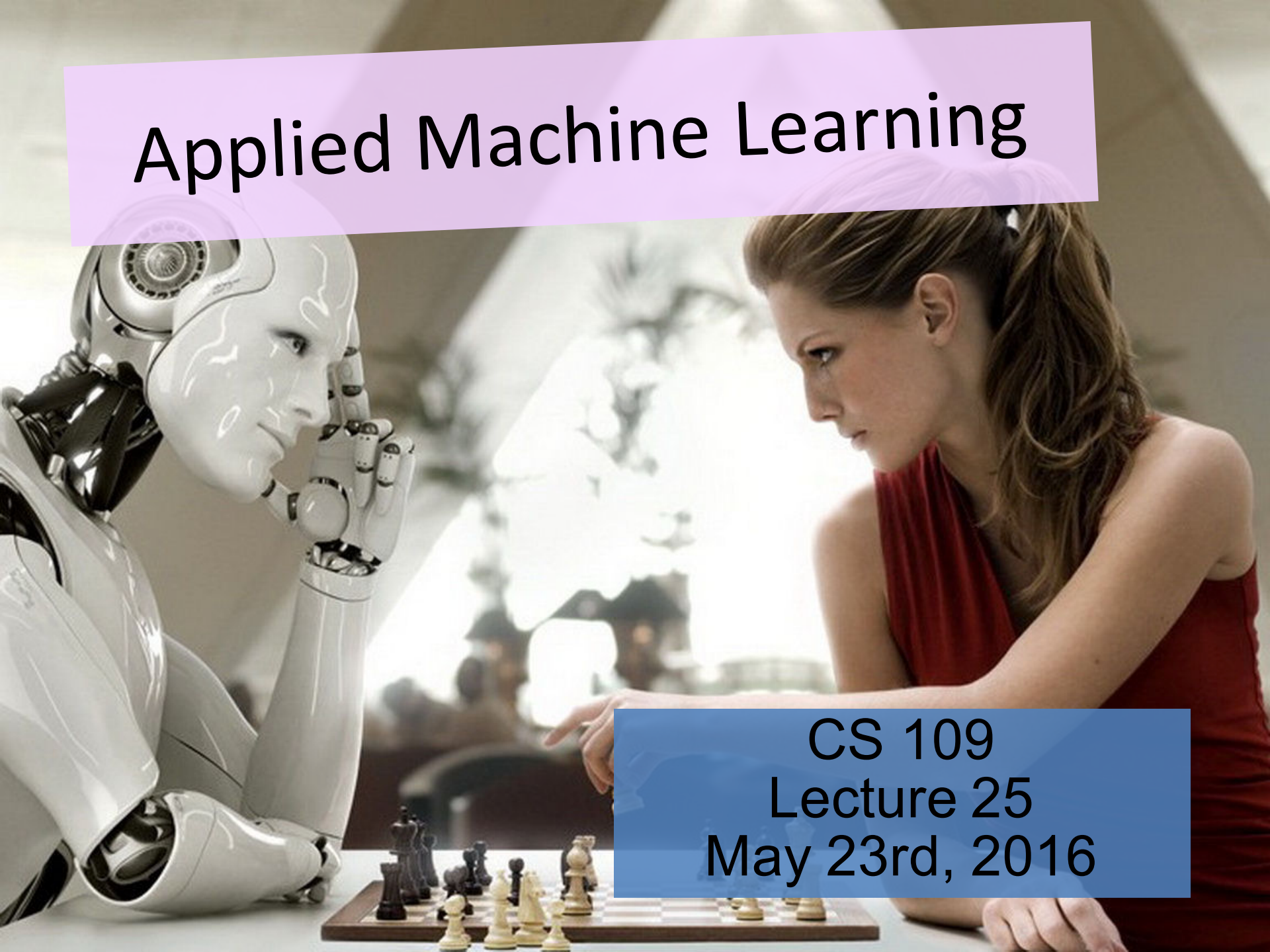
Applied Machine Learning



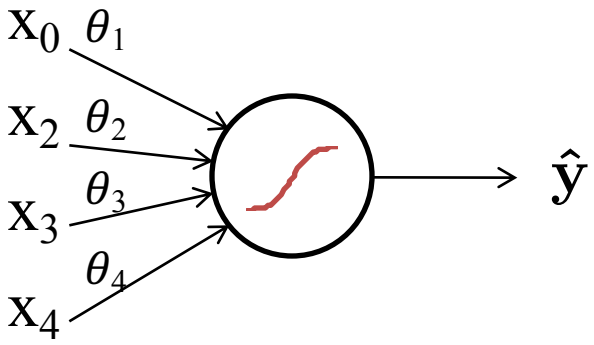
CS 109
Lecture 25
May 23rd, 2016

Yesterday: foundation
Today: big picture

Review

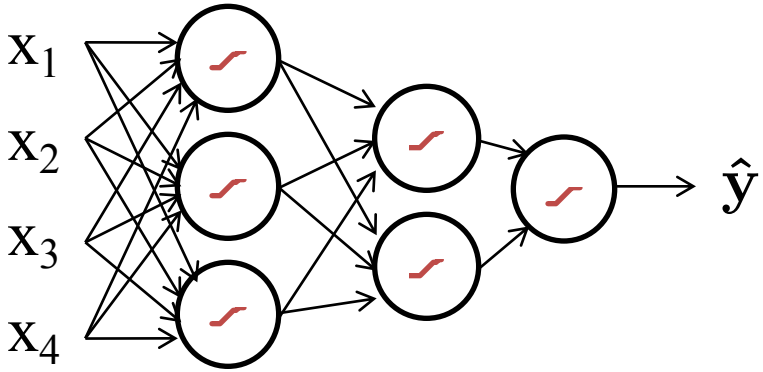
Logistic Regression and Neural Networks

- Consider logistic regression as:



Logistic regression is same as a one node neural network

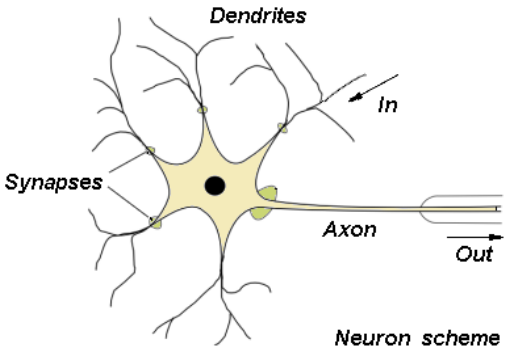
- Neural network



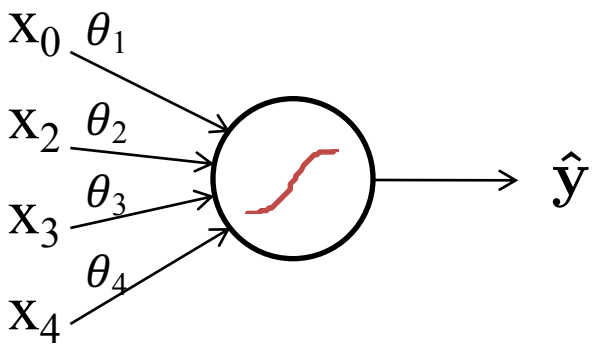
Piech

Biological Basis for Neural Networks

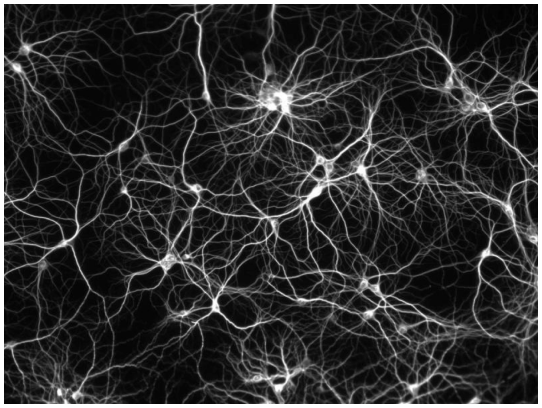
A neuron



Logistic Regression

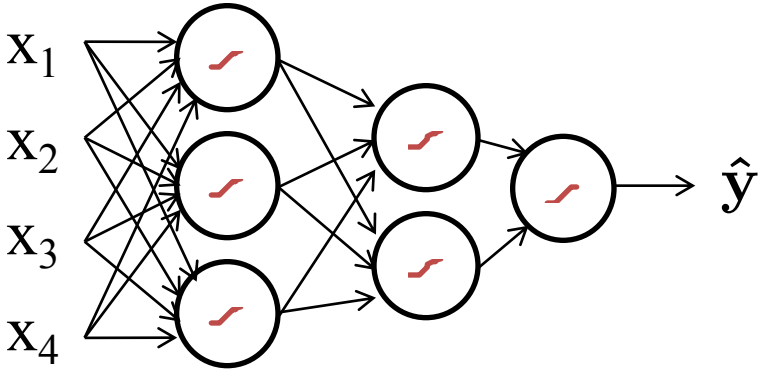


Your brain



Actually, it's probably someone else's brain

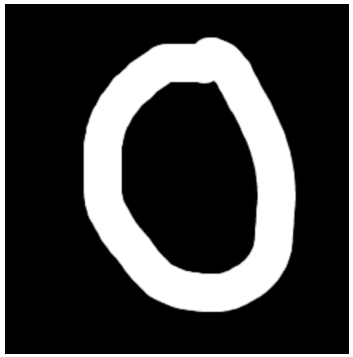
Neural Network



Piech

Digit Recognition Example

Let's make feature vectors from pictures of numbers

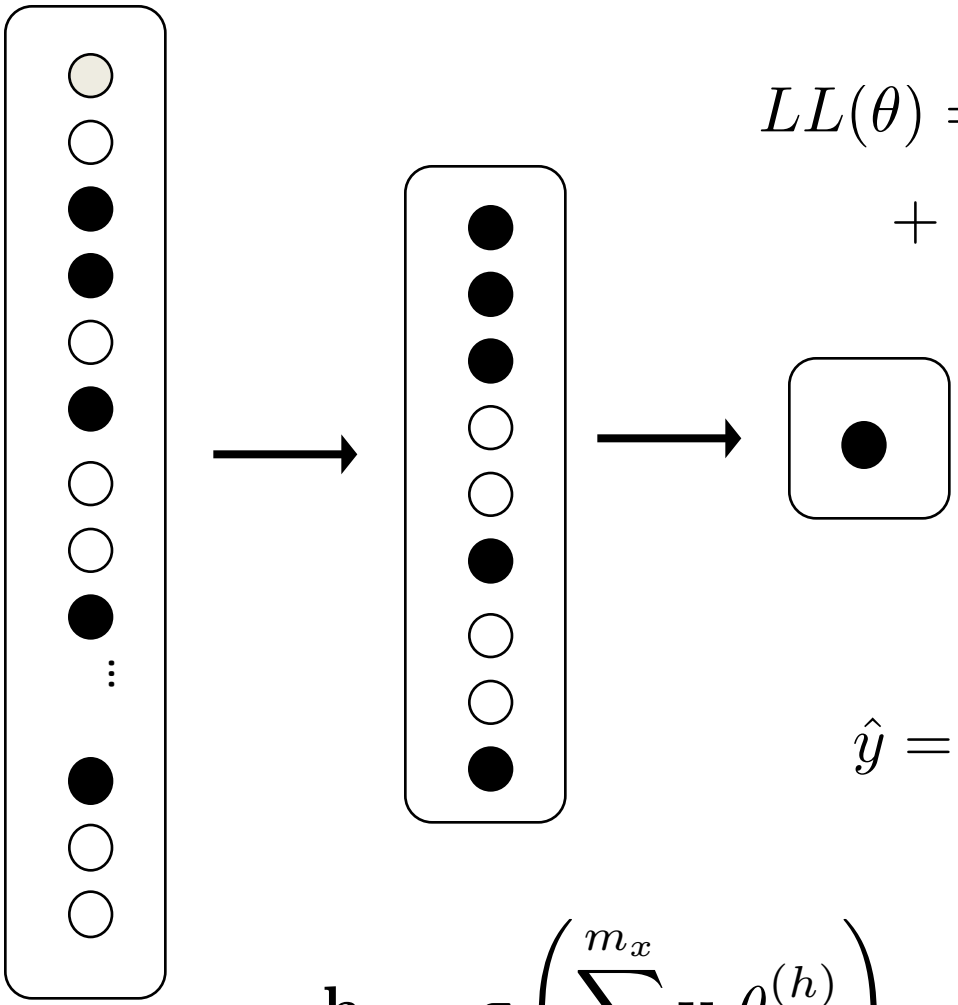


$$\mathbf{x}^{(i)} = [0, 0, 0, 0, \ldots, 1, 0, 0, 1, \ldots 0, 0, 1, 0]$$
$$y^{(i)} = 0$$

$$\mathbf{x}^{(i)} = [0, 0, 1, 1, \ldots, 0, 1, 1, 0, \ldots 0, 1, 0, 0]$$
$$y^{(i)} = 1$$

Piech

Forward Pass



$$LL(\theta) = y \log \hat{y} + (1 - y) \log[1 - \hat{y}]$$

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

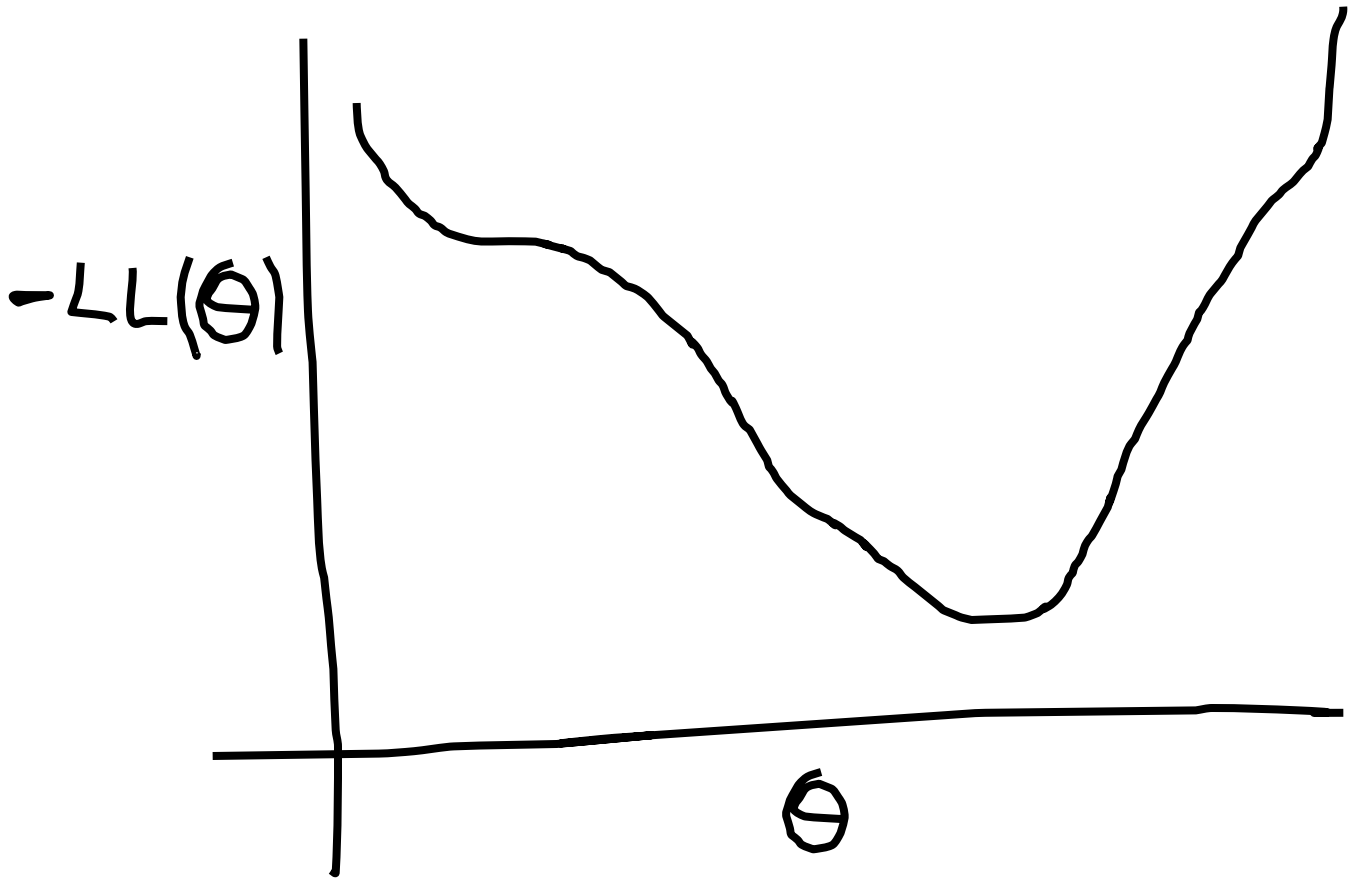
$$\mathbf{h}_j = \sigma\left(\sum_{i=0}^{m_x} \mathbf{x}_i \theta_{i,j}^{(h)}\right)$$

Piech

Gotta Get Those Good Thetas!

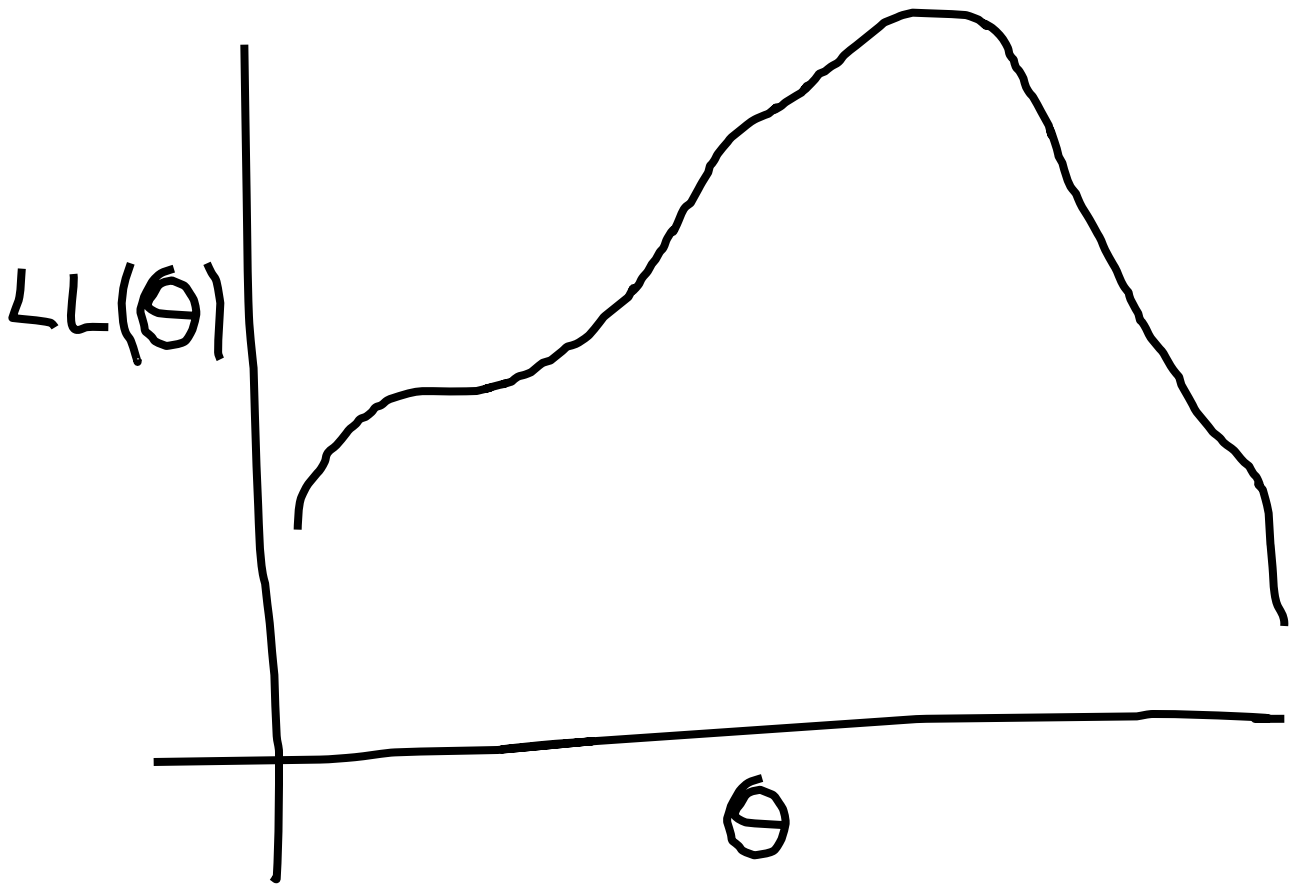
Gradient Ascent

Or is it descent?



