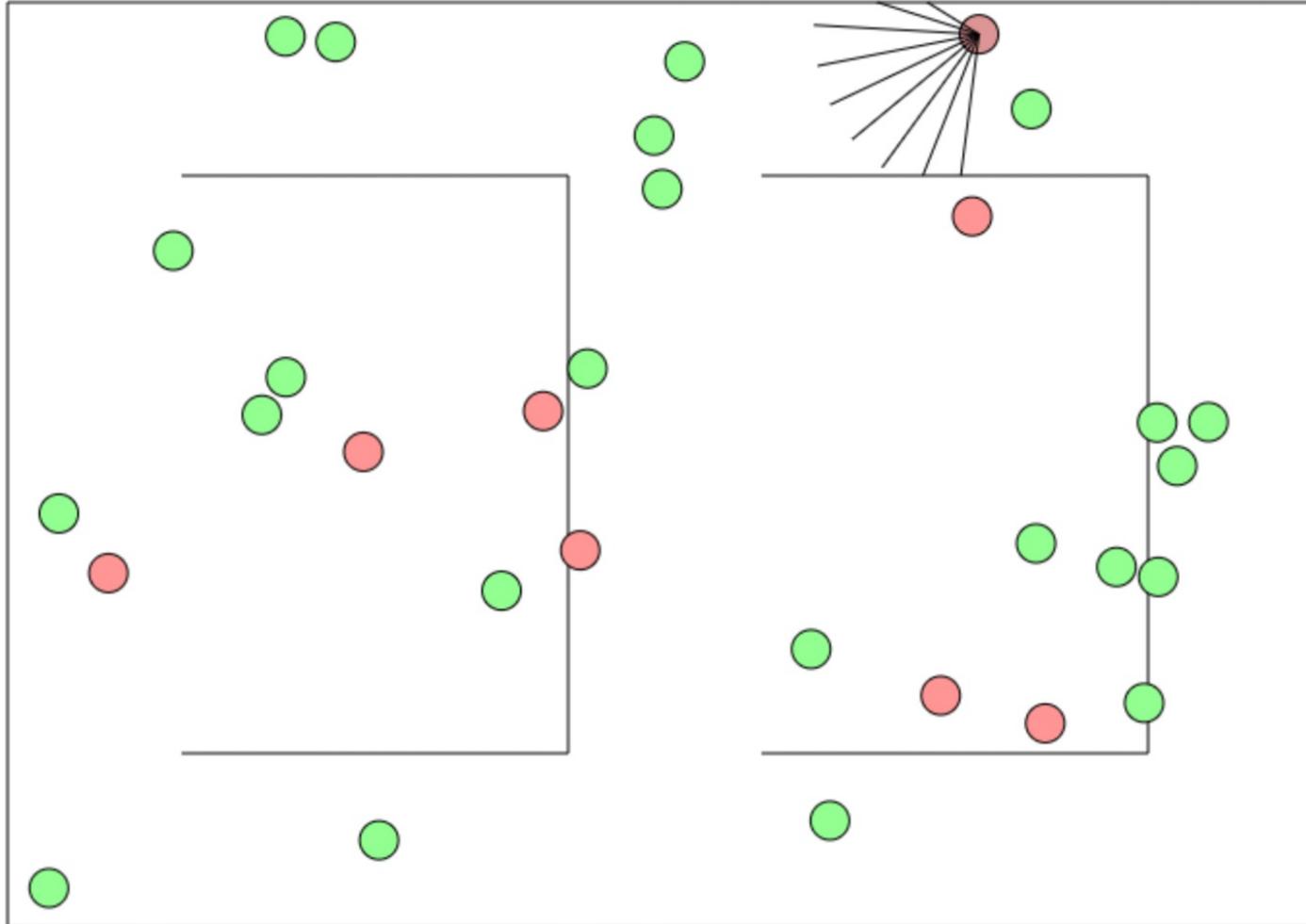
Making Decisions?



Deep Reinforcement Learning
Instead of having the output of a model be a probability you can make it an expectation.



Deep Reinforcement Learning

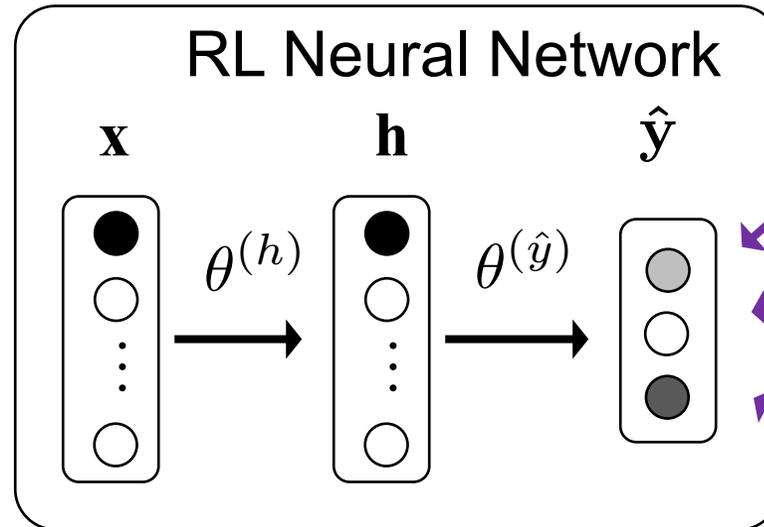


<http://cs.stanford.edu/people/karpathy/convnetjs/demo/rldemo.html>

Deep Reinforcement Learning

R is a reward and A_i is a legal action

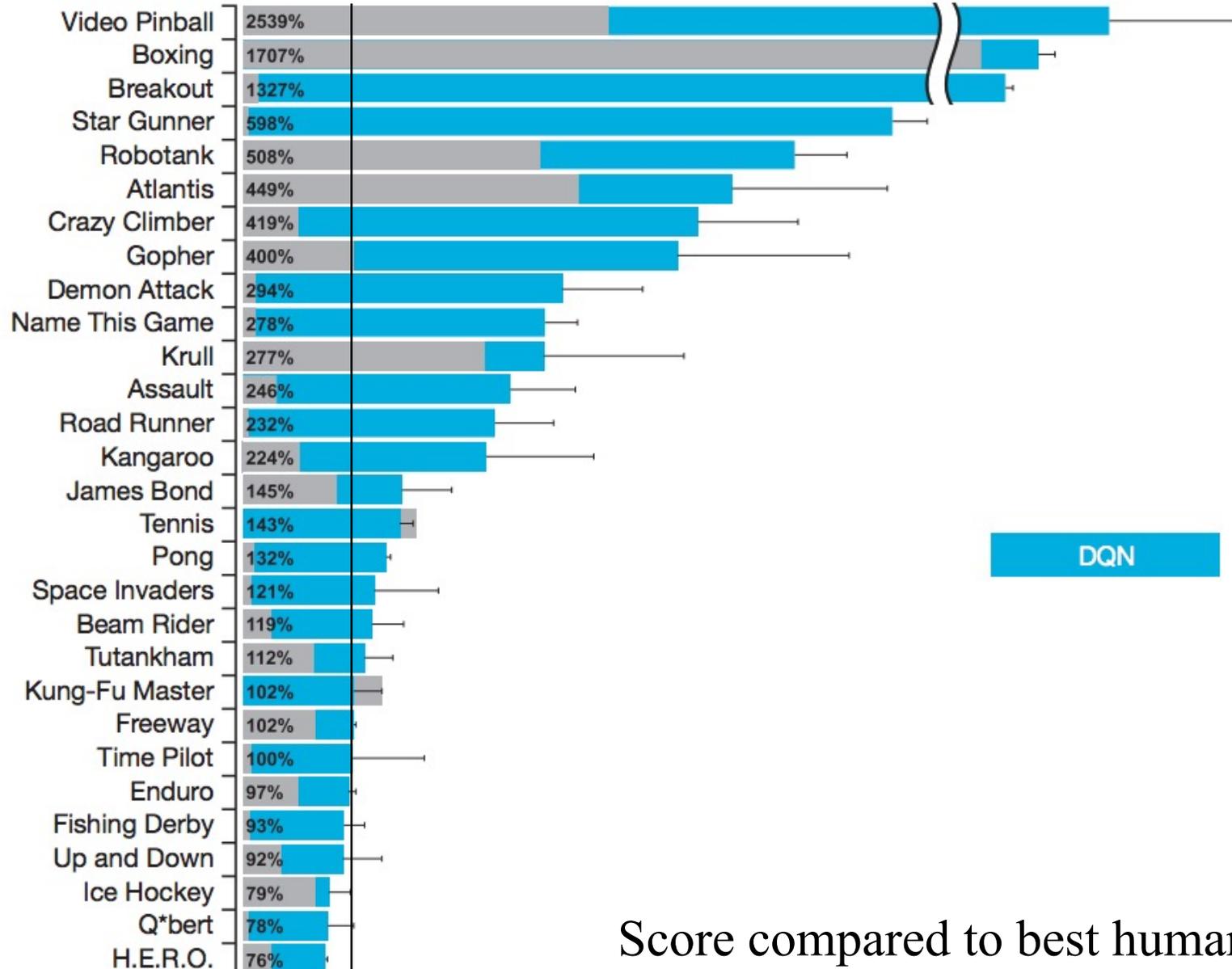
Input is a representation of current state (S)



$E[R | A_1, S]$
 $E[R | A_2, S]$
 $E[R | A_3, S]$

Interpret outputs as expected reward for a given action

Deep Mind Atari Games



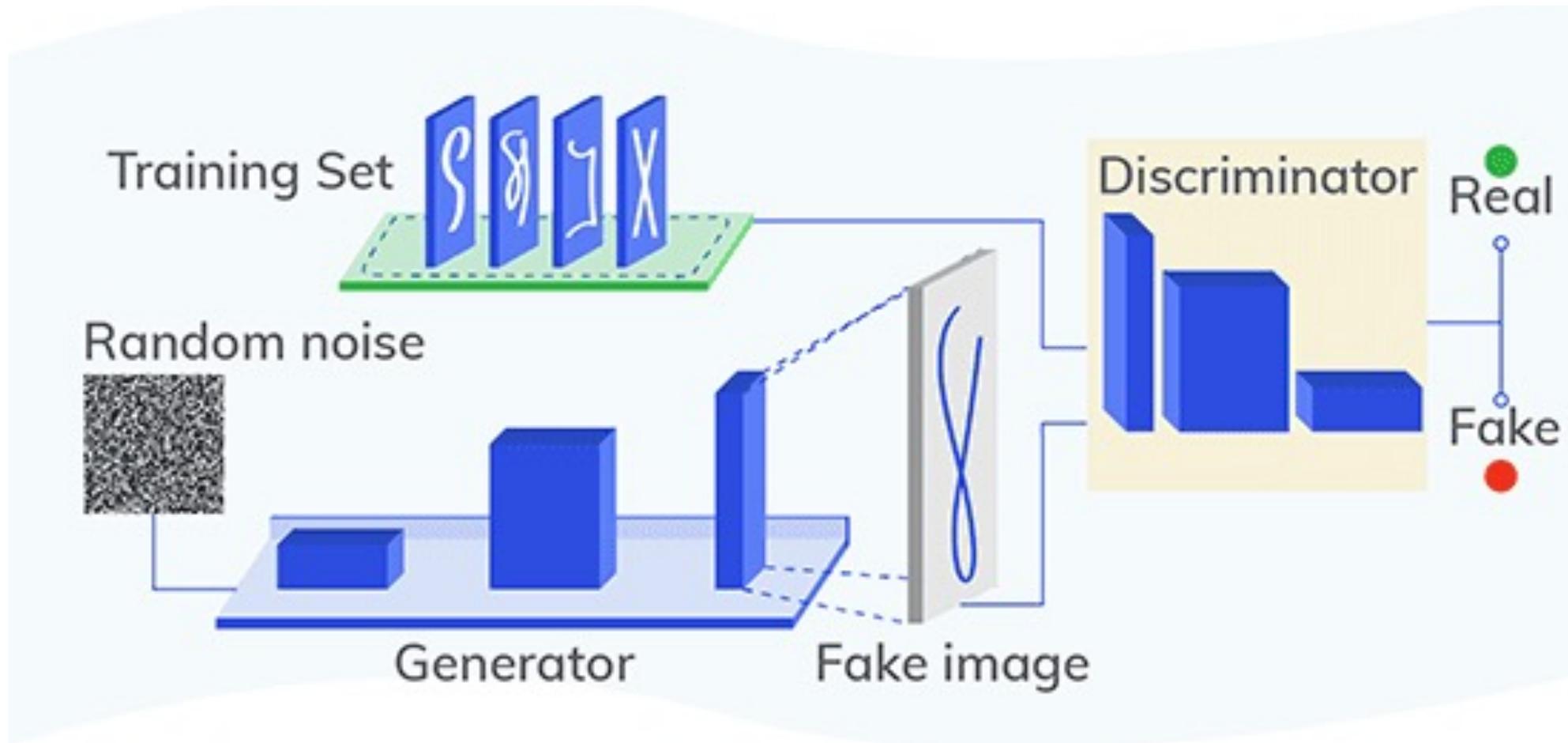
Score compared to best human

Generating New Examples



<https://generated.photos/>

Generative Adversarial Networks





When Do I Get a Robo Tutor?