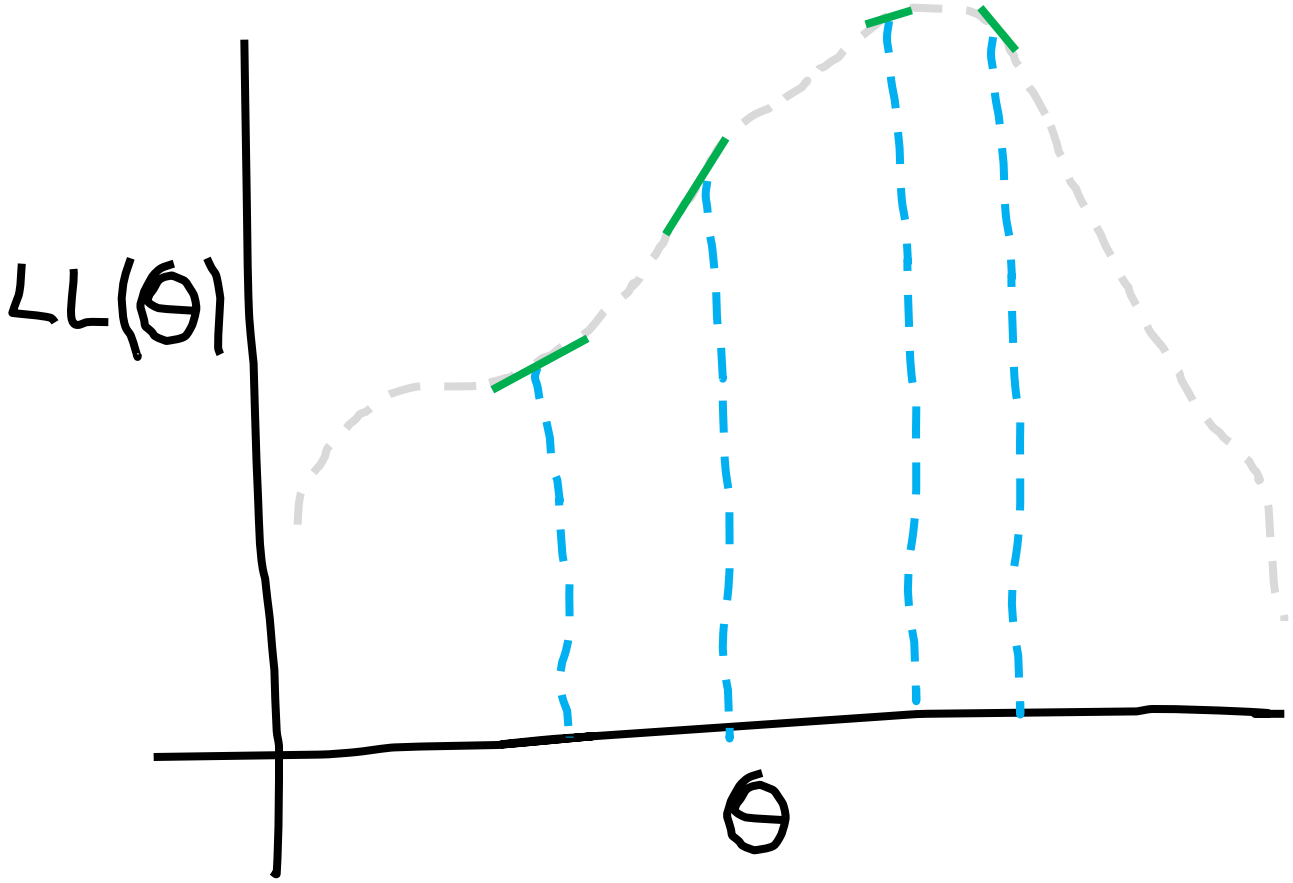
Piech

Gradient Ascent



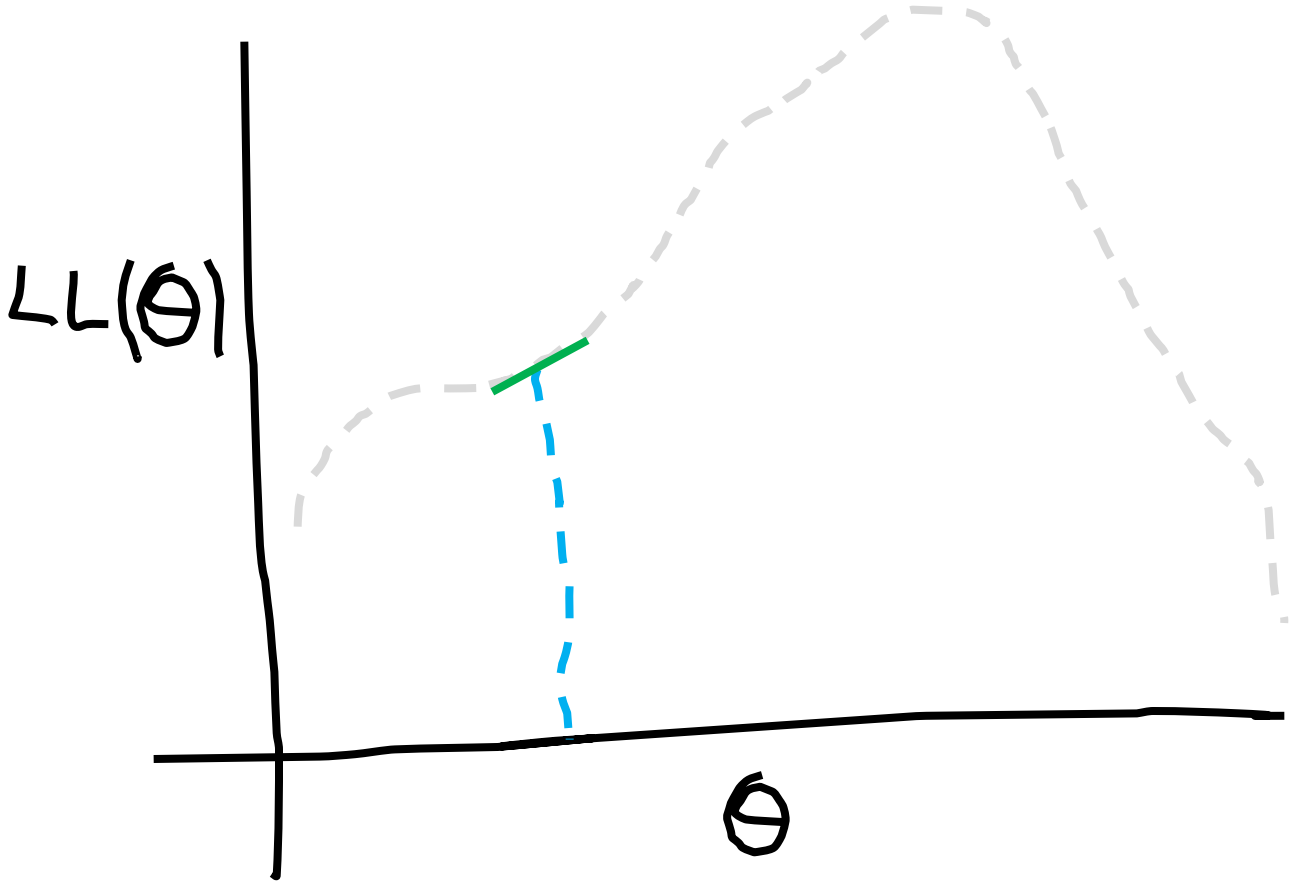
Piech

Gradient Ascent



Piech

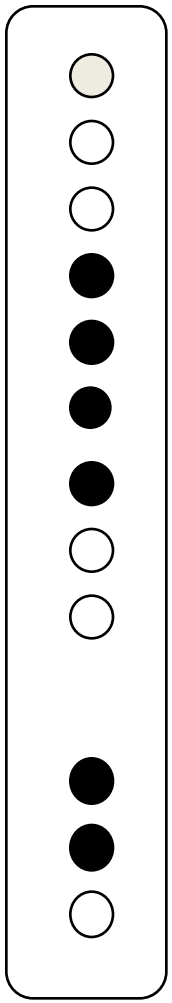
Gradient Ascent Yuge/Tiny Step Size



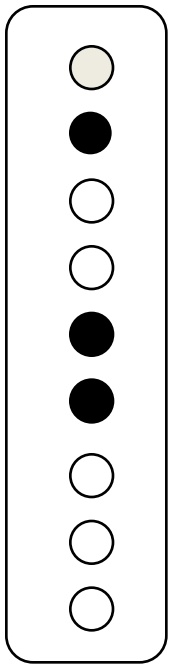
Piech

New Notation

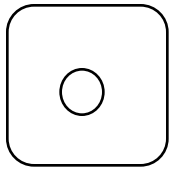
Layer **x**



Layer **h**

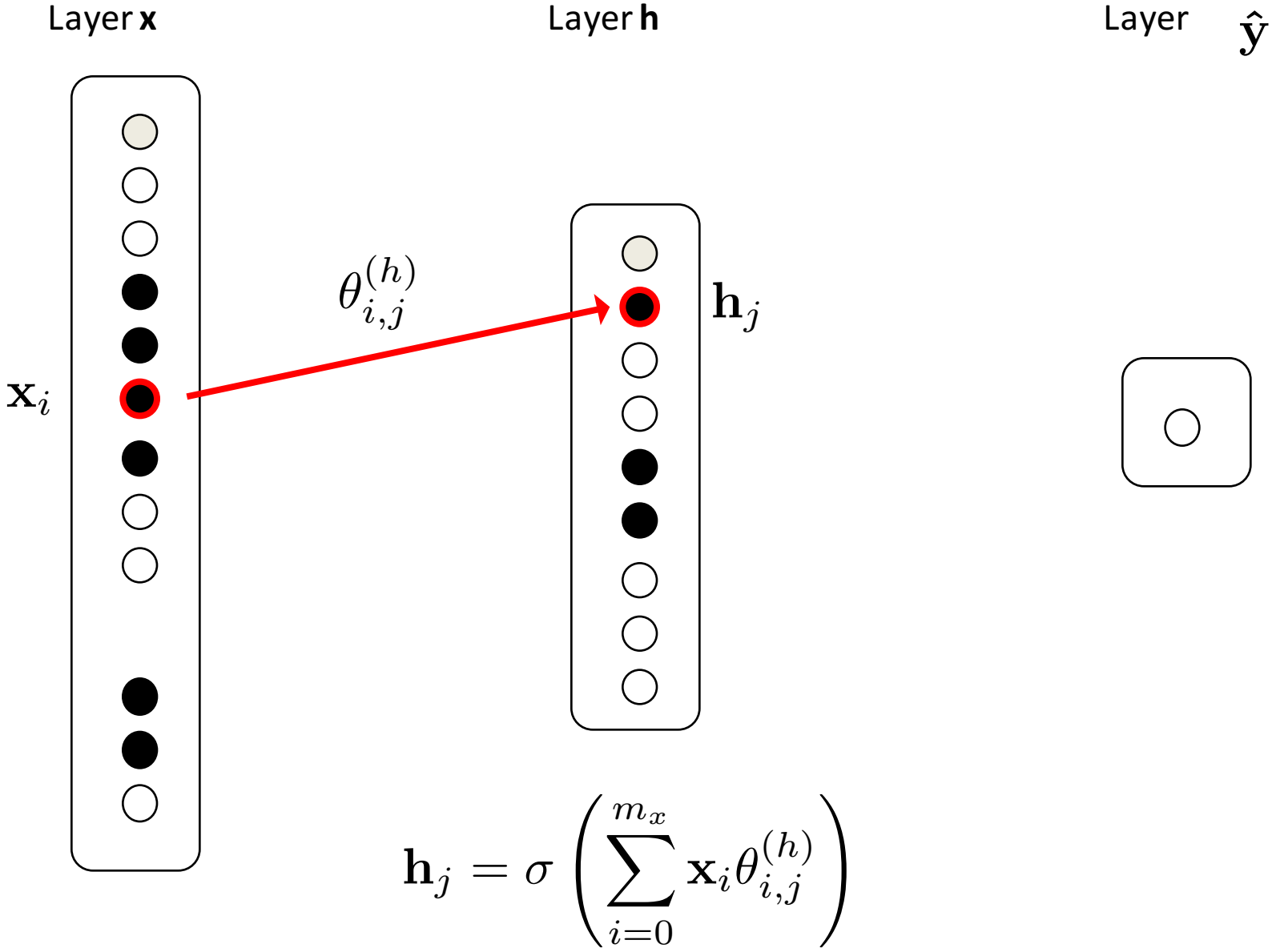


Layer $\hat{\mathbf{y}}$



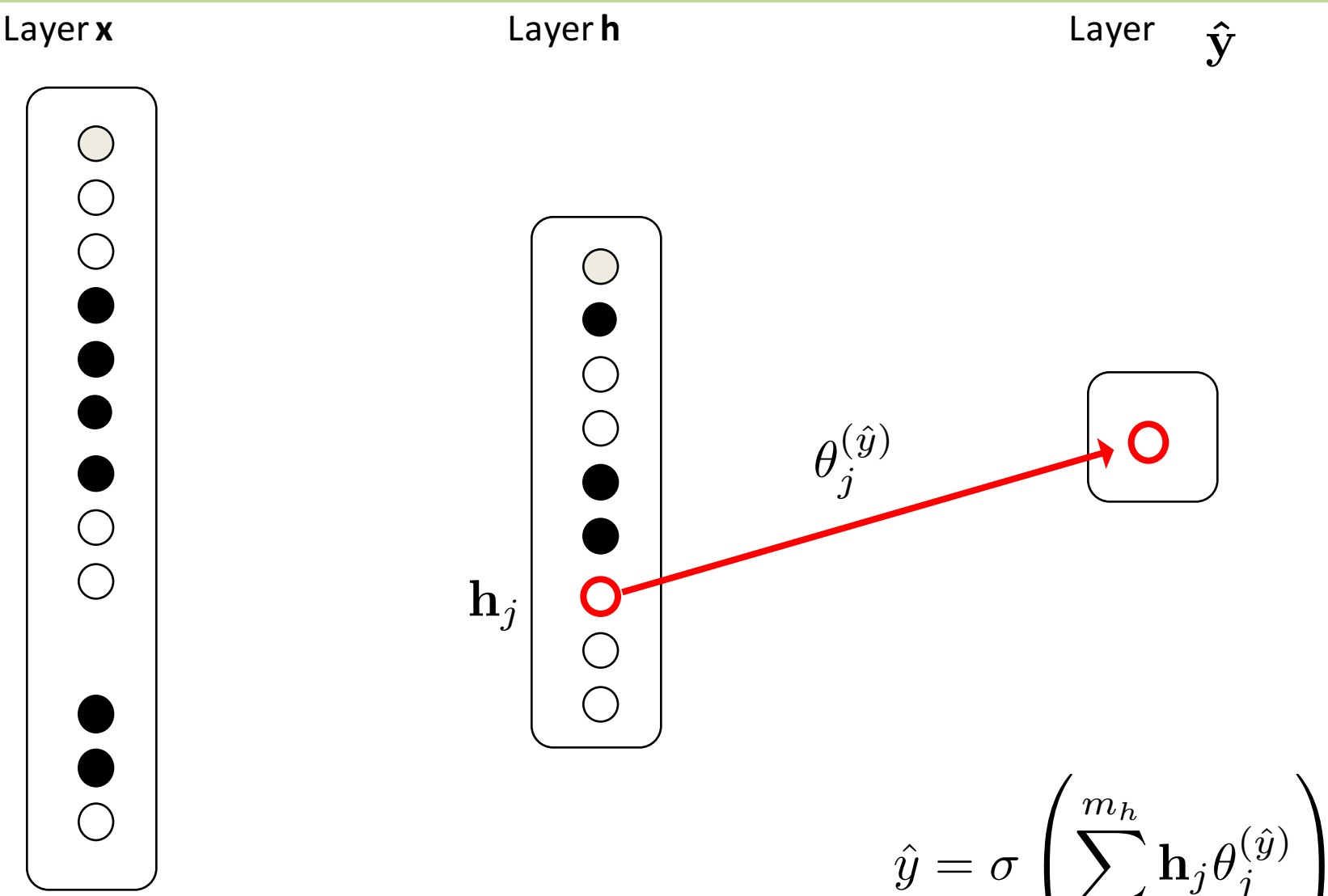
Piech

New Notation



$$\mathbf{h}_j = \sigma\left(\sum_{i=0}^{m_x} \mathbf{x}_i \theta_{i,j}^{(h)}\right)$$

New Notation

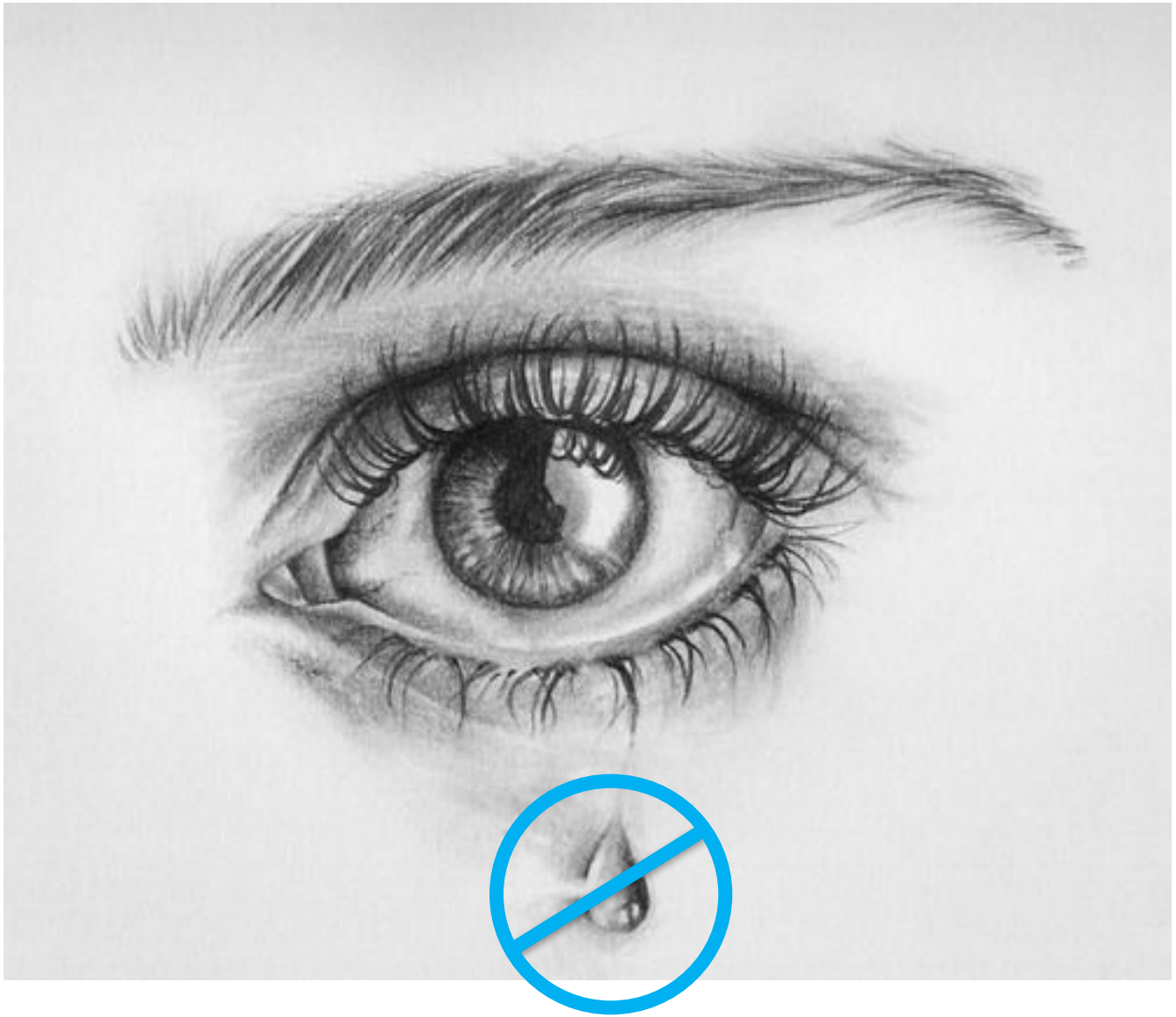


$$\hat{y} = \sigma\left(\sum_{j=0}^{m_h} \mathbf{h}_j \theta_j^{(\hat{y})}\right)$$