Story of Riley



Exercise Type:

-  Solving for x-intercept
-  Solving for y-intercept
-  Graphing linear equations
-  Square roots
-  Slope of a line

Answer:

-  Correct
-  Incorrect



Story of Riley



Exercise Type:

-  Solving for x-intercept
-  Solving for y-intercept
-  Graphing linear equations
-  Square roots
-  Slope of a line

Answer:

-  Correct
-  Incorrect



Story of Riley



Exercise Type:

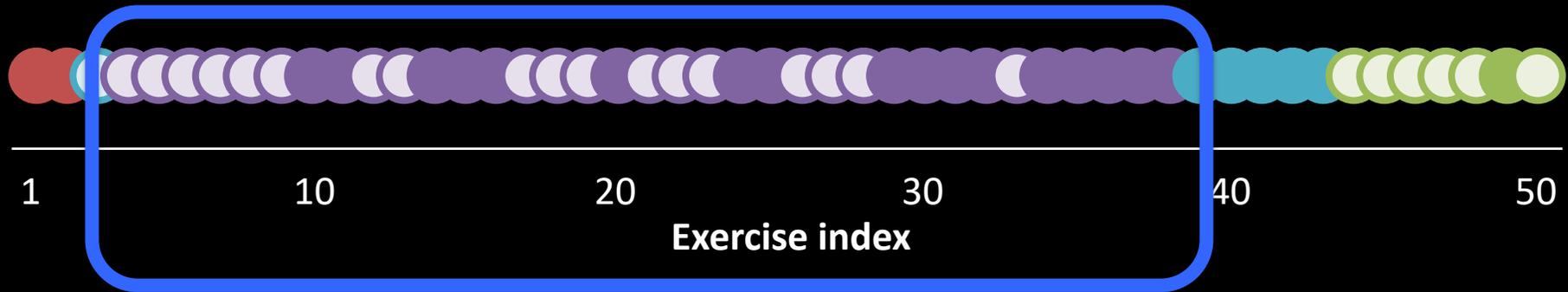
-  Solving for x-intercept
-  Solving for y-intercept
-  Graphing linear equations
-  Square roots
-  Slope of a line

Answer:

-  Correct
-  Incorrect



Story of Riley



Exercise Type:

-  Solving for x-intercept
-  Solving for y-intercept
-  Graphing linear equations
-  Square roots
-  Slope of a line

Answer:

-  Correct
-  Incorrect



Story of Riley



Exercise Type:

-  Solving for x-intercept
-  Solving for y-intercept
-  Graphing linear equations
-  Square roots
-  Slope of a line

Answer:

-  Correct
-  Incorrect



Story of Riley



Exercise Type:

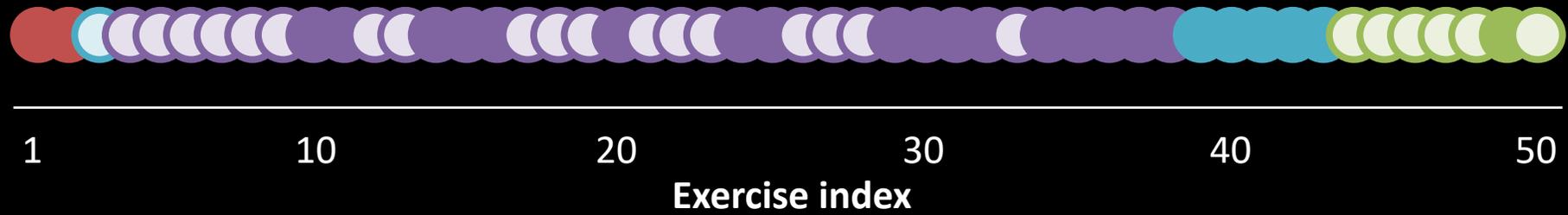
-  Solving for x-intercept
-  Solving for y-intercept
-  Graphing linear equations
-  Square roots
-  Slope of a line

Answer:

-  Correct
-  Incorrect



Story of Riley



What does Riley know?

Exercise Type:

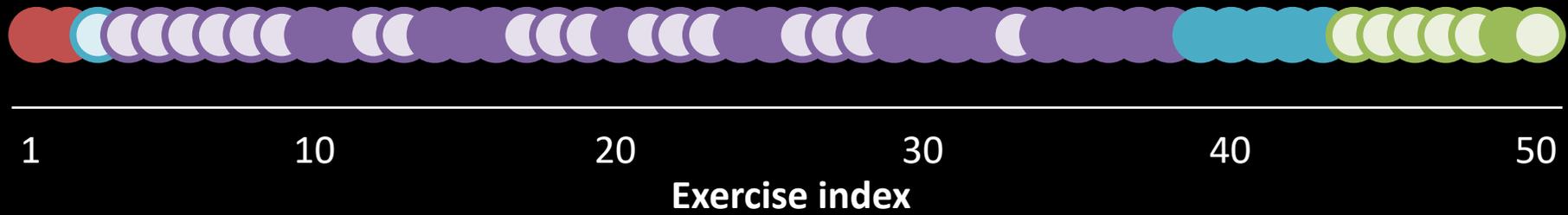
-  Solving for x-intercept
-  Solving for y-intercept
-  Graphing linear equations
-  Square roots
-  Slope of a line

Answer:

-  Correct
-  Incorrect



Story of Riley



What should Riley do next?

Exercise Type:

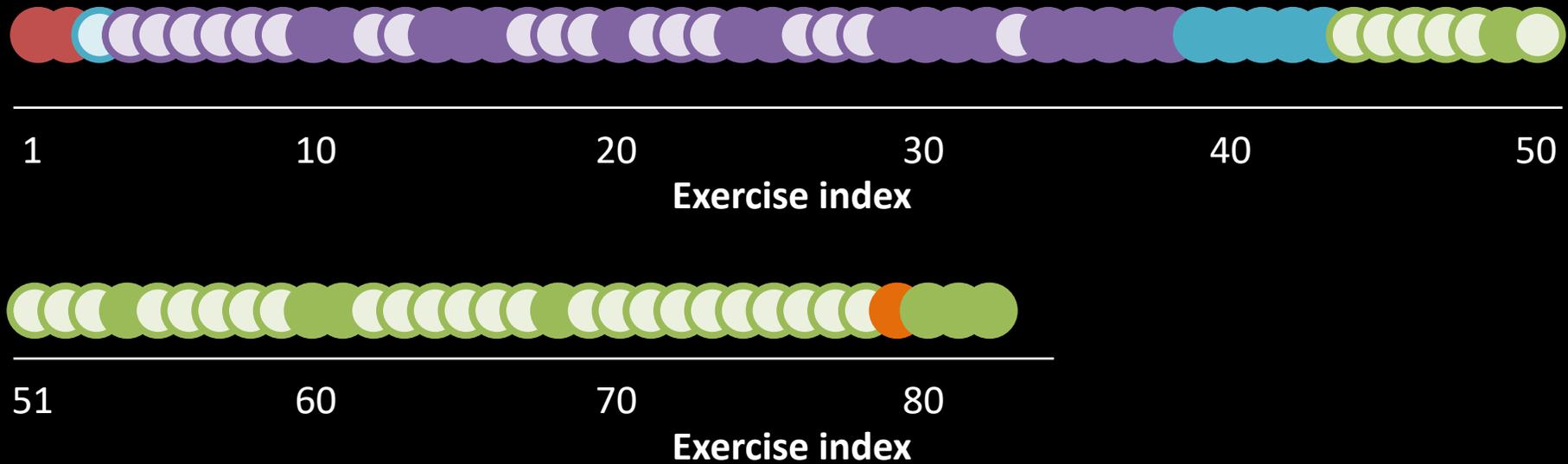
-  Solving for x-intercept
-  Solving for y-intercept
-  Graphing linear equations
-  Square roots
-  Slope of a line

Answer:

-  Correct
-  Incorrect



Story of Riley



Exercise Type:

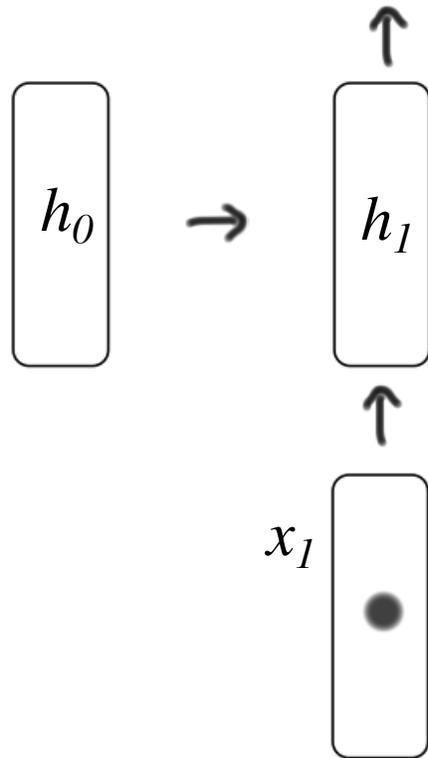
-  Solving for x-intercept
-  Solving for y-intercept
-  Graphing linear equations
-  Square roots
-  Slope of a line

Answer:

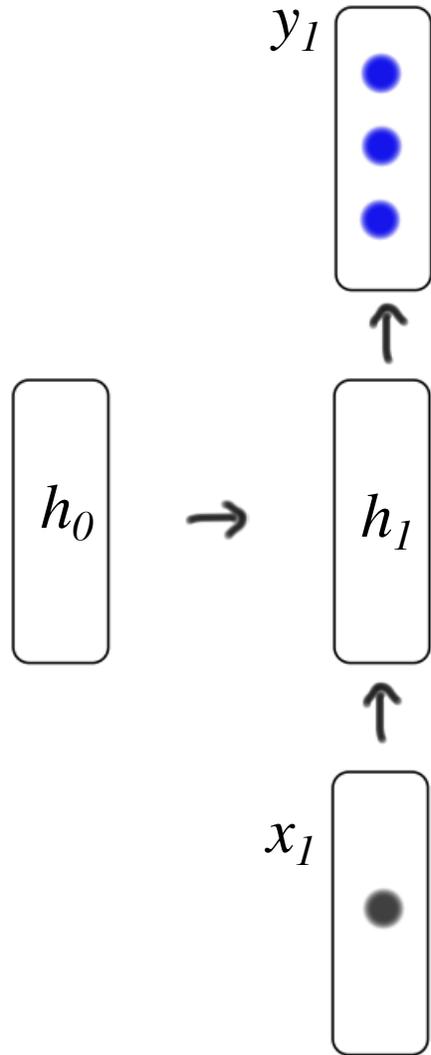
-  Correct
-  Incorrect



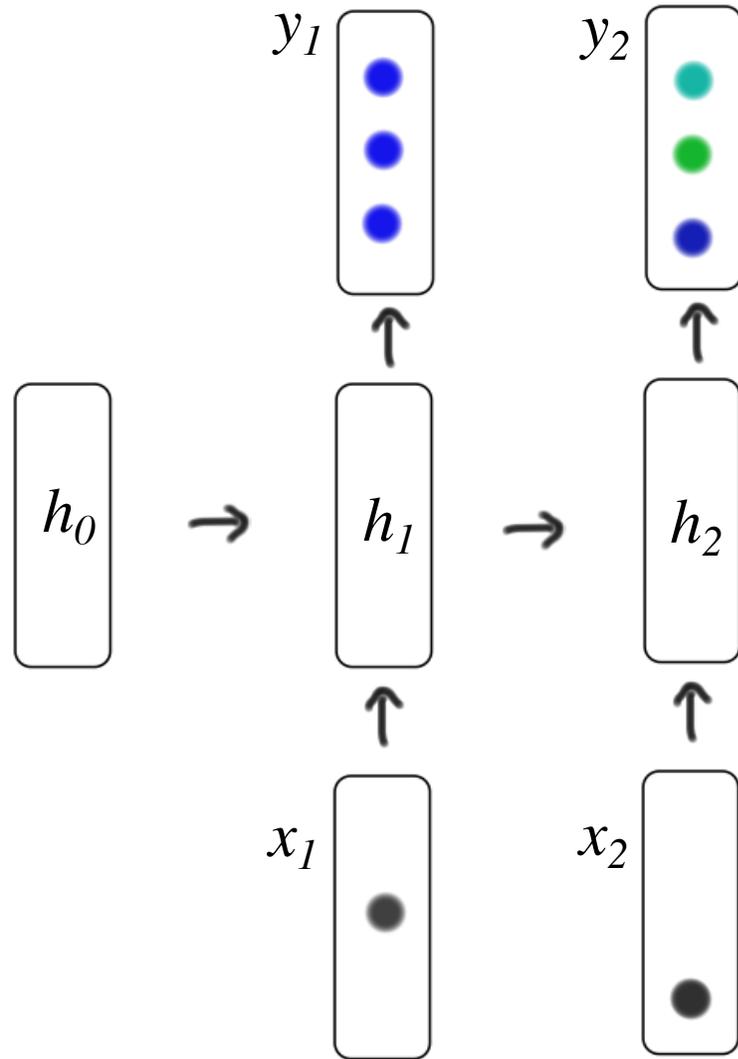
Recurrent Neural Network



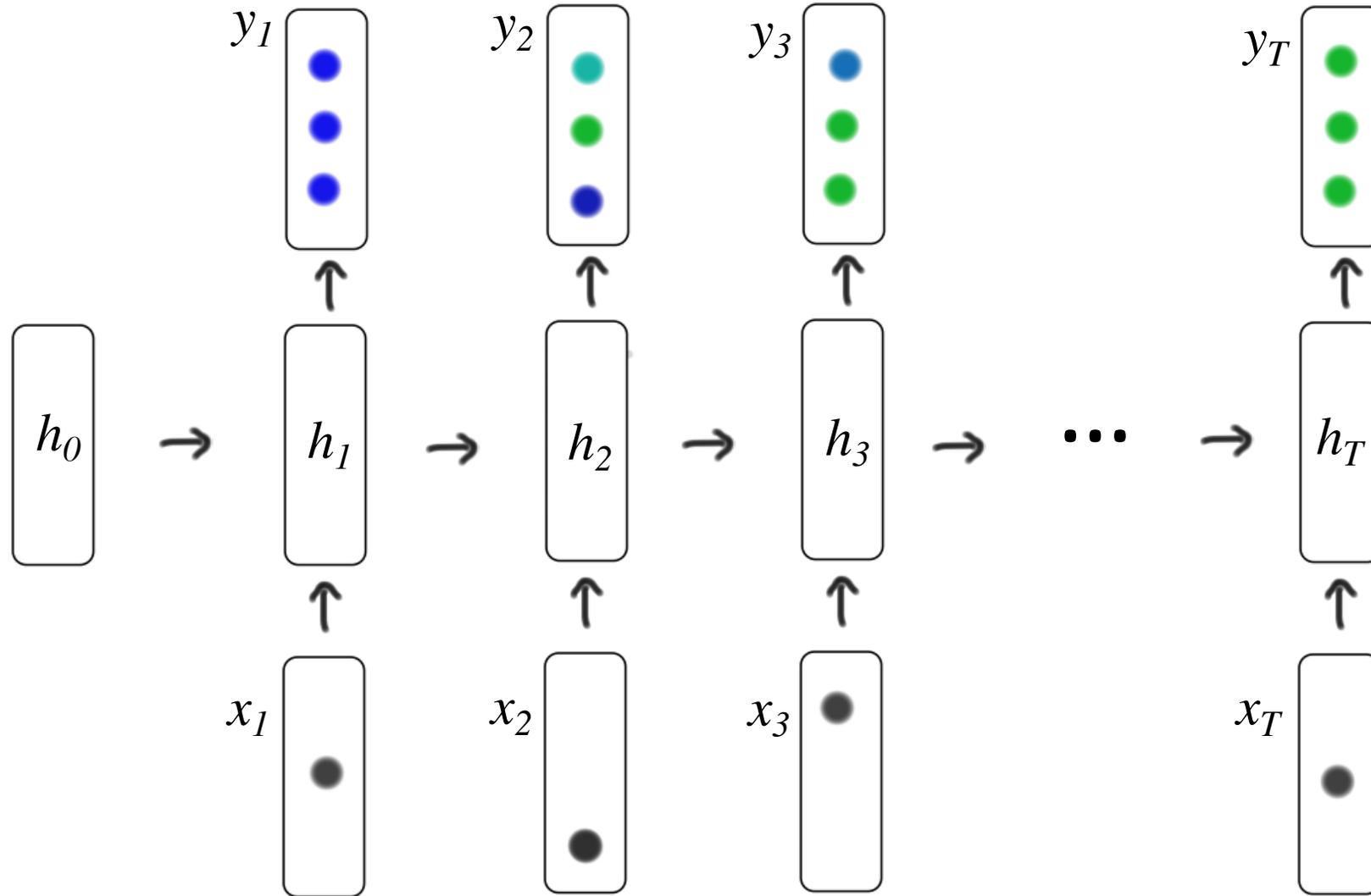
Recurrent Neural Network



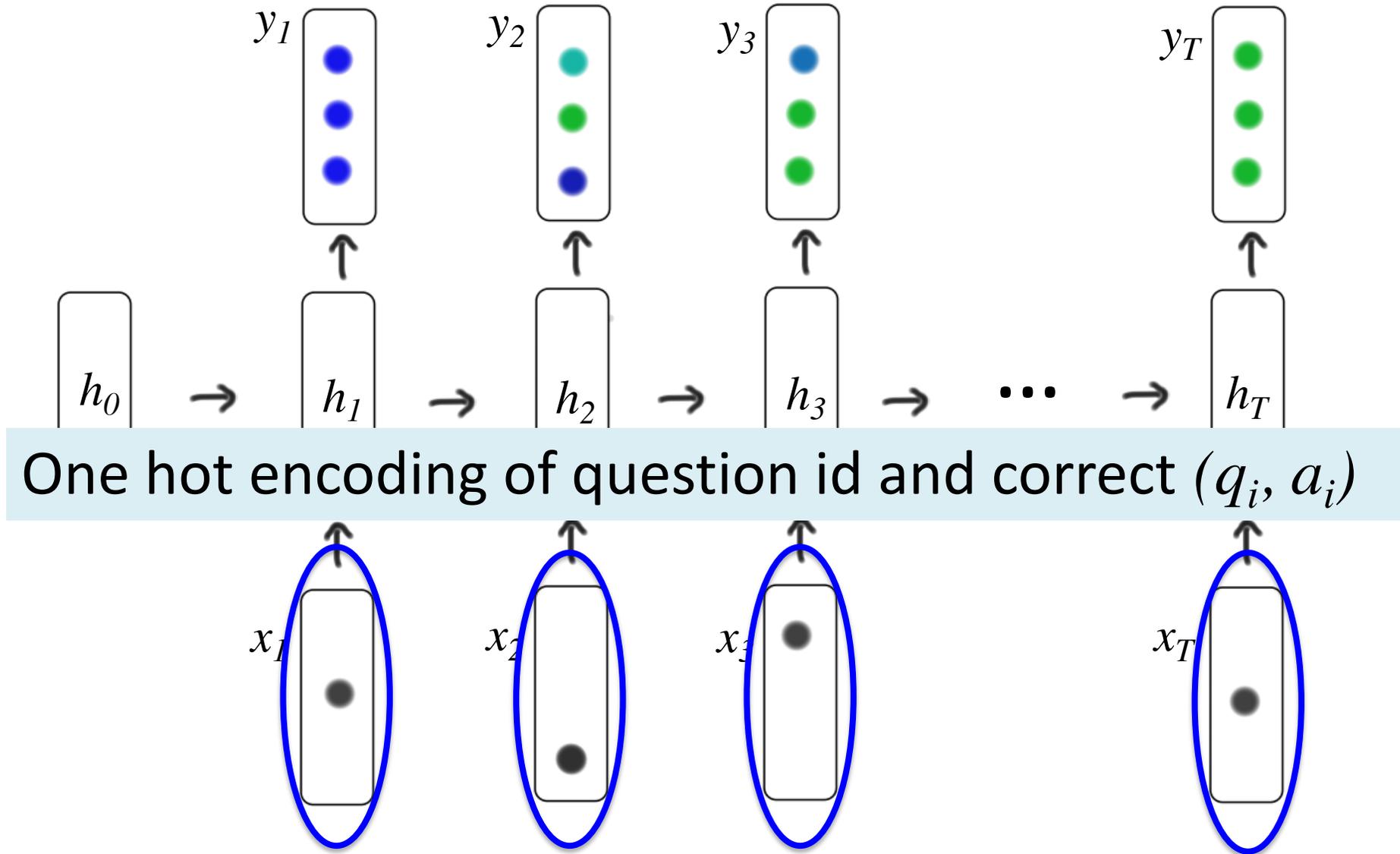
Recurrent Neural Network



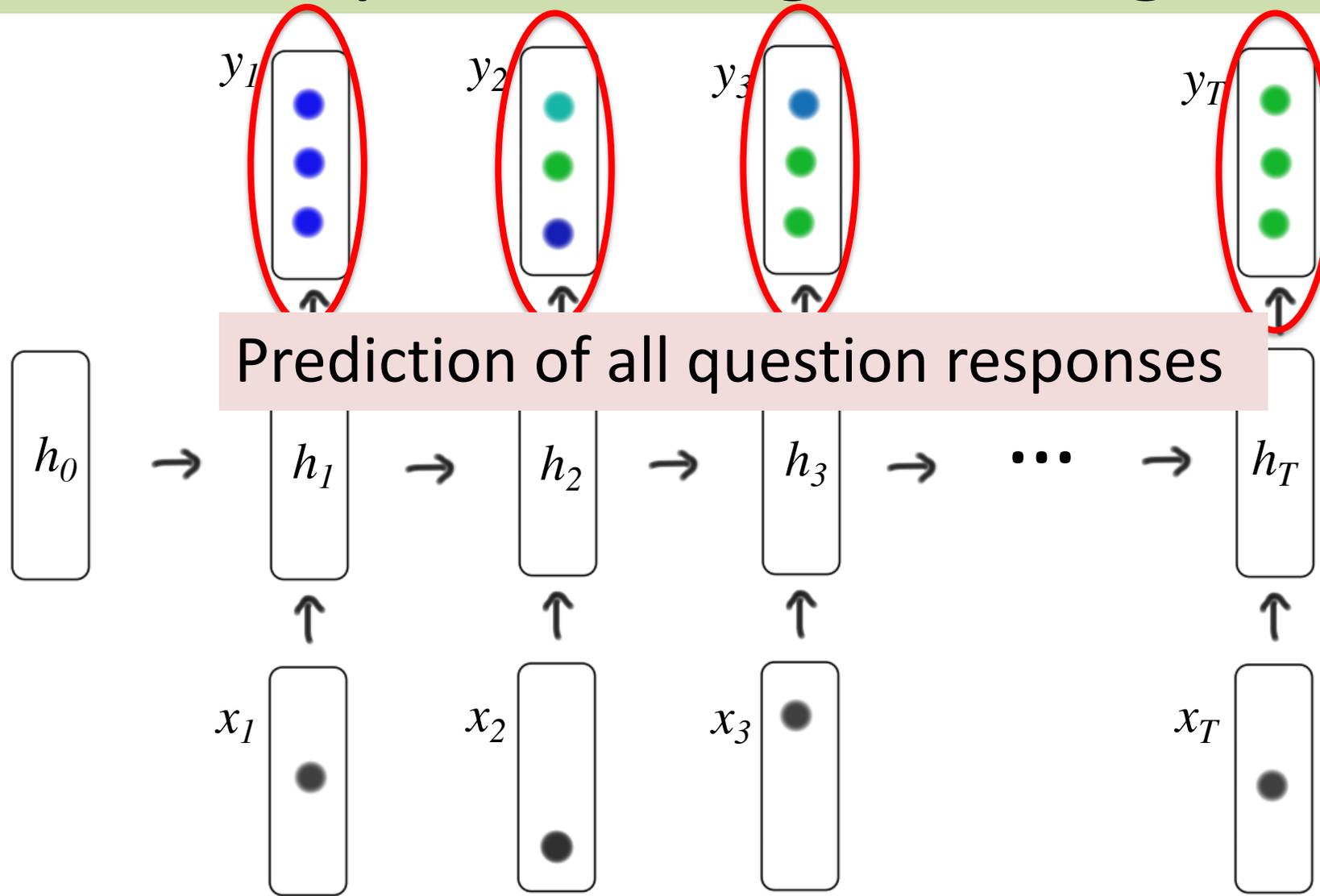
Recurrent Neural Network



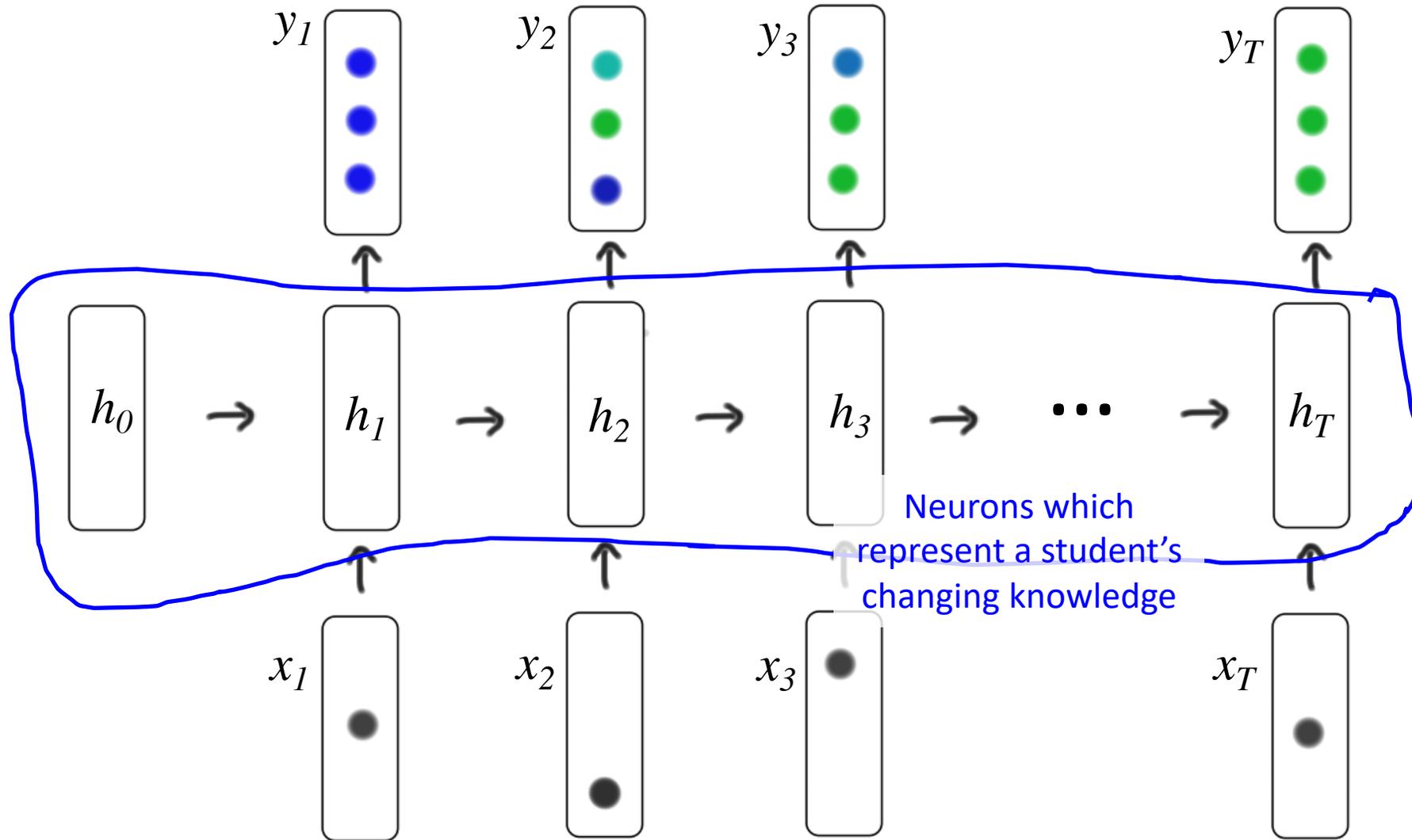