Piech

Derivatives without tears

Derivatives Without Tears



Piech

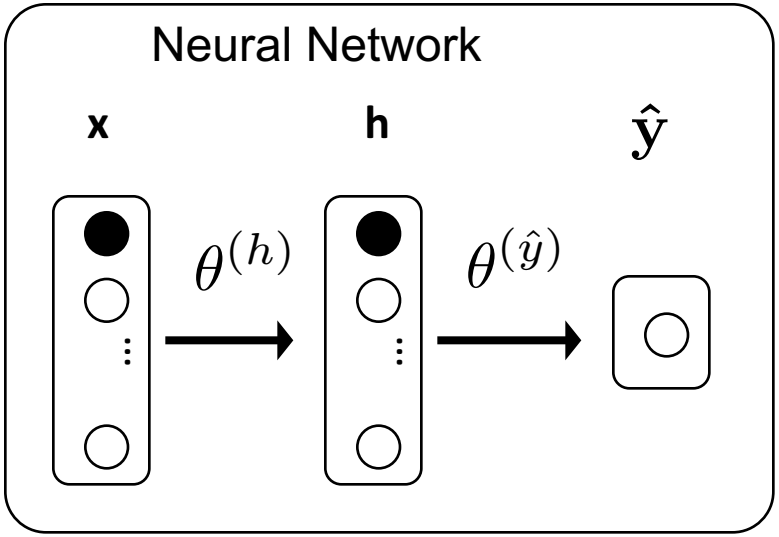
Derivative Goals

Loss with respect to output layer params

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

Loss with respect to hidden layer params

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

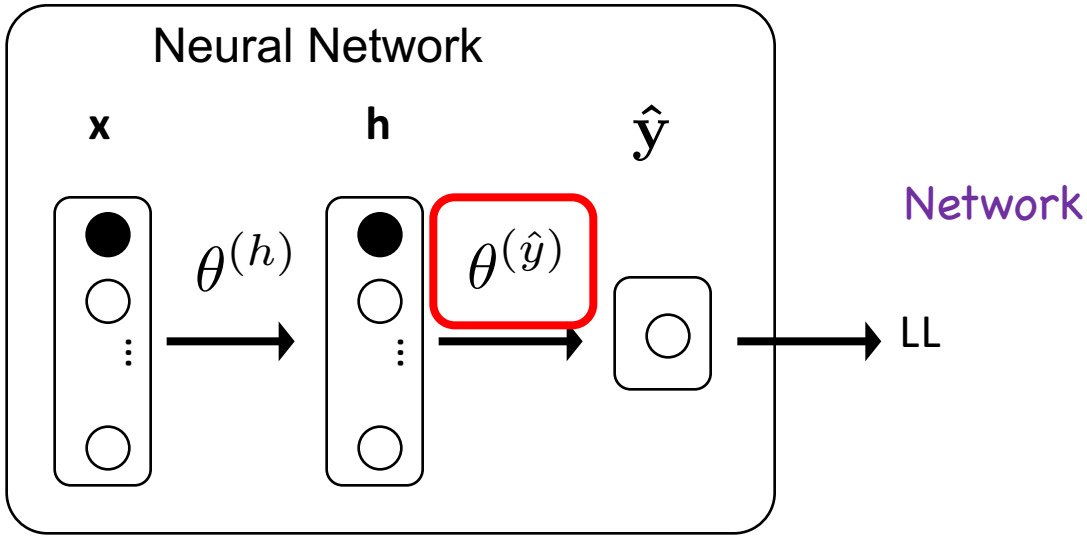


Piech

Output Parameter Gradient

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

Goal



Network

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

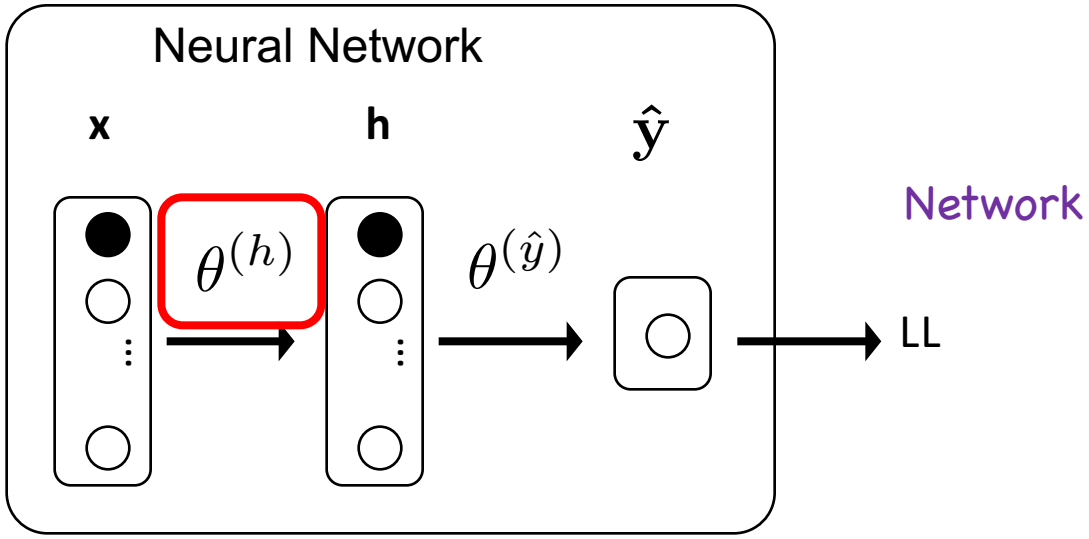
Decomposition

Piech

Hidden Layer Parameter Gradient

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

Goal



Network

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

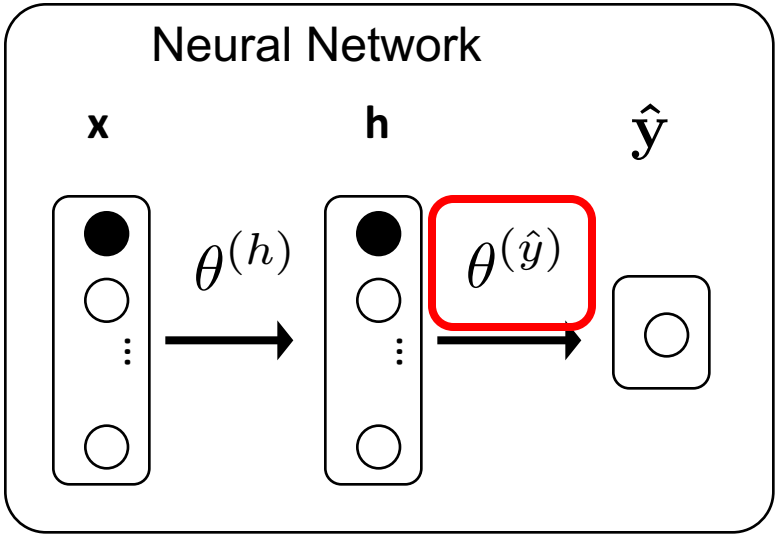
Decomposition

Piech

Break it down mr tamborine man

Example of Gradient

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



Piech

Gradient of output layer params

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

$$LL(\theta) = y \log \hat{y} + (1 - y) \log[1 - \hat{y}]$$

$$\frac{\partial LL(\theta)}{\partial \hat{y}} = \frac{y}{\hat{y}} + \frac{(1 - y)}{(1 - \hat{y})} \cdot \frac{\partial (1 - \hat{y})}{\partial \hat{y}}$$

$$= \frac{y}{\hat{y}} - \frac{(1 - y)}{(1 - \hat{y})}$$

Piech

Recall: Sigmoid has a Beautiful Slope

True fact about sigmoid functions

$$\frac{\partial}{\partial z}\sigma(z) = \sigma(z)[1 - z]$$

Sigmoid, you should be a ski hill

Piech

Gradient of output layer params

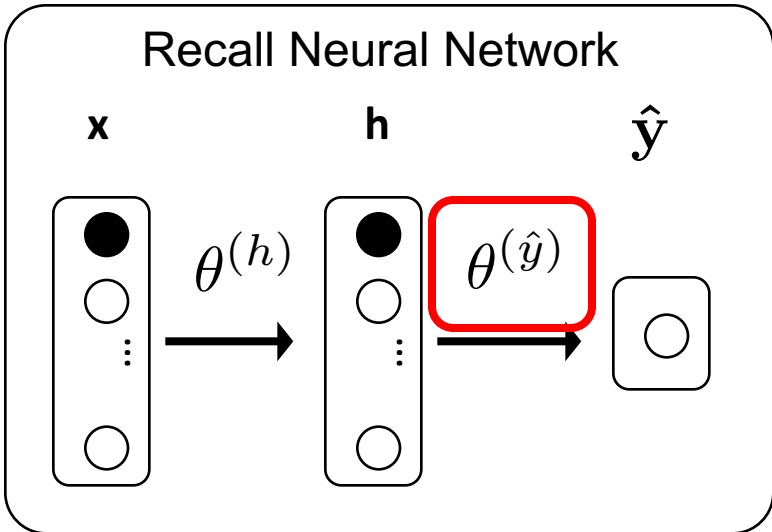
$$\frac{\partial LL(\theta)}{\partial \theta_i^{(\hat{y})}} = \frac{\partial LL}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial \theta_i^{(\hat{y})}}$$

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

$$\frac{\partial \hat{y}}{\partial \theta_i^{(\hat{y})}} = \hat{y}[1-\hat{y}] \cdot \frac{\partial}{\partial \theta_i^{(\hat{y})}} \sum_{j=0}^{m_h} \mathbf{h}_j \theta_j^{(\hat{y})}$$

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

What! That's not scary!



Make it Simple

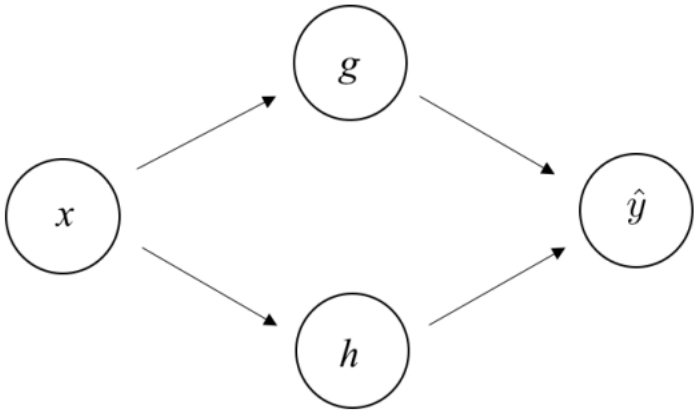
$$\frac{\partial LL(\theta)}{\partial \theta_i^{(\hat{y})}} = \text{  } \cdot \text{  }$$

$$\text{  } = \frac{y}{\hat{y}} - \frac{(1-y)}{(1-\hat{y})}$$

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

Boom!

What If You Had a Neural Network Like This?