Deep Knowledge Tracing



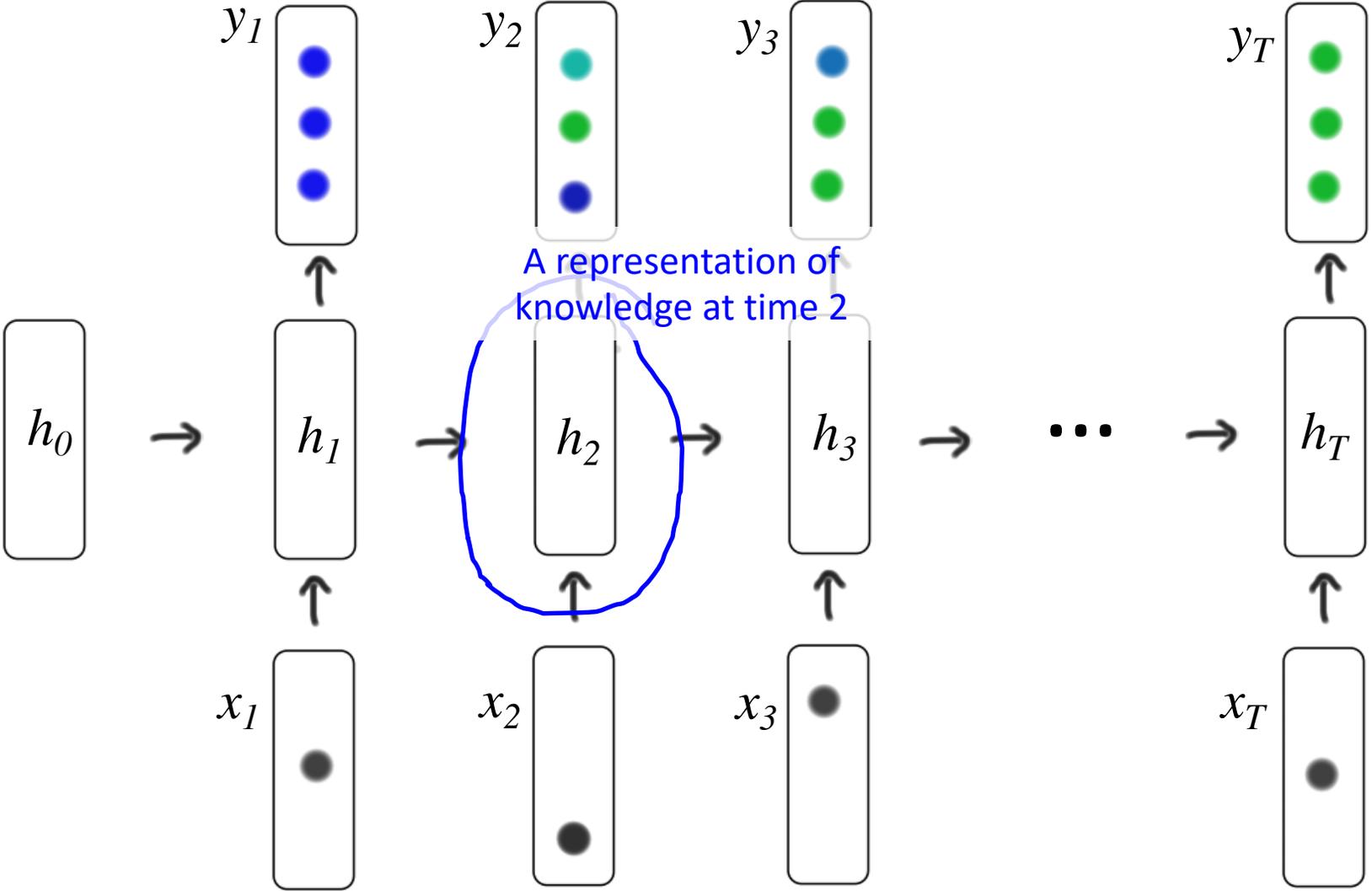
Deep Knowledge Tracing



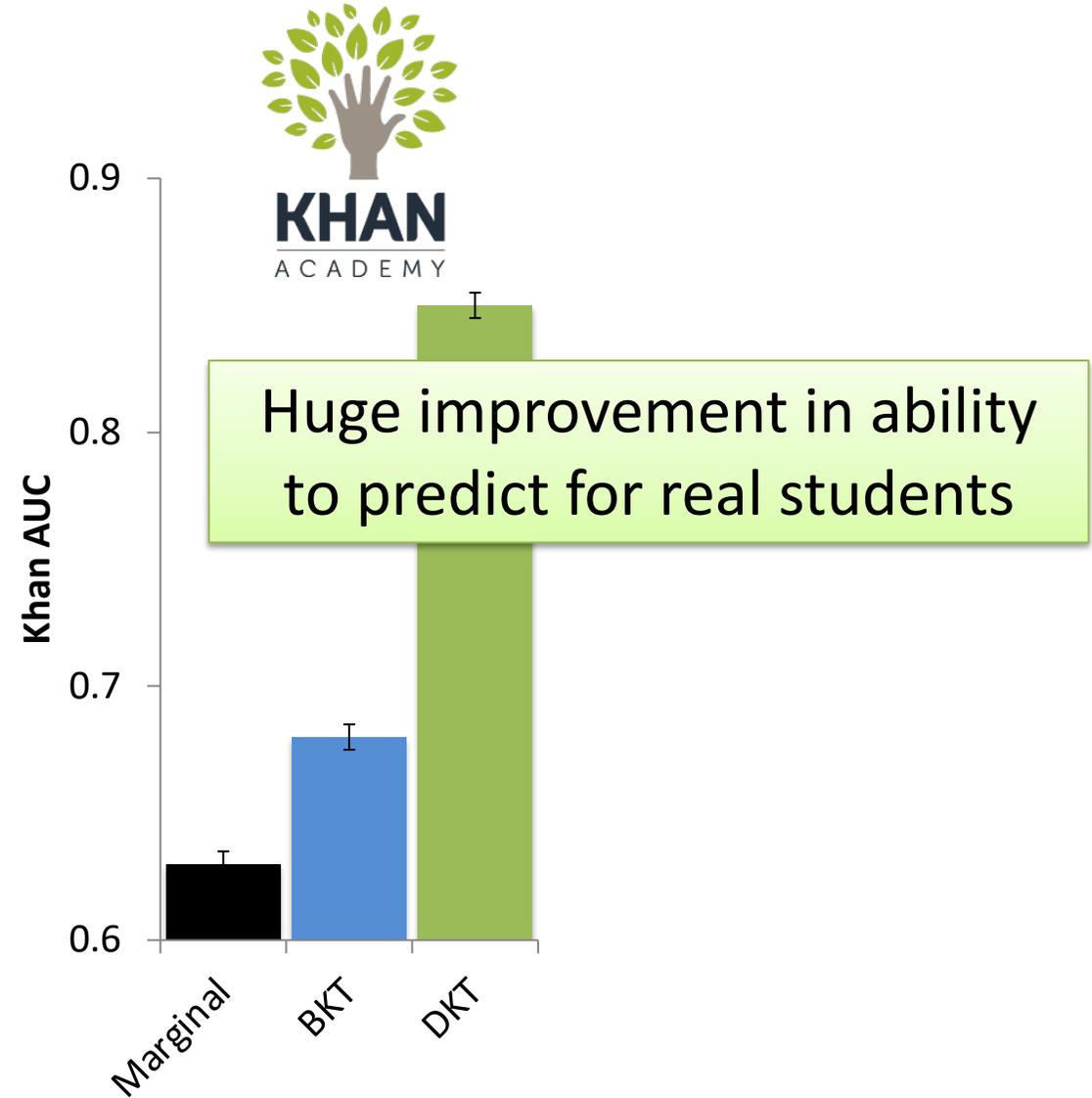
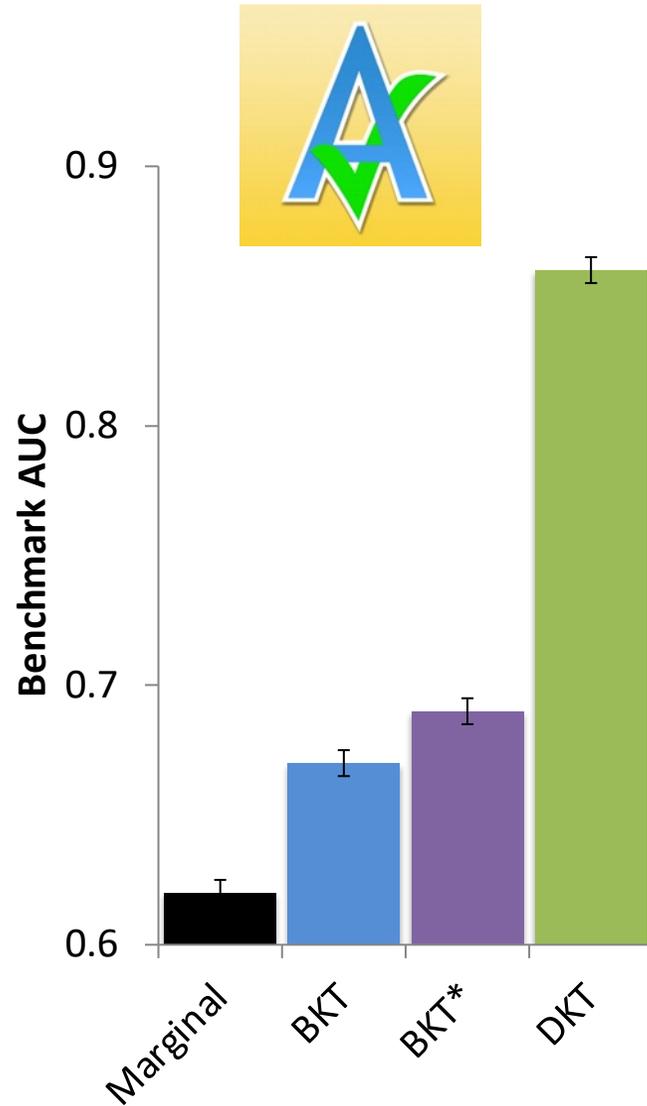
Recurrent Neural Network



Recurrent Neural Network



Prediction Results

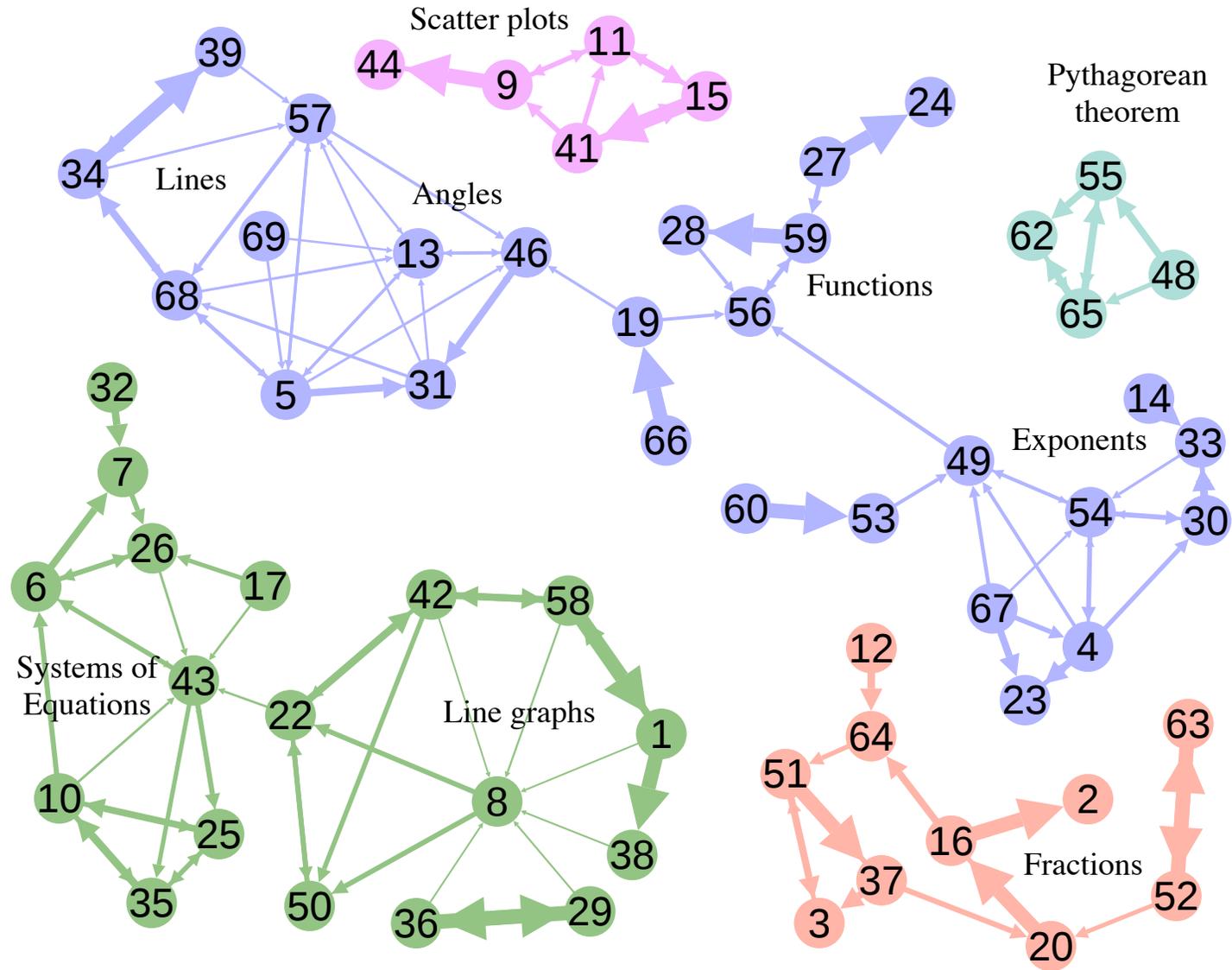


Interpretation

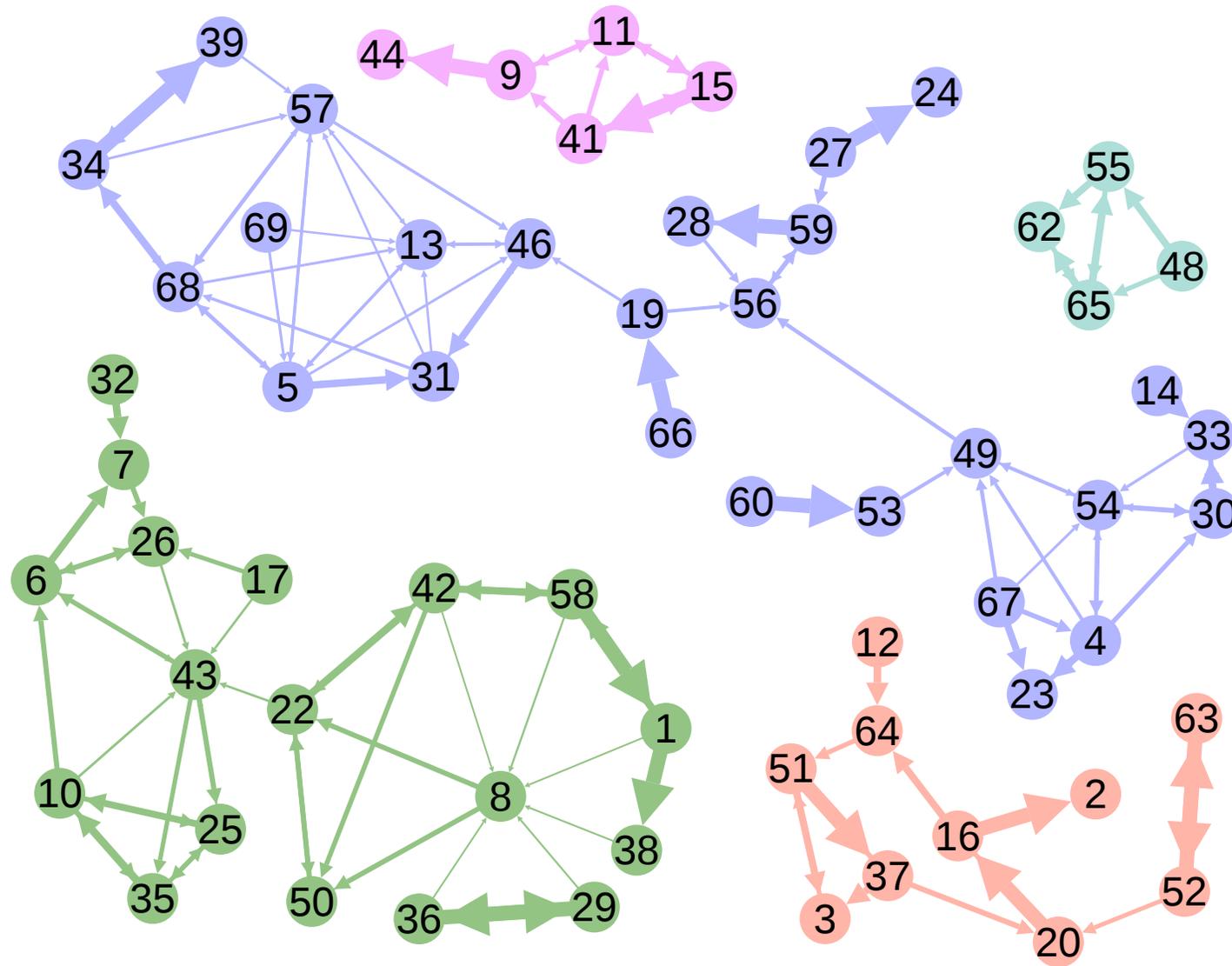
For each pair of concepts (X, Y)
estimate:

$$P(Y = \text{correct} \mid X = \text{correct})$$

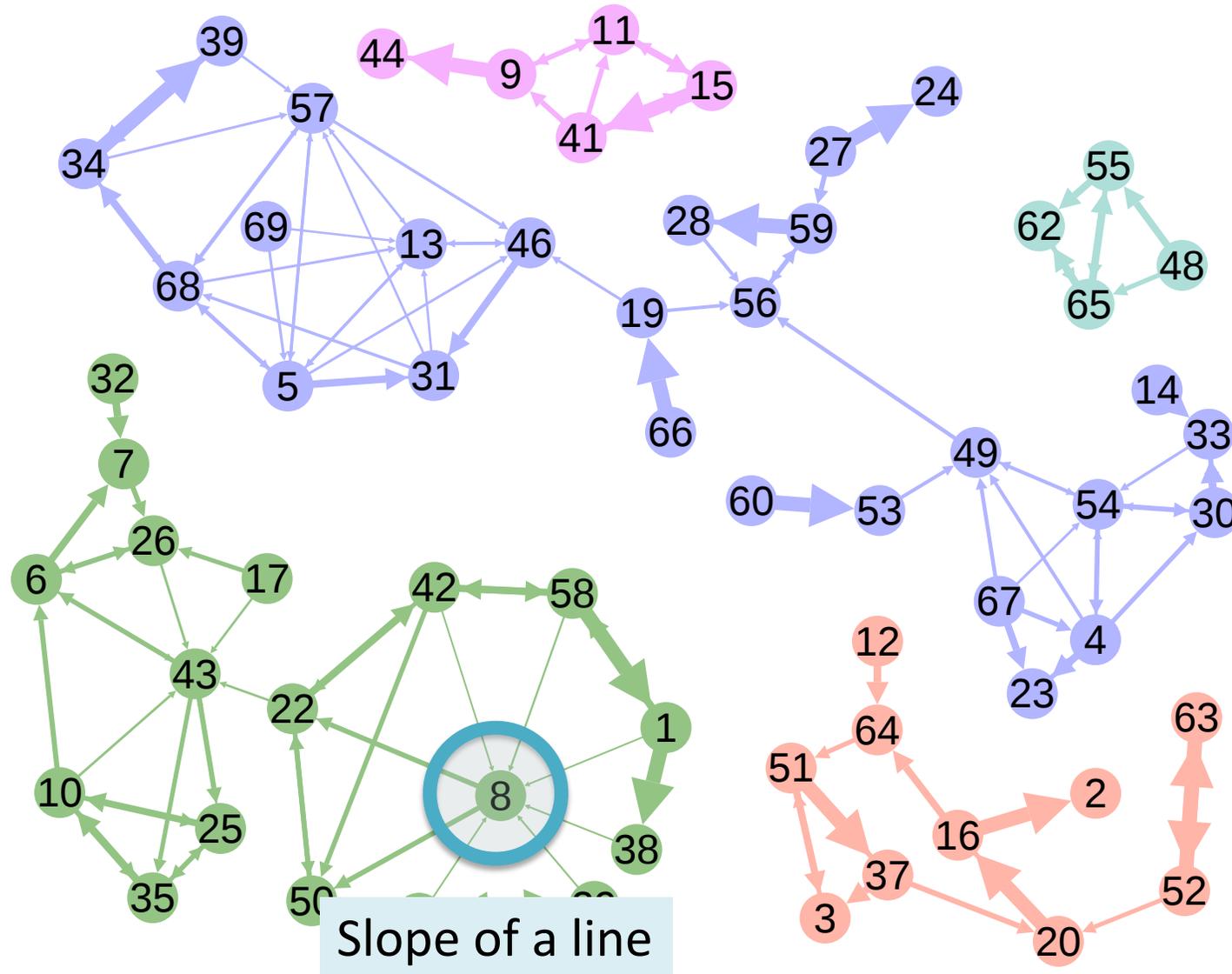
Learns Concept Relationships



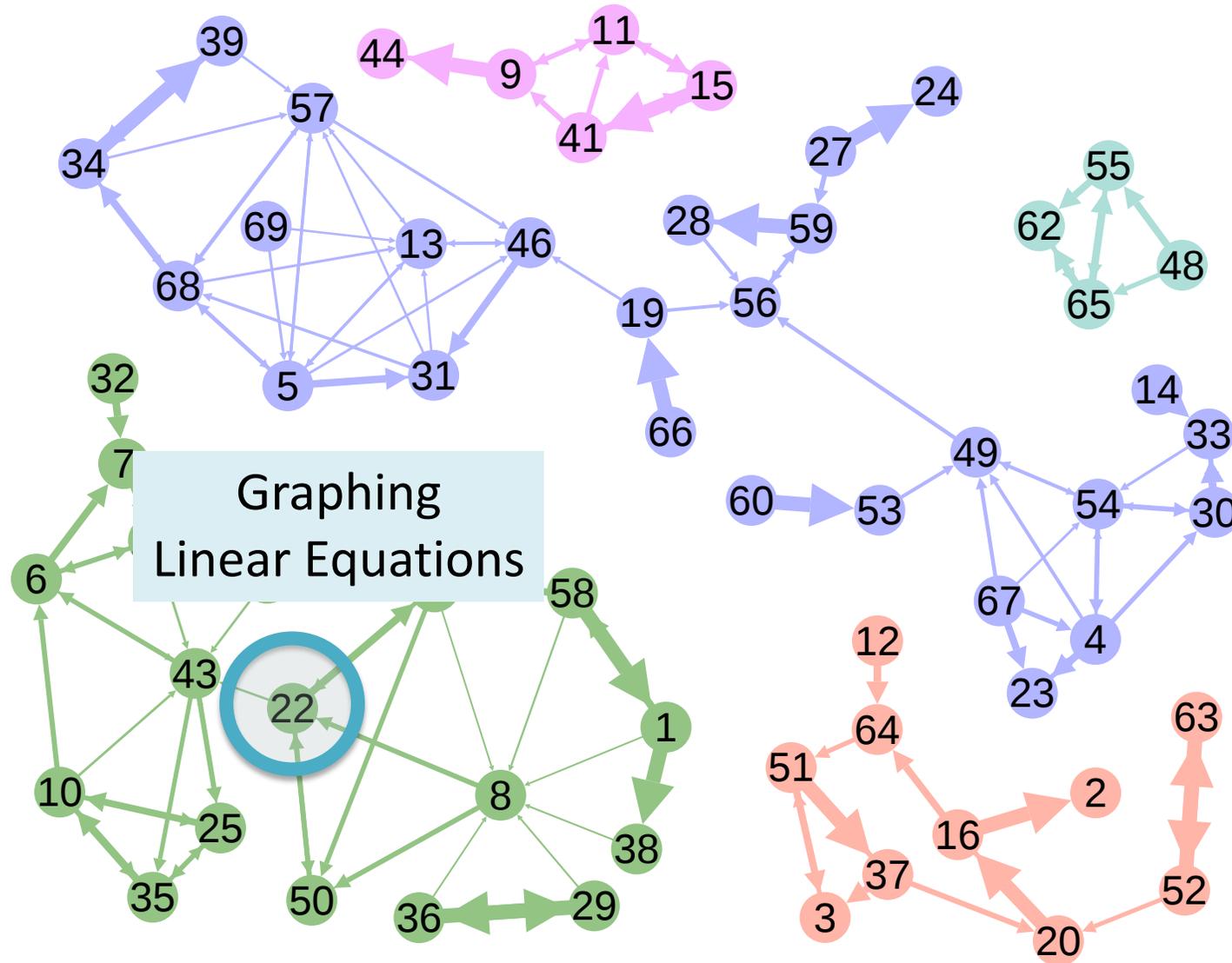
Learns Concept Relationships



Learns Concept Relationships



Learns Concept Relationships



The future of education is not multiple choice
questions...

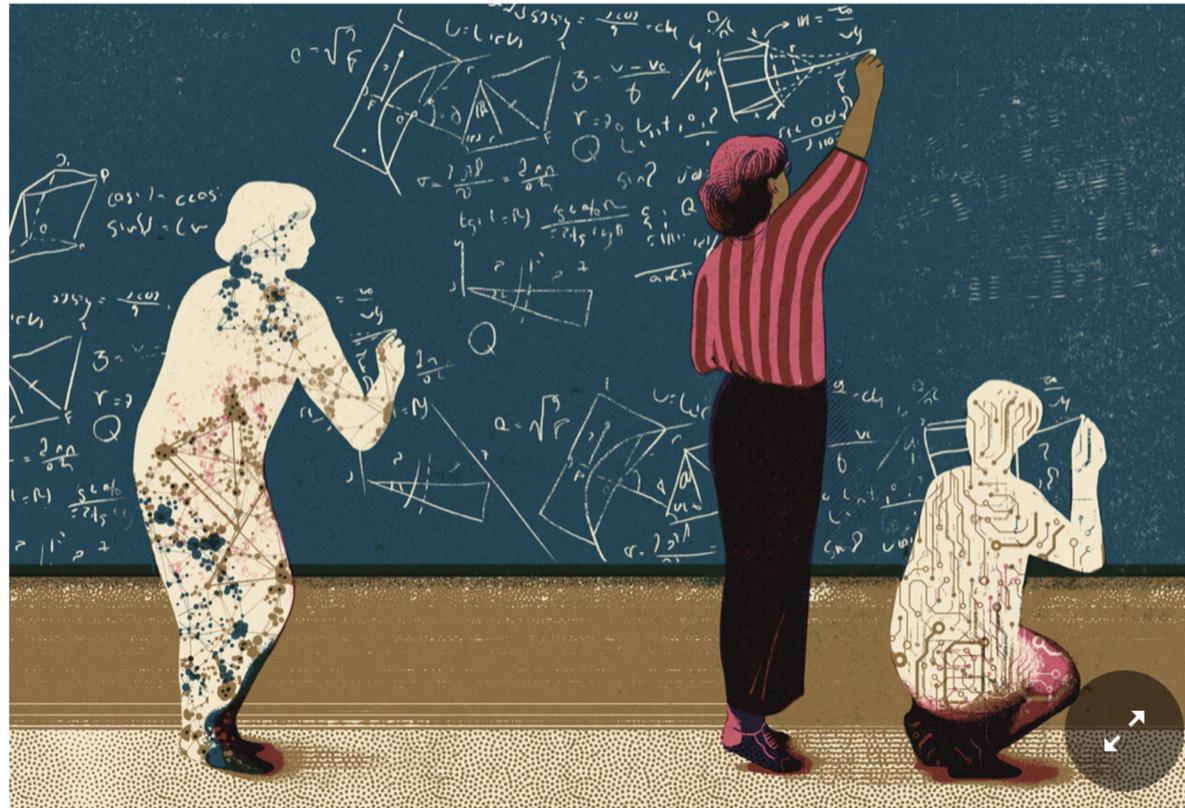
The
New York
Times

Can A.I. Grade Your Next Test?

Neural networks could give online education a boost by providing automated feedback to students.



Stanford | News



Give Feedback to Open Ended Work?



Human Level

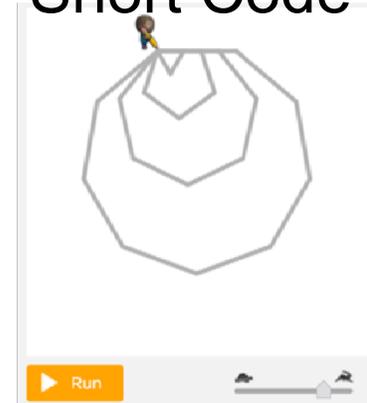
Simple Algebra

$$\begin{aligned} & 9 \times 6 \\ &= (10 - \boxed{}) \times 6 \\ &= 10 \times 6 - \boxed{} \times 6 \\ &= 60 - \boxed{} \\ &= \boxed{} \end{aligned}$$

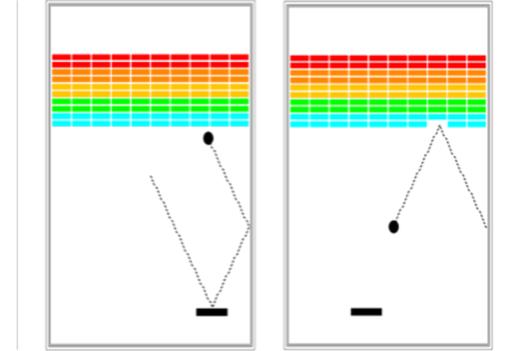
Short Answer

Why did the original colonists come to America?