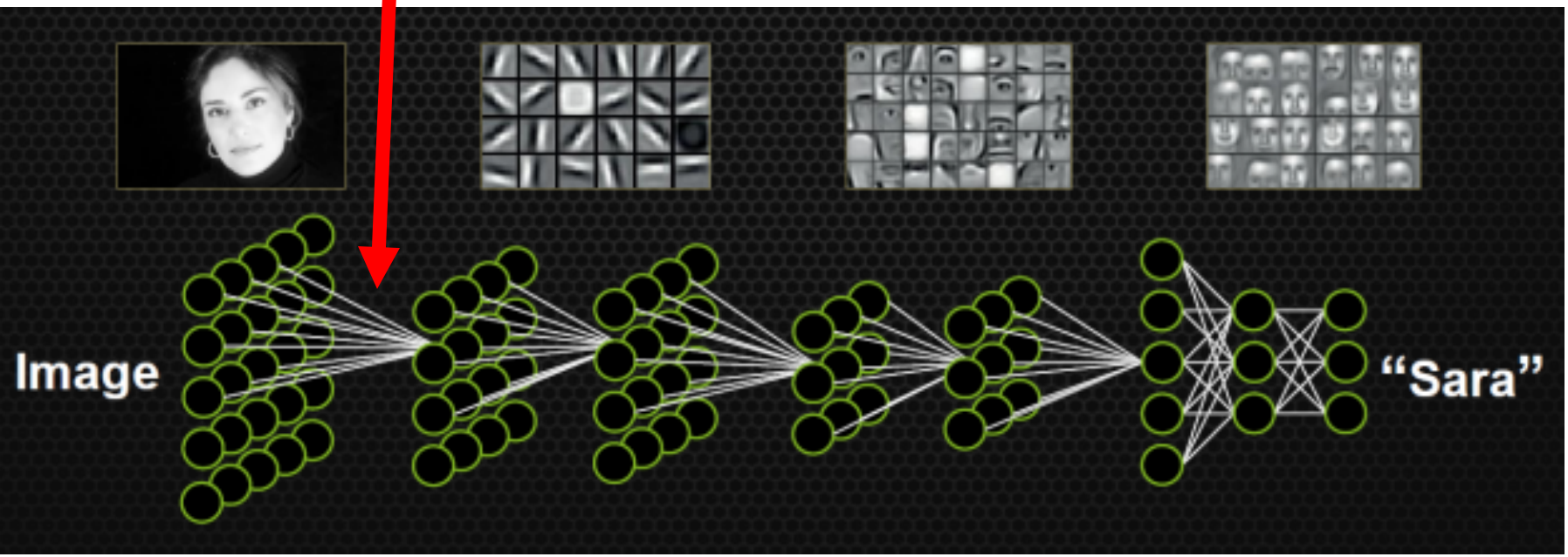
$$g = \text{sigmoid}(\theta_1 \cdot x)$$
$$h = \text{sigmoid}(\theta_2 \cdot x)$$
$$\hat{y} = \text{sigmoid}(\theta_3 \cdot g + \theta_4 \cdot h)$$
$$LL(\theta) = \sum_{i=1}^{n} y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})$$

1. Calculate partial derivative for one data instance
2. Use chain rule!
3. Sigmoid derivatives come out simple if you use the right notation
4. You don't need to give the most reduced answer

Piech

Works for any number of layers

$\theta^{(\mathbf{a})}$



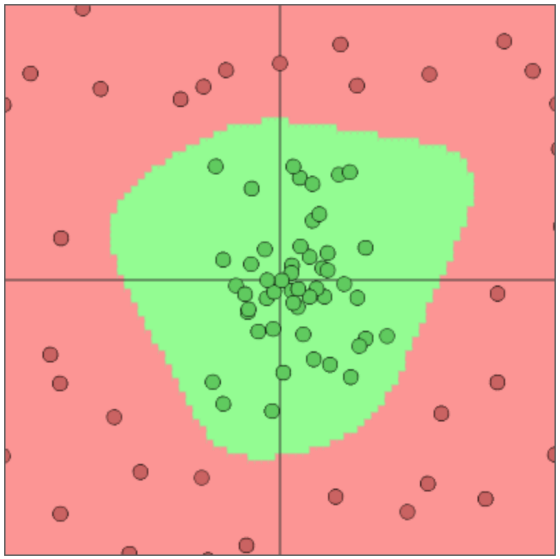
x a b c d e f g $\hat{y}$ LL

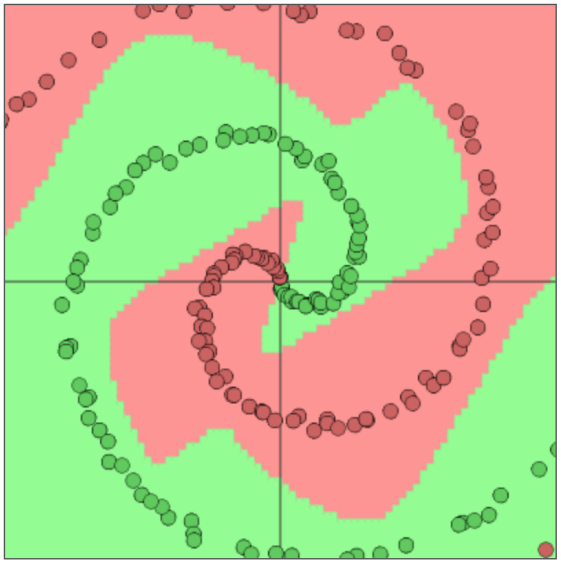
Piech

model.update(data)

Neural Networks are Turing Complete

- Some datasets are not linearly separable

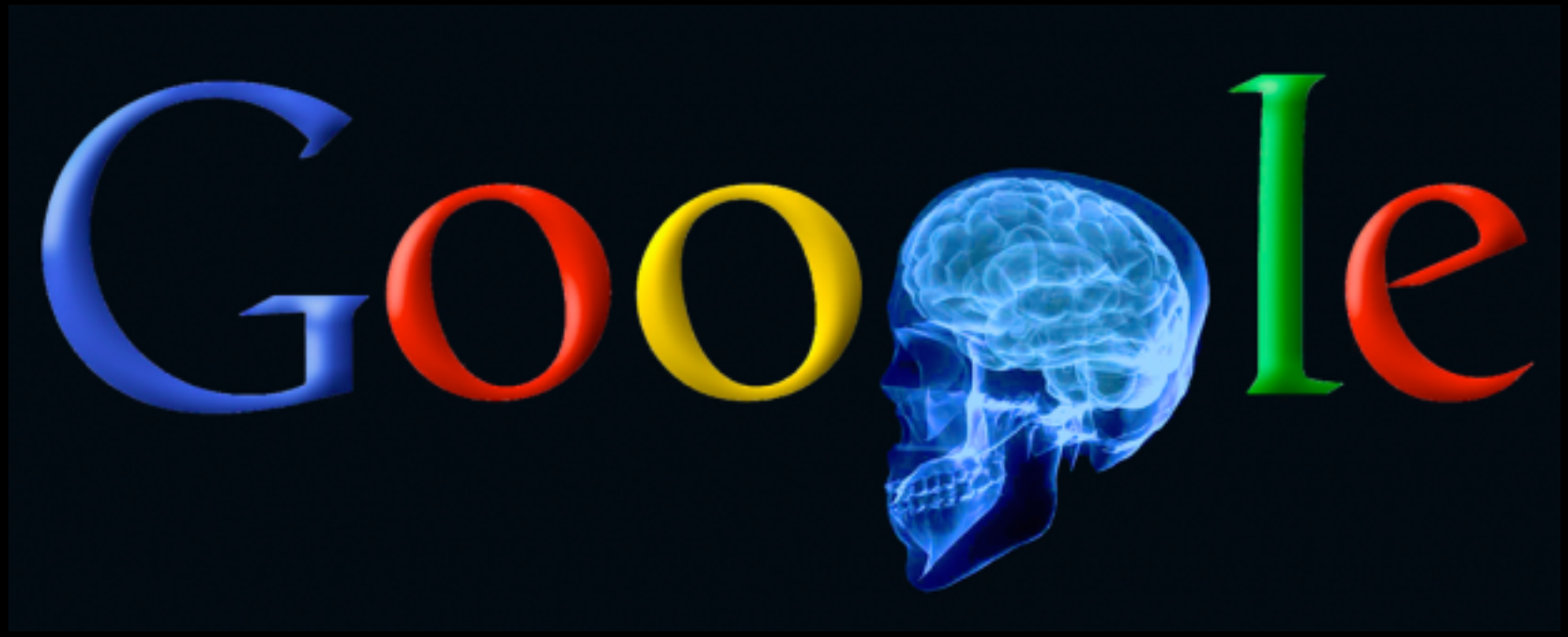




  - These are classifiers learned by neural networks

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

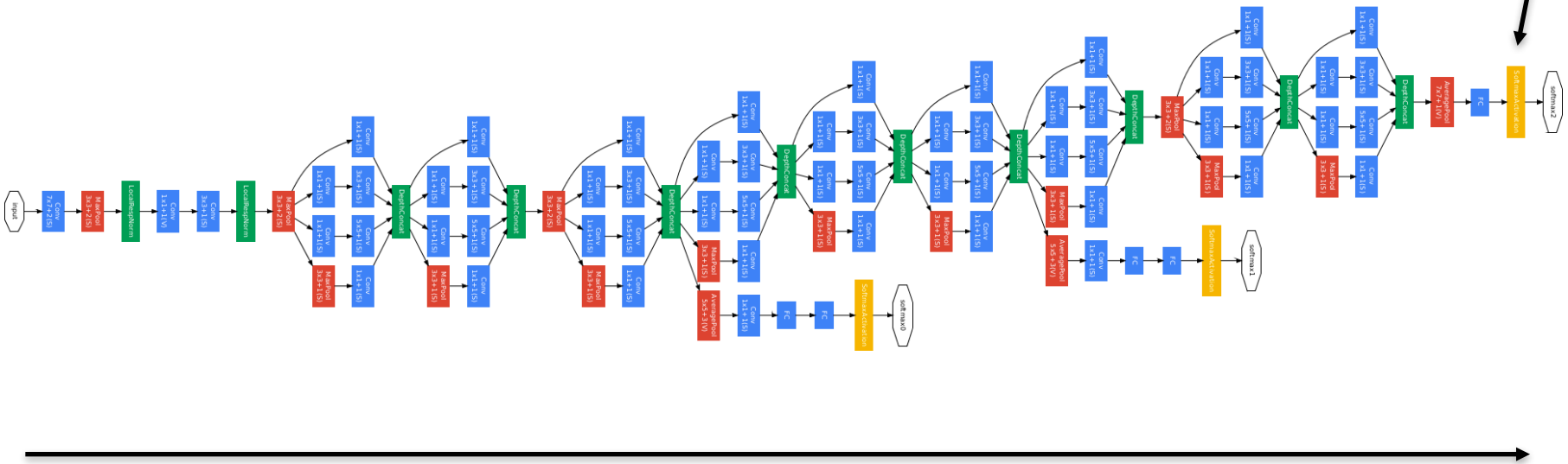
GoogLeNet Brain



1 Trillion Artificial Neurons

GoogLeNet Brain

Multiple,
Multi class output

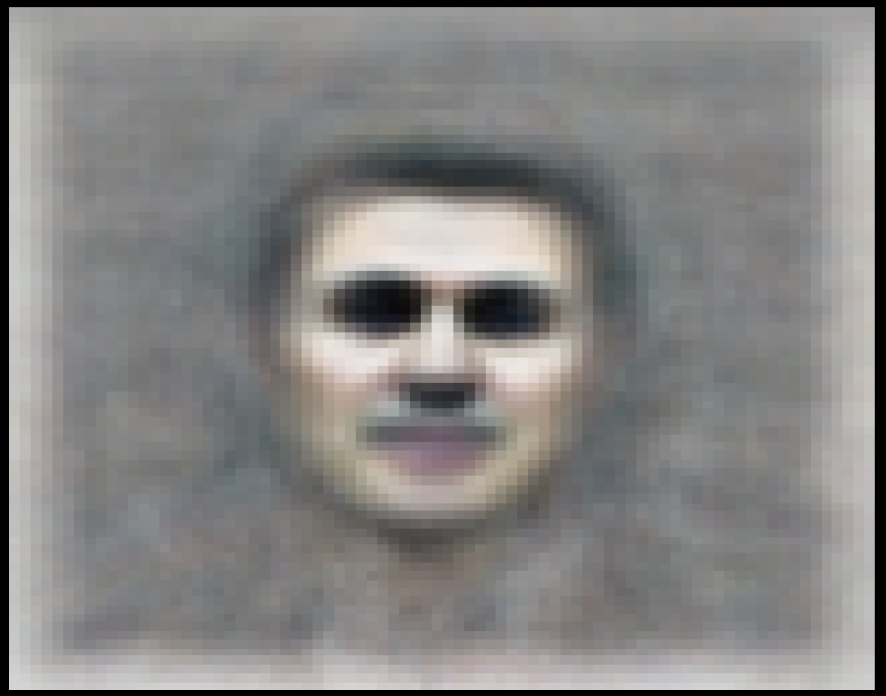


22 layers deep

Piech

The Face Neuron

Top stimuli from the test set



Optimal stimulus
by numerical optimization

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

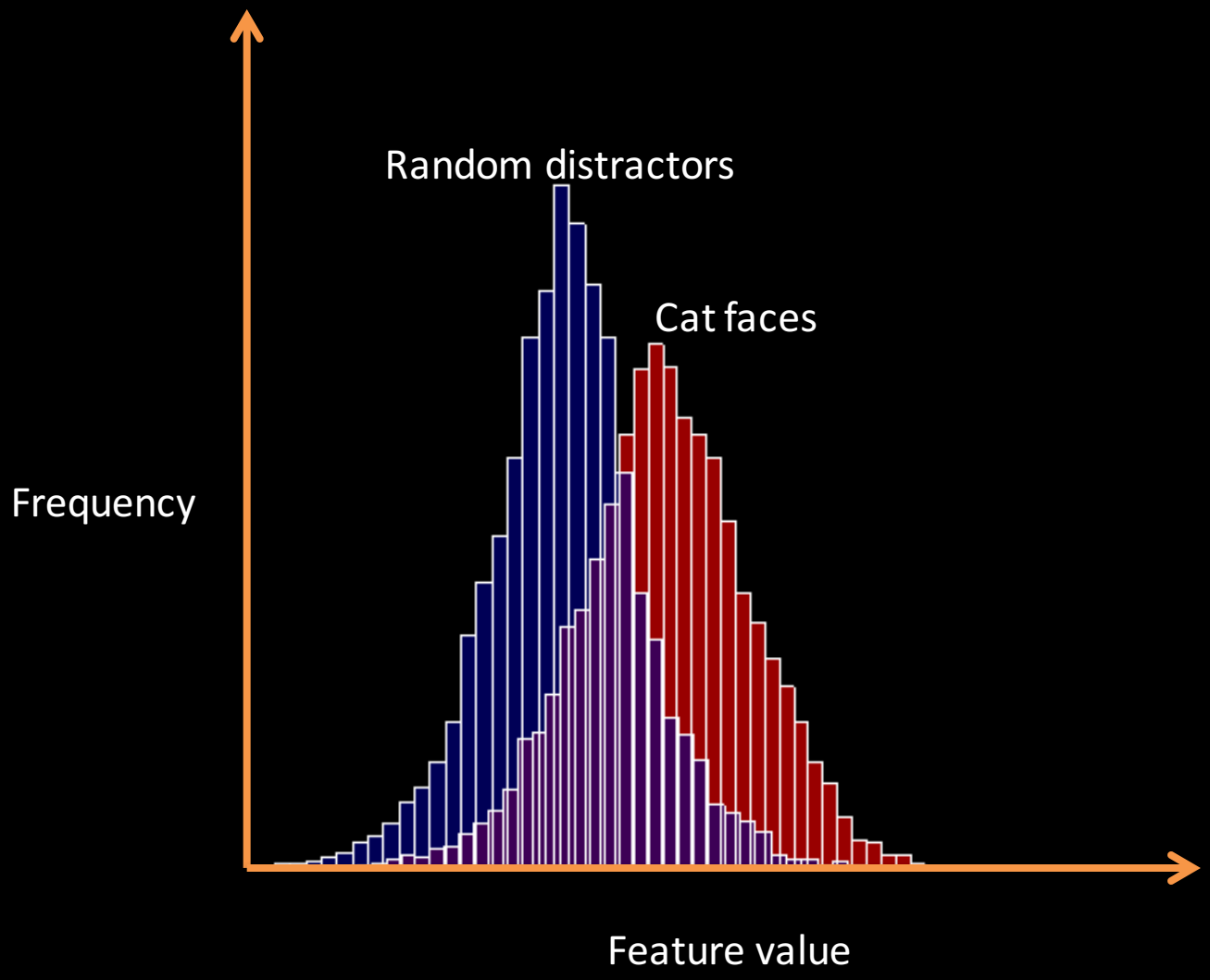
The Cat Neuron

Top stimuli from the test set

Optimal stimulus
by numerical optimization

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

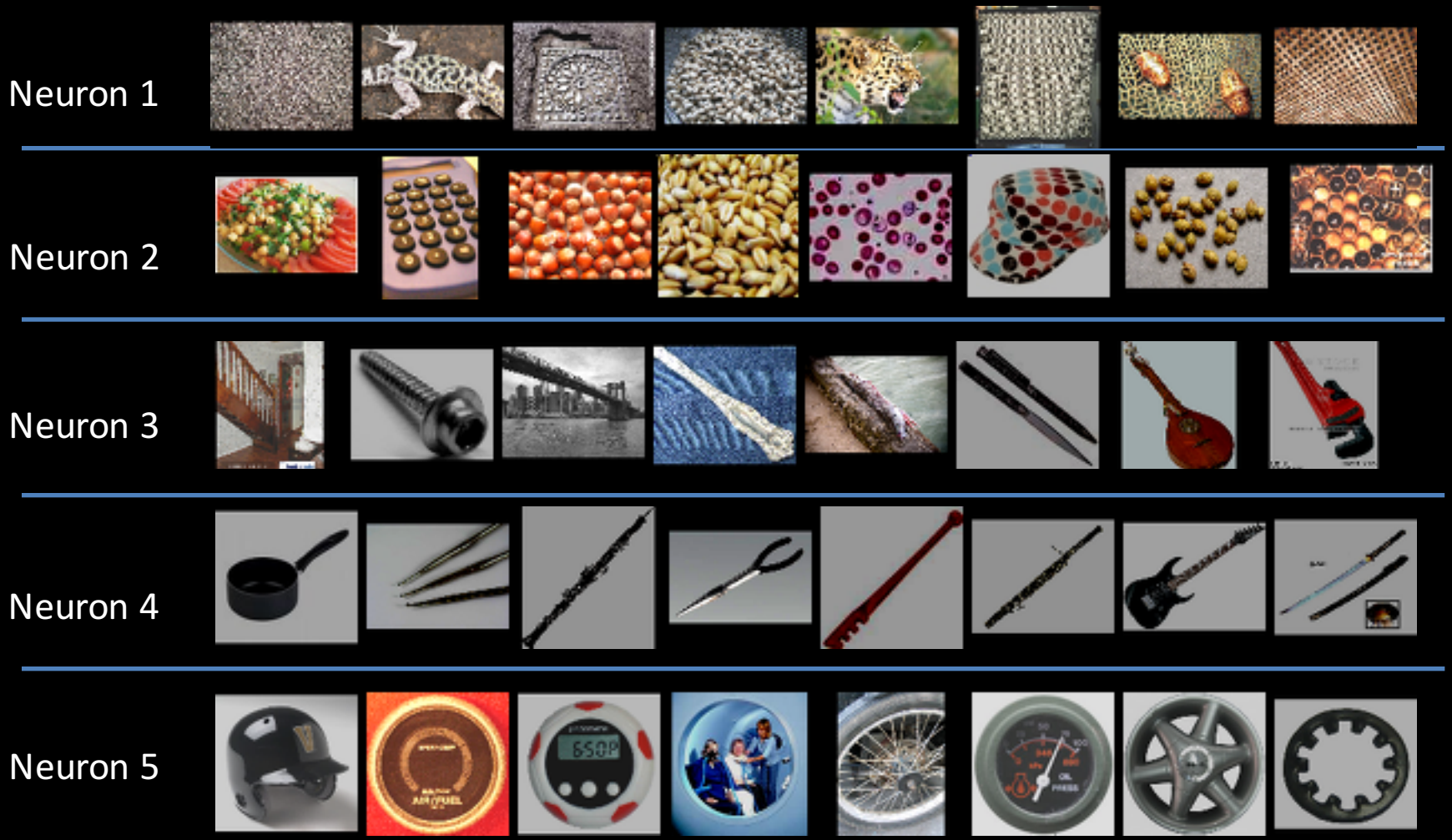
The Cat Neuron



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


Hire the smartest people in the world
Invent cat detector

Best Neuron Stimuli



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

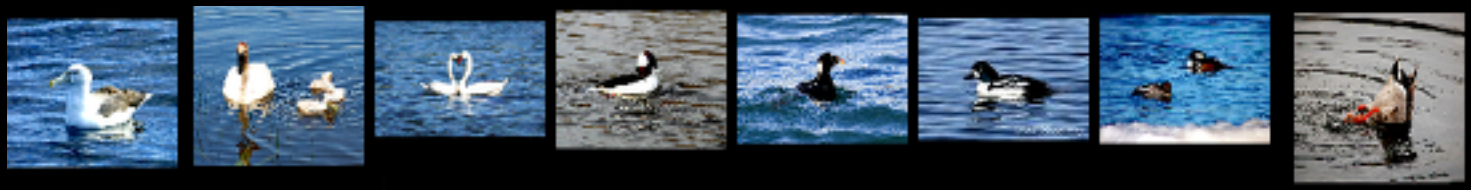
Best Neuron Stimuli

Neuron 6

Neuron 7

Neuron 8

Neuron 9



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

Best Neuron Stimuli

Neuron 10

Neuron 11

Neuron 12

Neuron 13

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

ImageNet Classification

22,000 categories

14,000,000 images

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

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

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
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



0.005%

Random guess

1.5%

Pre Neural Networks

?