Short Code



Long Code

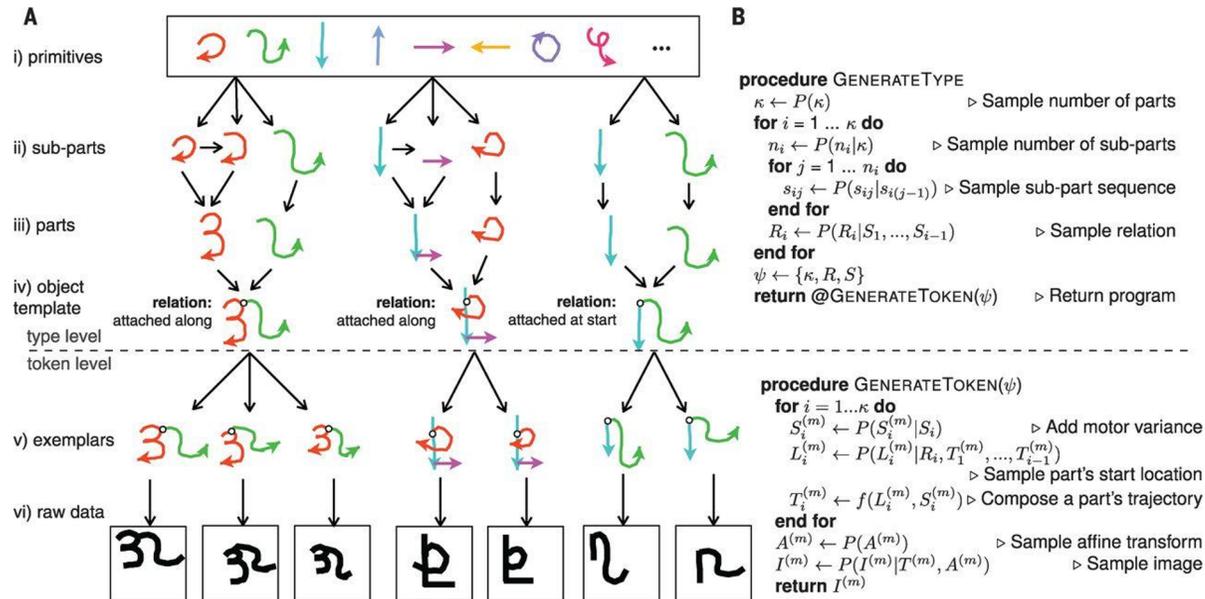


Deep Learning
Circa 2019

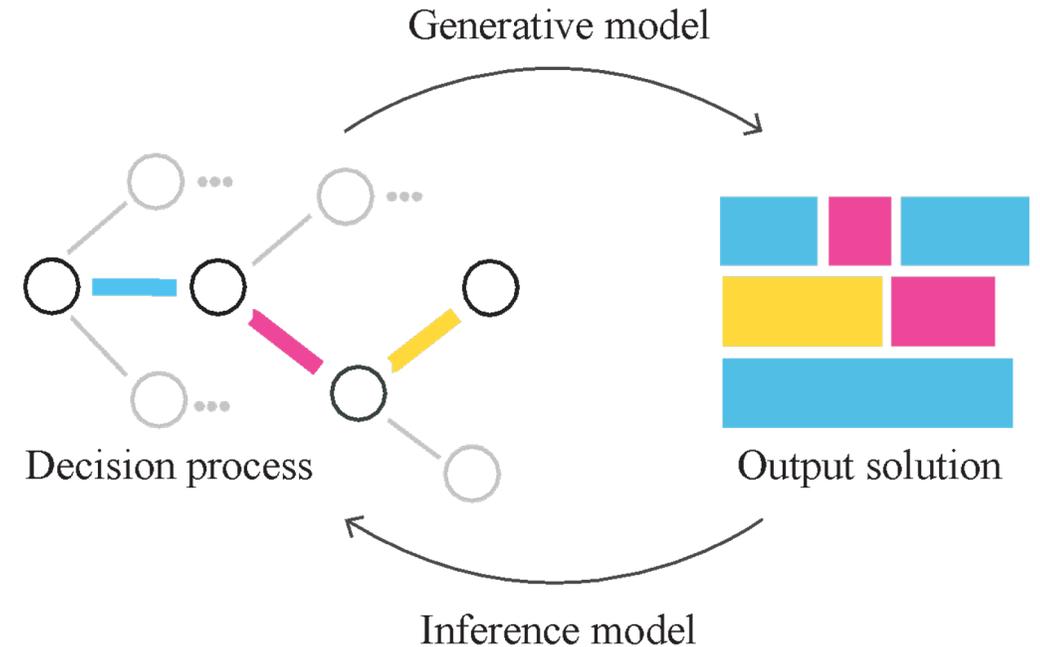


Generative Model of Grading

Lake et al, 2015



Our Team, EDM 2021



Generative Grading: Near Human-level Accuracy for Automated Feedback on Richly Structured Problems
 Ali Malik, Mike Wu, Vrinda Vasavada, Jinpeng Song, Madison Coats, John Mitchell, Noah Goodman, Chris Piech

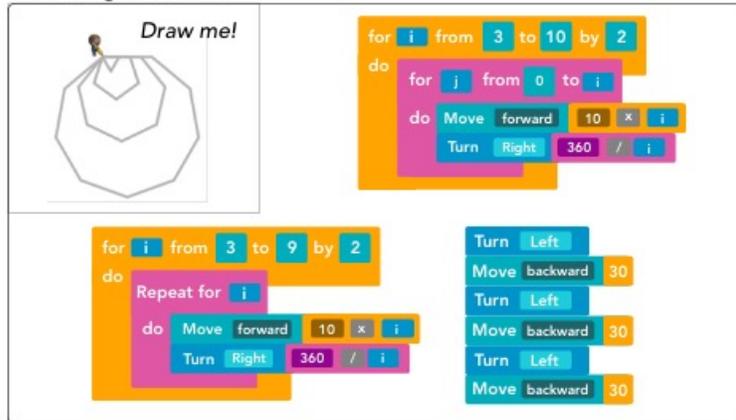
Stanford Univ



Generative Grading

(a) Datasets in Computational Education

Code.org Problem 8



Draw me!

```

for i from 3 to 10 by 2
do
  for j from 0 to i
  do
    Move forward 10 x i
    Turn Right 360 / i
  
```

```

for i from 3 to 9 by 2
do
  Repeat for i
  do
    Move forward 10 x i
    Turn Right 360 / i
  
```

```

Turn Left
Move backward 30
Turn Left
Move backward 30
Turn Left
Move backward 30

```

Powergrading P13

What is one reason the original colonists came to America?

- Religious freedom
- For religious freedom
- Freedom

- declared our independence from england
- religeous freedom
- as a criminal punishment

- to create a new colony
- to find better economic prospects
- to break away from the church in great britain

CS1: Liftoff

Write a Java Program to print the numbers 10 down to 1 and then write liftoff. You must use a loop.

```

public void run() {
  for (int i=START; i>0; i -=1)
  {
    println(i);
  }
  println("Liftoff");
}

```

```

public void run() {
  int x = START;
  int y = 1;
  int z = 9;
  while (x>=1) {
    println(x);
    x=z;
    z=x-y;
  }
  println("Liftoff");
}

```

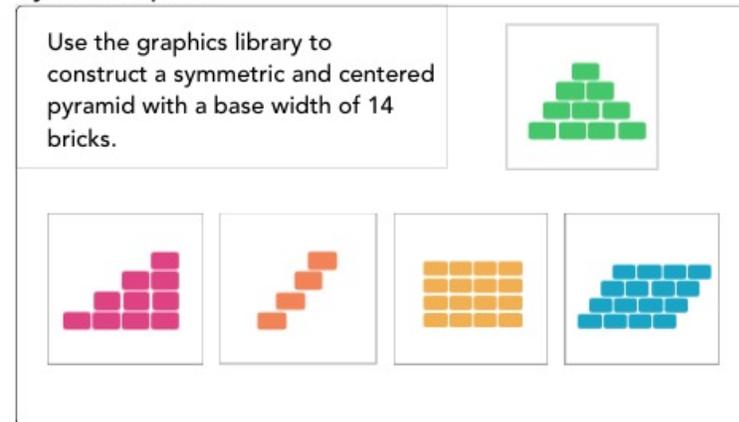
```

public void run() {
  for(int i = START; i>0; i--)
  {
    println(i);
    pause(1000);
  }
  println("Liftoff!");
}

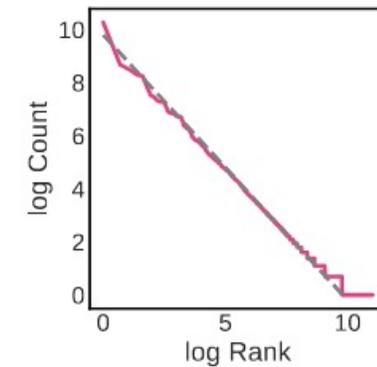
```

PyramidSnapshot

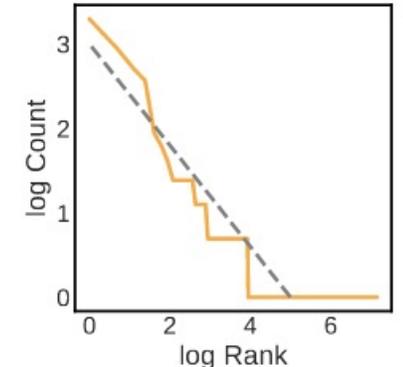
Use the graphics library to construct a symmetric and centered pyramid with a base width of 14 bricks.



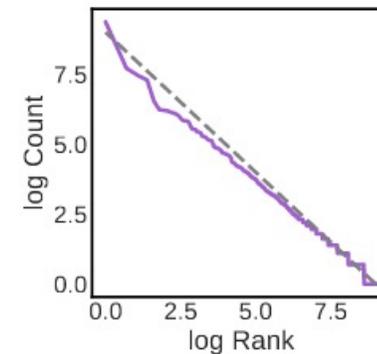
(b) Code.org P8



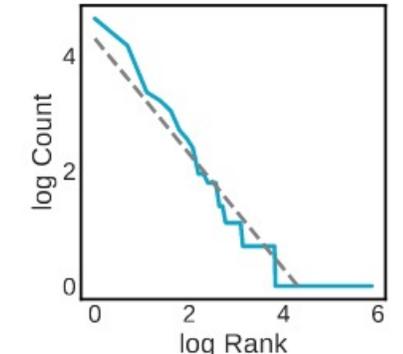
(c) CS1: Liftoff



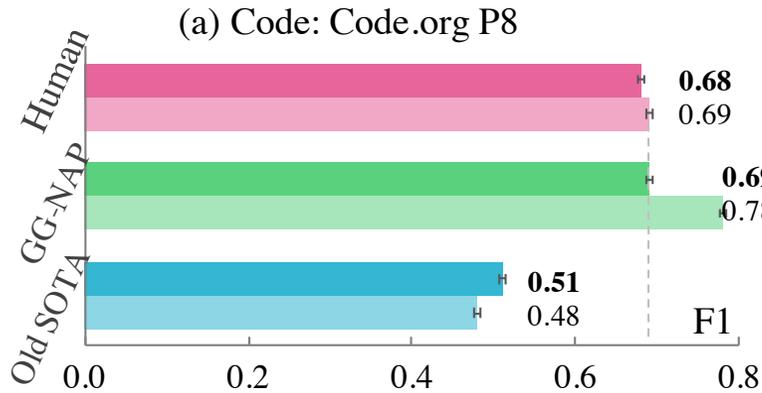
(d) Pyramid



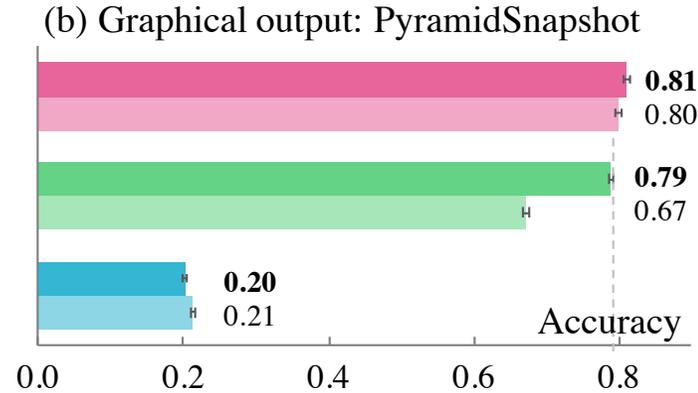
(e) Powergrading



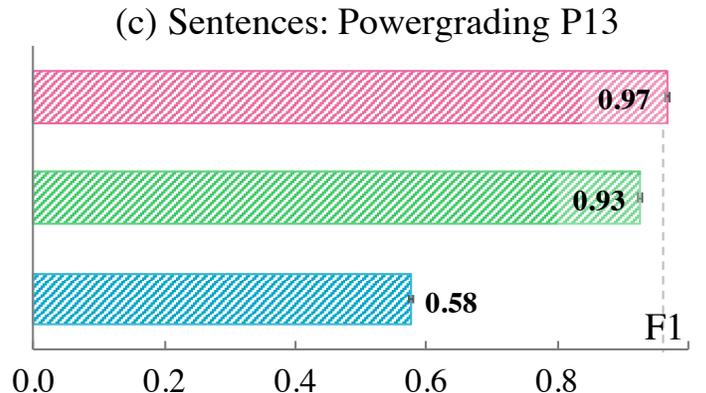
Generative Grading



Model	Body F1	Tail F1
Output CNN [17]	0.10	0.10
Program RNN [11]	0.27	0.22
MVAE [16]	0.38	0.26
Rubric Sampling [17]	0.51	0.48
GG-LSH	0.31	0.33
GG-NAP	0.69	0.78
Human	0.68	0.69



Model	Body F1	Tail F1
kNN [??]	0.20	0.12
NeuralNet [??]	0.20	0.21
GG-kNN	timeout	timeout
GG-NAP	0.79	0.67
Human	0.81	0.80



Model	Avg F1	Tail Acc
Handcrafted [5]	0.58	-
T&N Best [12]	0.55	-
GG-kNN	0.78	0.63
GG-NAP	0.93	0.76
Human	0.97	0.90

Give Feedback to Open Ended Work?



Human Level

Simple Algebra

$$\begin{aligned} & 9 \times 6 \\ &= (10 - \boxed{}) \times 6 \\ &= 10 \times 6 - \boxed{} \times 6 \\ &= 60 - \boxed{} \\ &= \boxed{} \end{aligned}$$

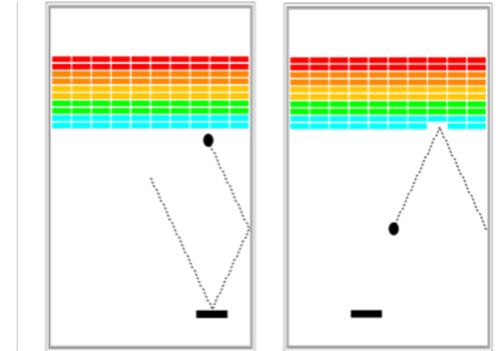
Short Answer

Why did the original colonists come to America?

Code.org



Stanford Midterm



Deep Learning
Circa 2019



Generative Grading

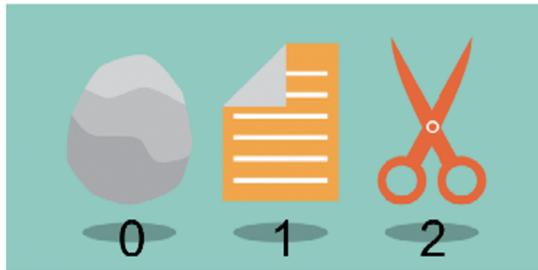


Too hard?

Midterm Grading Challenge

Question

Rock Paper Scissors (30 points)



Write a `ConsoleProgram` that has a user play rock paper scissors against a computer until either the user or the computer has **three** "wins". To make the code simpler, use integers to represent the different plays (0 is rock, 1 is paper, 2 is scissors). Example run:

```
RockPaperScissors [completed]
Rock Paper Scissors!
0) Rock
1) Paper
2) Scissors

Your move: 1
Computer move: 2
Computer wins.

Your move: 2
Computer move: 0
Computer wins.

Your move: 1
Computer move: 1
Human wins.

Your move: 1
Computer move: 2
```

← A brief intro message

← Round 1 The user's paper lost to the computer's scissors

← Round 2 The user's scissors lost to the computer's rock

← Round 3 The user's scissors beat the computer's paper

← Round 4 Both played paper

Student Answer

```
1 public class RockPaperScissors extends ConsoleProgram {
2
3     /* constants */
4     private static final int ROCK = 0;
5     private static final int PAPER = 1;
6     private static final int SCISSORS = 2;
7     private static final int N_WINS = 3;
8
9     private RandomGenerator rg = new RandomGenerator();
10
11    public void run() {
12        introMessage();
13        for (int i = 0; i < N_WINS; i++) {
14            inputNumber();
15            roundWinner();
16        }
17        gameWinner();
18    }
19
20    private void introMessage() {
21        println("Rock Paper Scissors!");
22        println("0) Rock");
23        println("1) Paper");
24        println("2) Scissors");
25        println(" ");
26    }
27
28    private void inputNumber() {
29        int a = readInt("Your move: ");
30        int computerMove = rg.nextInt(3);
31        int b = readInt("Computer move: " + computerMove);
32
33
34        private int computer = 0;
35        private int human = 0;
36
37    private void roundWinner {
38        if (a == 1 && b == 0) {
39            println("Human wins.");
```

Feedback

Grade: 26/30 points ✓ grade submitted!

Rubric: Standard ▾



- For loop instead of a while loop (should go up to 3 wins by computer or user not 3 games)

User Inputting Move (3 points)

- Perfect (0 points)
- Minor Error (1 points)
- Major Error (2 points)
- Totally Wrong (3 points)

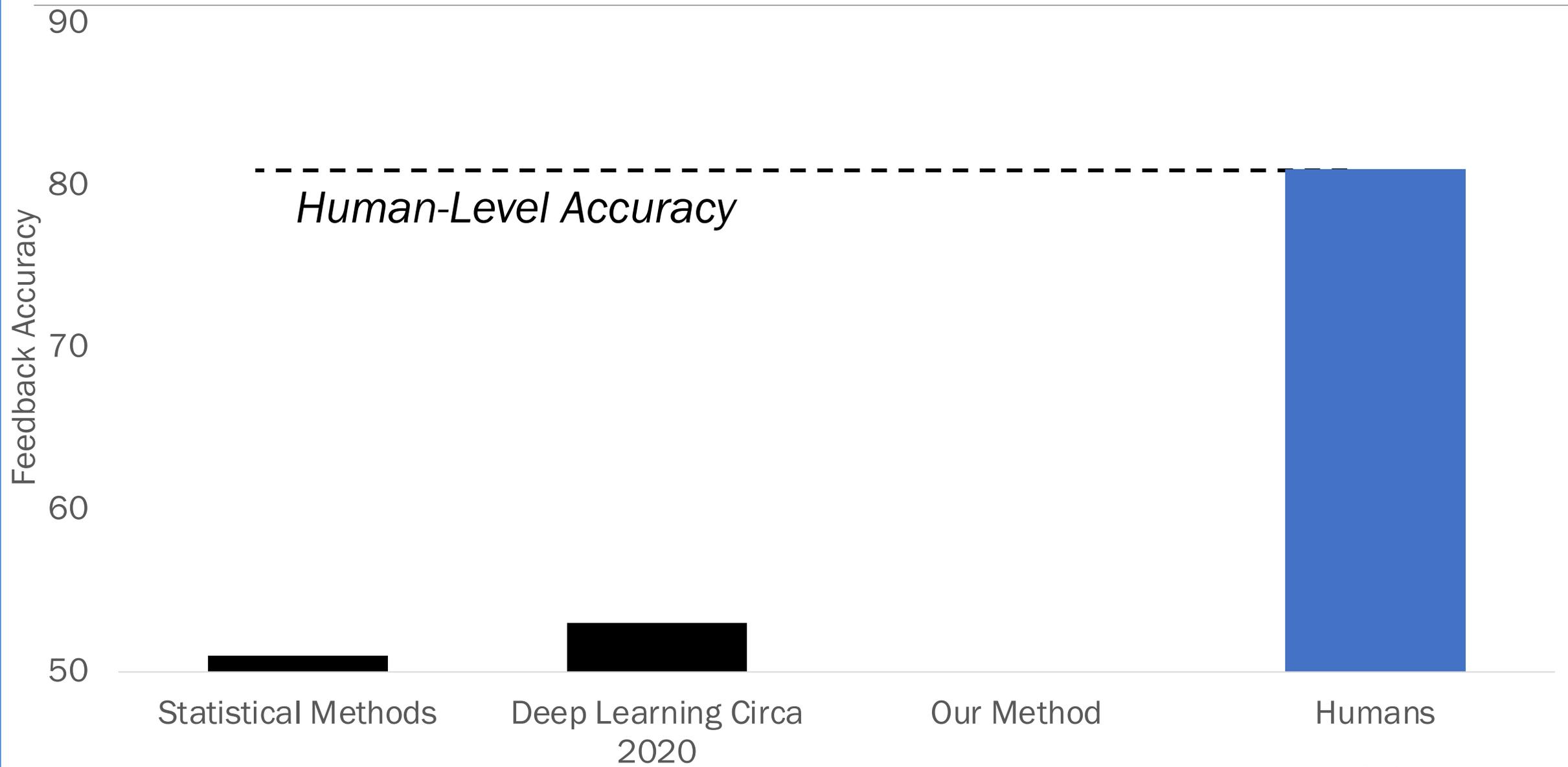
Computer Generating Move (4 points)

- Perfect (0 points)
- Minor Error (1 points)
- Major Error (2 points)
- Major Errors/No Attempt (4 points)

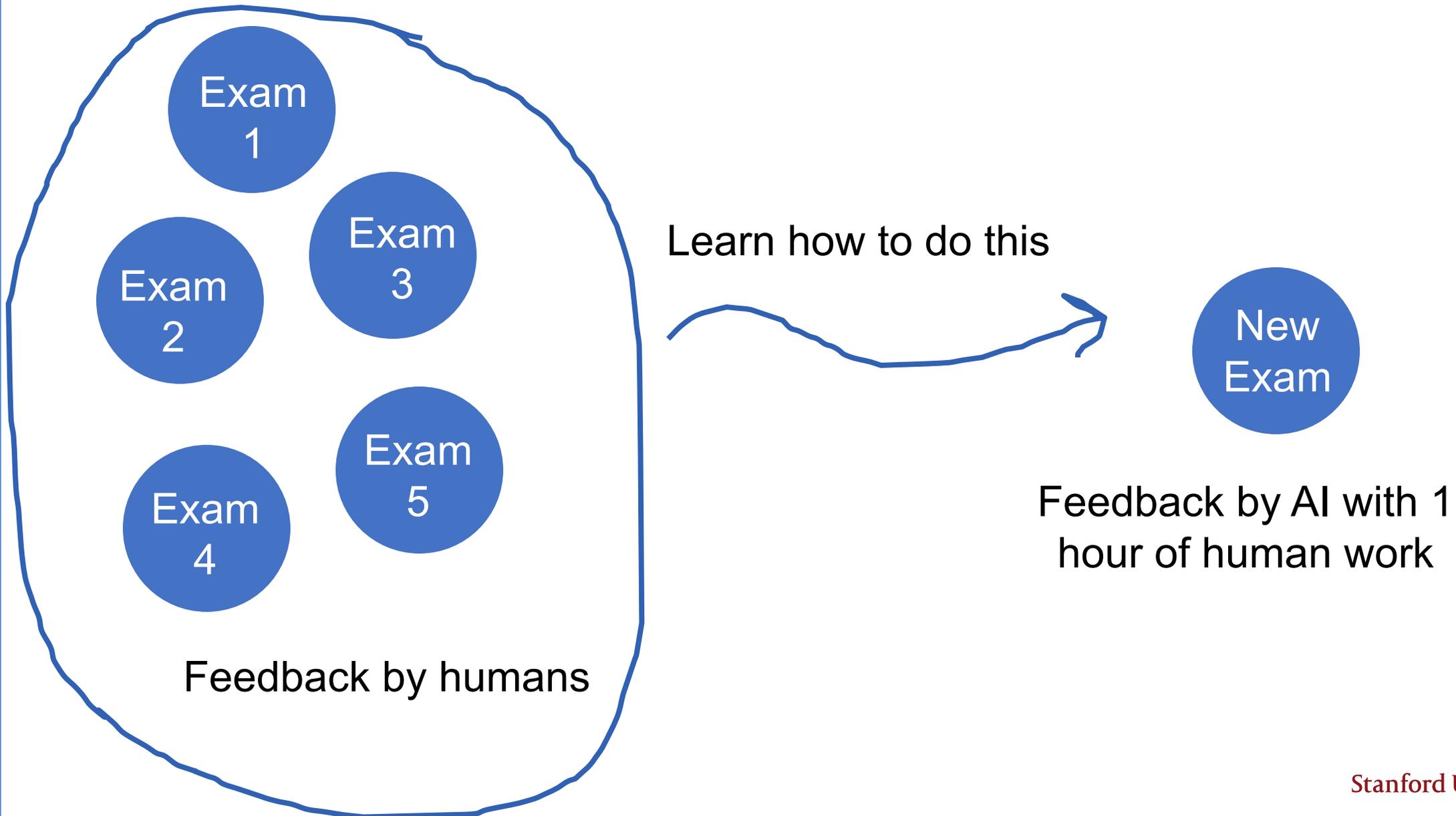
Determining the winner of a round (6 points)

- Perfect (0 points)
- Minor Error (1 points)
- Major Error (2 points)

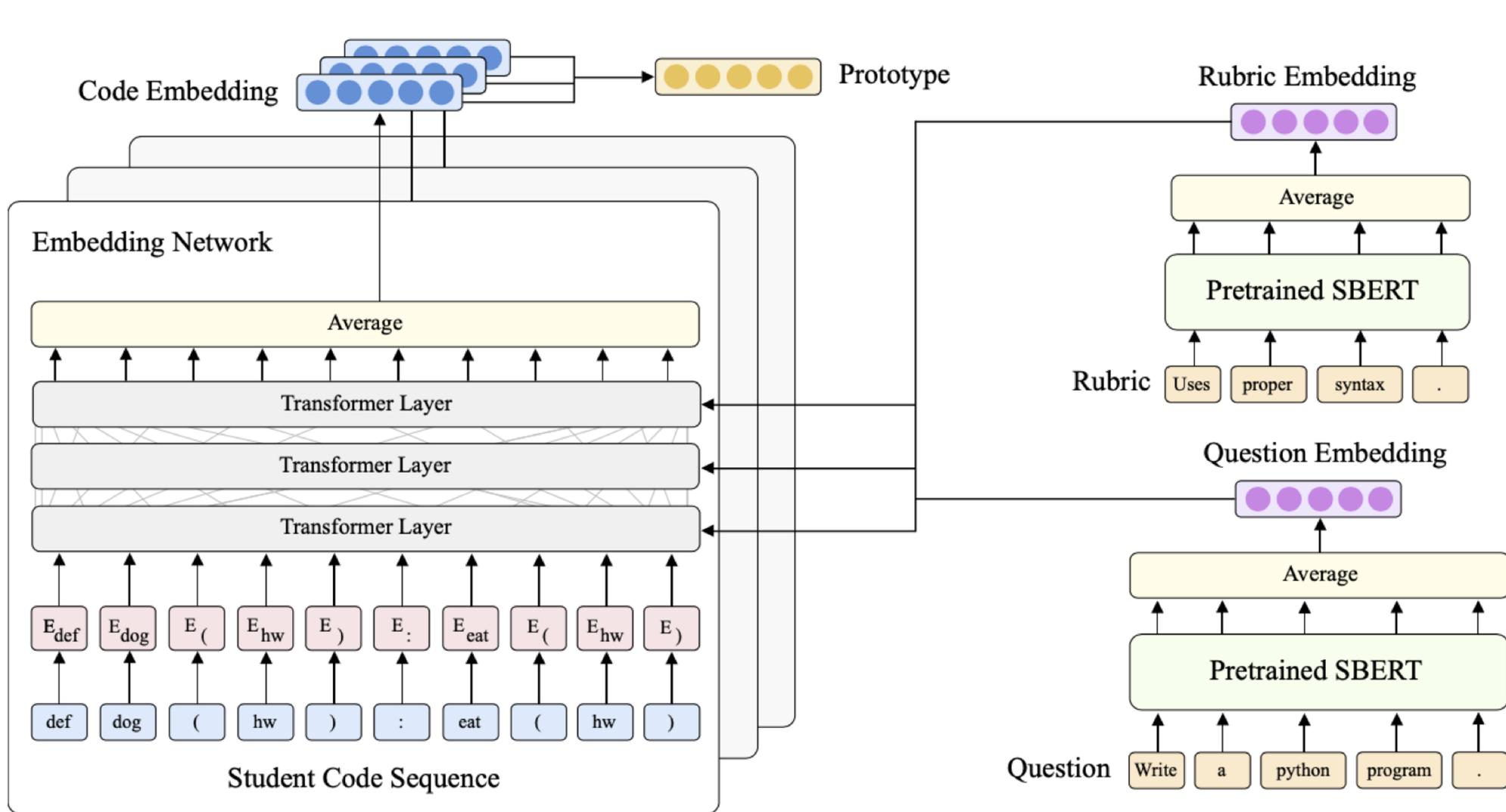
Rubric Level Accuracy on Few-Shot Grading a Novel Question



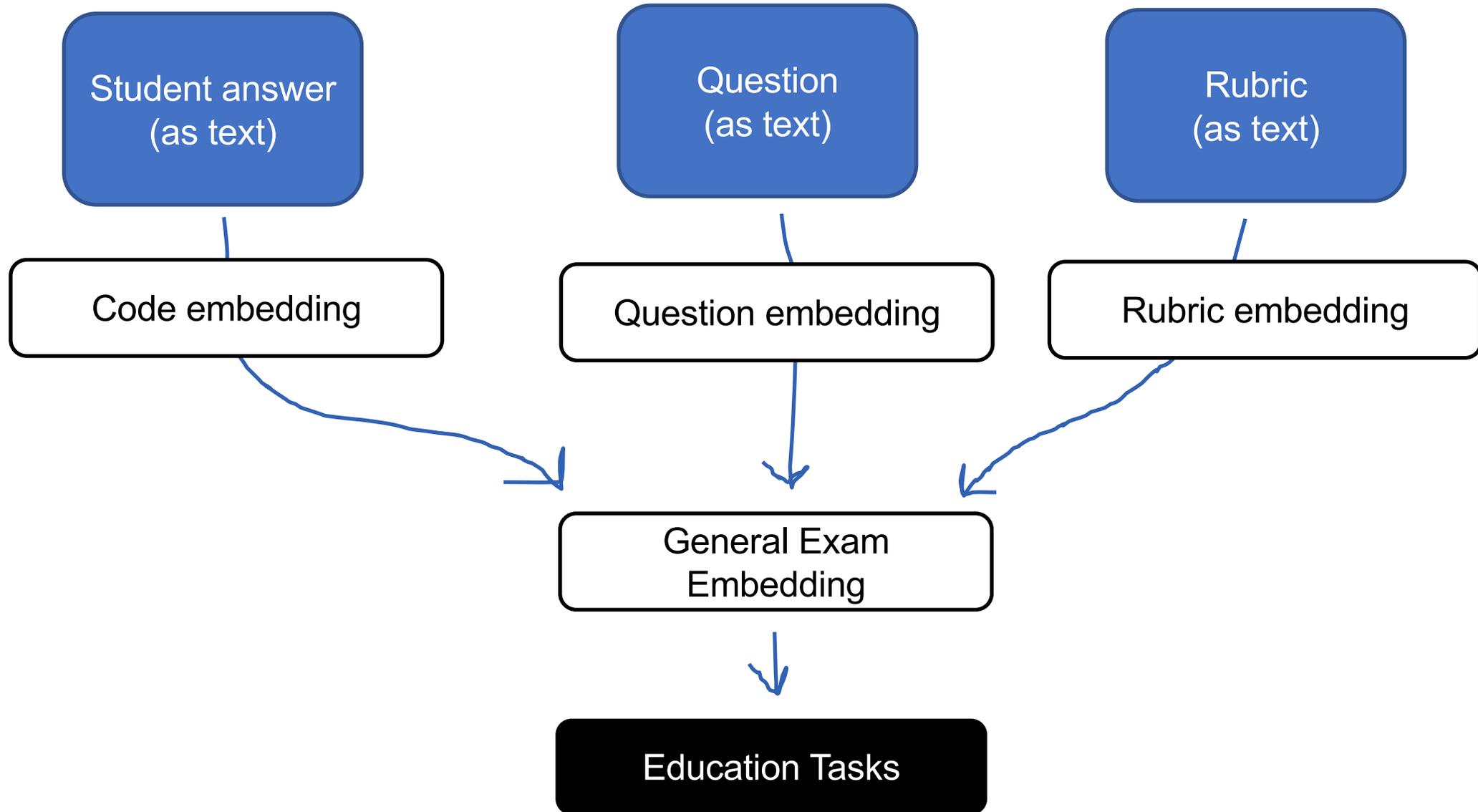
Give Feedback on Fresh Stanford Midterm



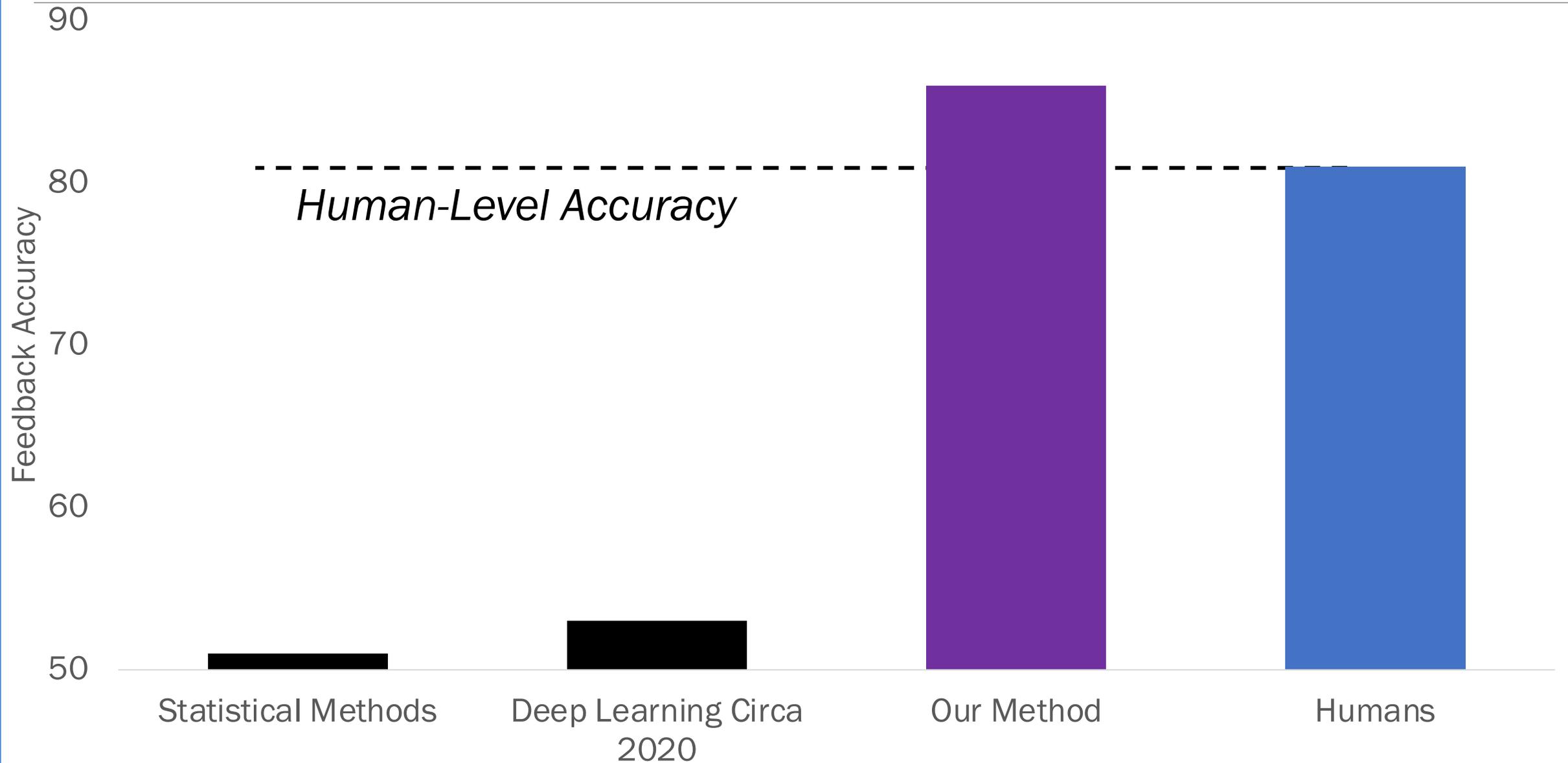
Invented the Proto-Transformer



General Exam Grading Model



Rubric Level Accuracy on Few-Shot Grading a Novel Question



Gave Feedback to 3,500 Real Students

Do you agree? AI feedback 97.9%. Human feedback 96.7%

The screenshot shows a web browser window titled "Code in Place Feedback" with the URL "codeinplace.stanford.edu/diagnostic/feedback". The page has a navigation bar with "Overview", "Question 1" (selected), "Question 2", "Question 3", "Question 4", "Question 5", and "Wrap-Up". Below the navigation bar are "Back", "Feedback", and "Next" buttons. The main content is split into two columns. The left column contains the question text: "GETTING INPUT FROM USER" and "This question requires you to get input from the user, convert it to a number, and save it as a variable. Did you correctly do all of these steps?". Below this is a purple feedback box: "Close. There is a minor error with your logic to get input from user. This could be something like forgetting to convert user input to a float". At the bottom of the left column are thumbs up/down icons and a "Please explain (optional):" field. The right column is titled "Your Solution" and contains Python code:

```
def main():  
    # TODO write your solution here  
    height=input("Enter your height in meters: ")  
    if height < 1.6:  
        print("Below minimum astronaut height")  
    if height > 1.9:  
        print("Above maximum astronaut height")  
    if height >= 1.6 and height <= 1.9:  
        print("Correct height to be an astronaut")  
  
if __name__ == "__main__":  
    main()
```

 Blue arrows point from external text to the code and feedback. One arrow points to the `height=input` line, another to the `if height >= 1.6 and height <= 1.9:` line, and a third to the feedback box.

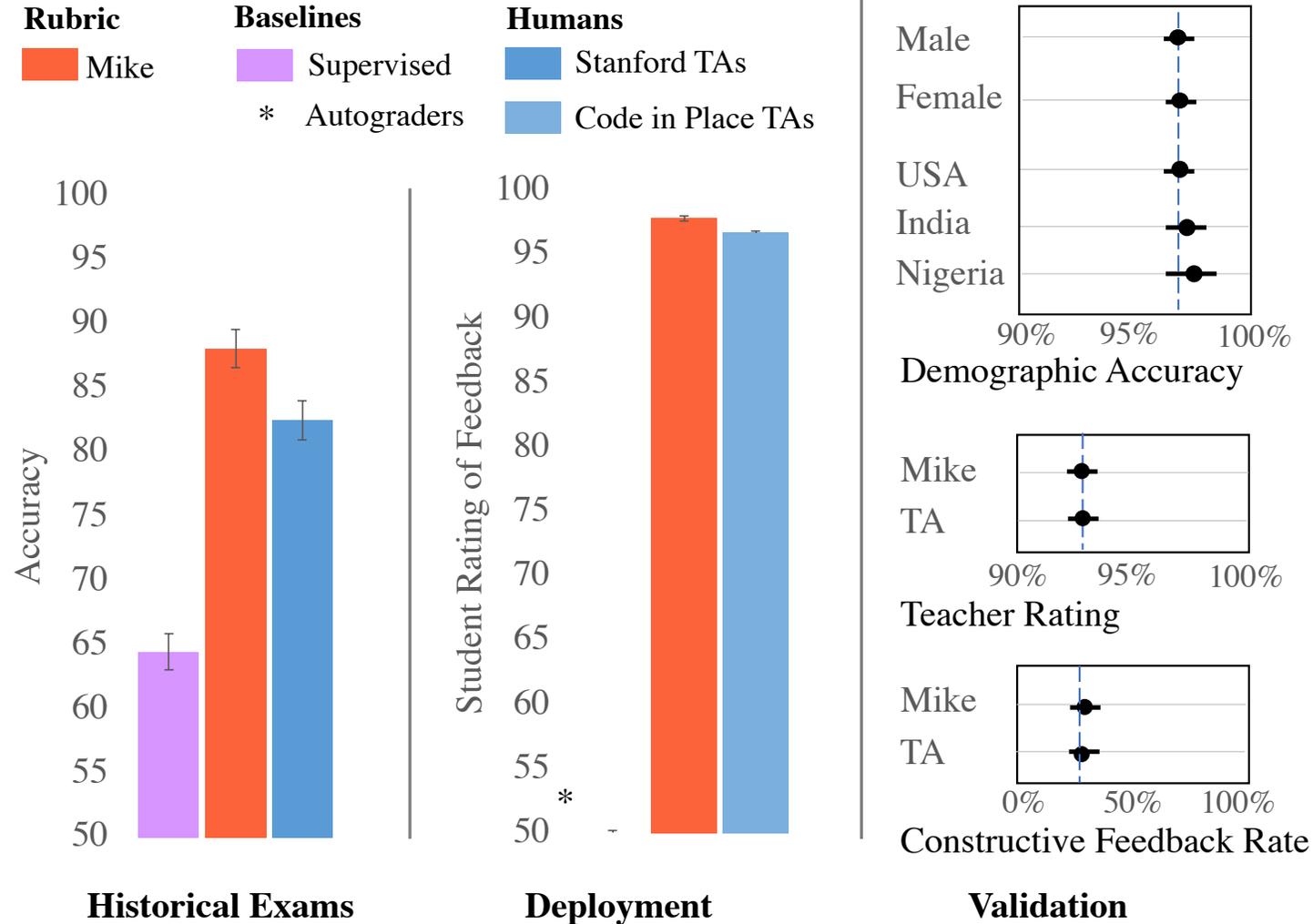
AI generated feedback

Students evaluate the feedback

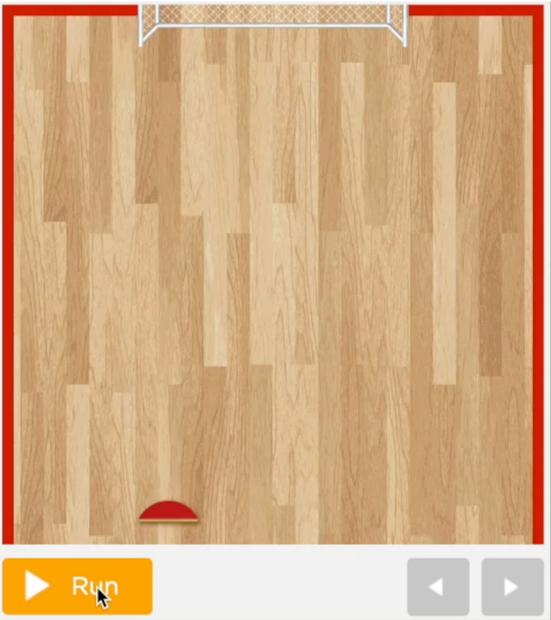
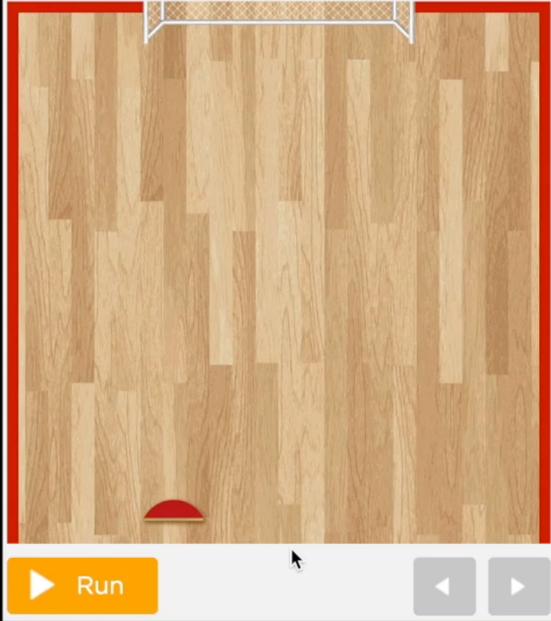
Algorithm uses attention to highlight where in the code the error comes from

Syntax error (missing ") here would prevent auto graders from being useful.

Pretty Figure

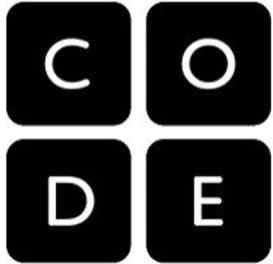


But what about interactive, creative assignments?

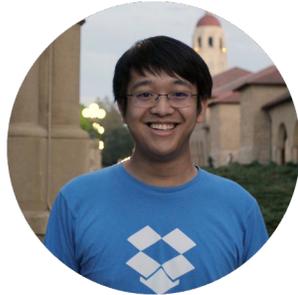


1M ungraded code.org assignments.

The AI is shown a brand new student game. Does it work?



Simultaneously learn to grade and play to grade.



Majority class: 50%
Code-as-text: 67%
Play-to-grade: **94%**



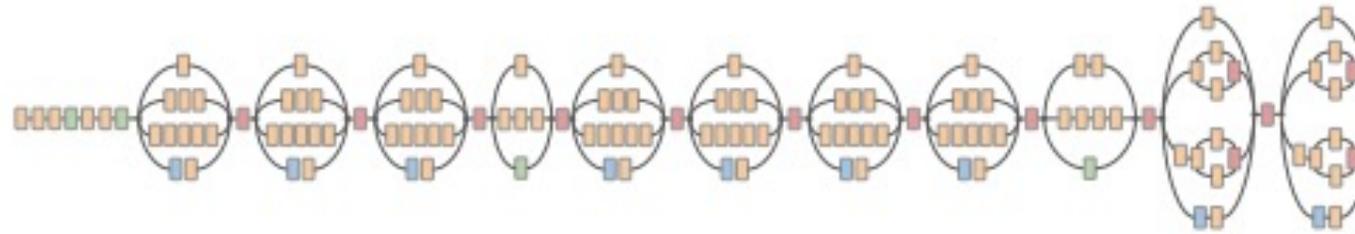


Detecting skin cancer

Skin Lesion Image



Deep Convolutional Neural Network (Inception-v3)



Training Classes (757)

- Acral-lent. melanoma
- Amelanotic melanoma
- Lentigo melanoma
- ...
- Blue nevus
- Halo nevus
- Mongolian spot
- ...
-
-
-

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