GoogLeNet

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

0.005%

Random guess

1.5%

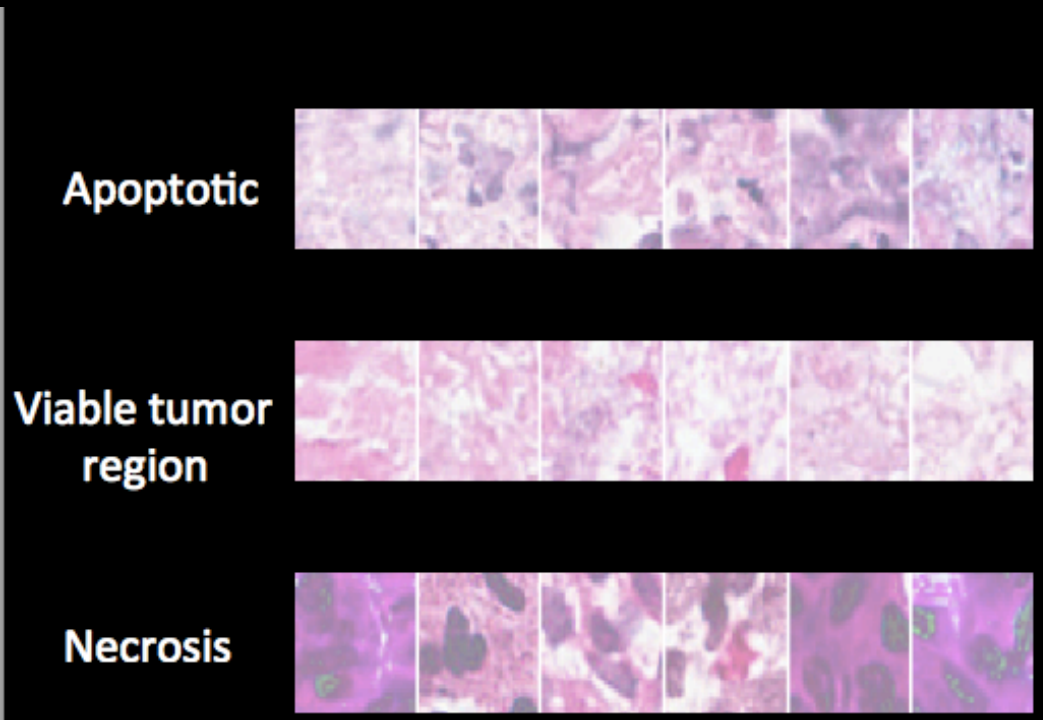
Pre Neural Networks

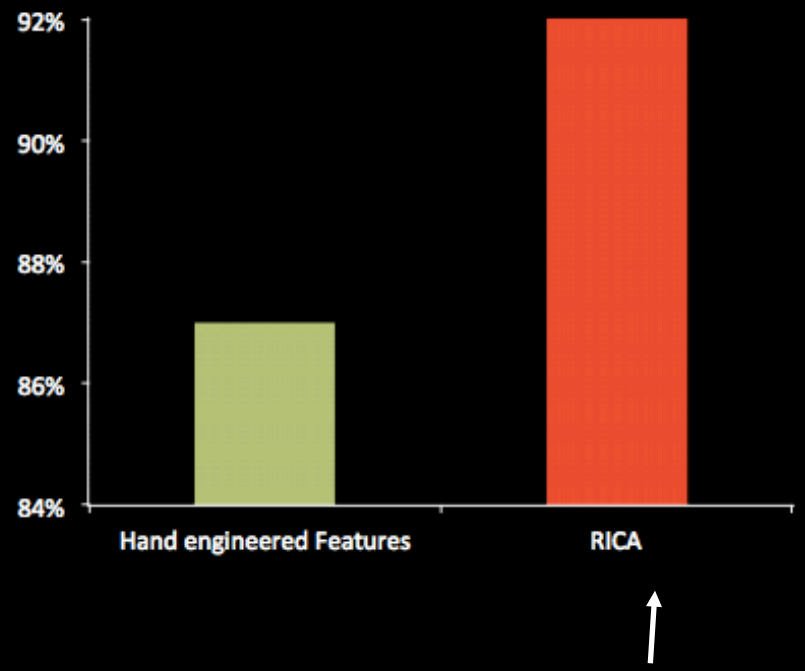
43.9%

GoogLeNet

Szegedy et al, Going Deeper With Convolutions, CVPR 2015

Vision has Social Implications





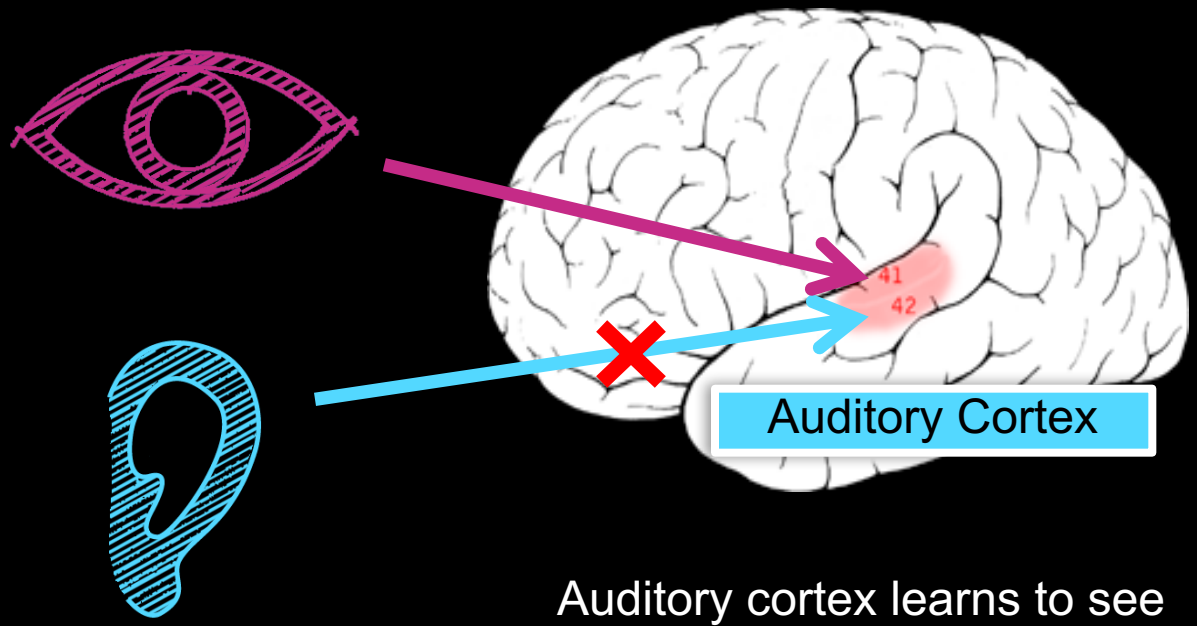
Neural network

One Algorithm Hypothesis

Much of perception in the brain can be explained with a single learning algorithm.

[Andrew Ng]

One Algorithm Hypothesis

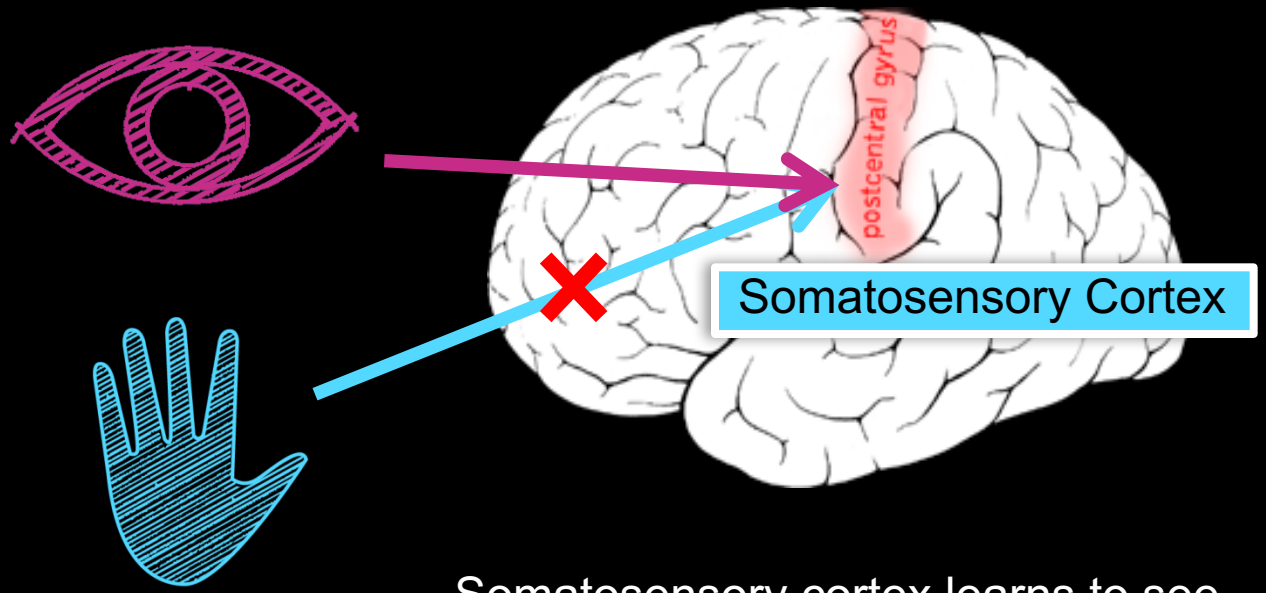


Auditory cortex learns to see

[Roe et al., 1992]

[Andrew Ng]

One Algorithm Hypothesis



Somatosensory cortex learns to see

[Metin & Frost, 1989]

[Andrew Ng]

Live Modelling

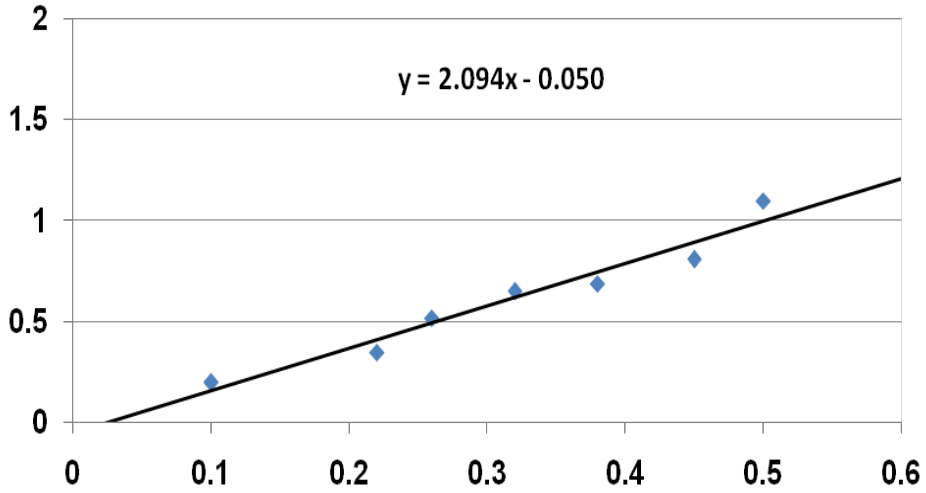
Goal: Predict your final grade

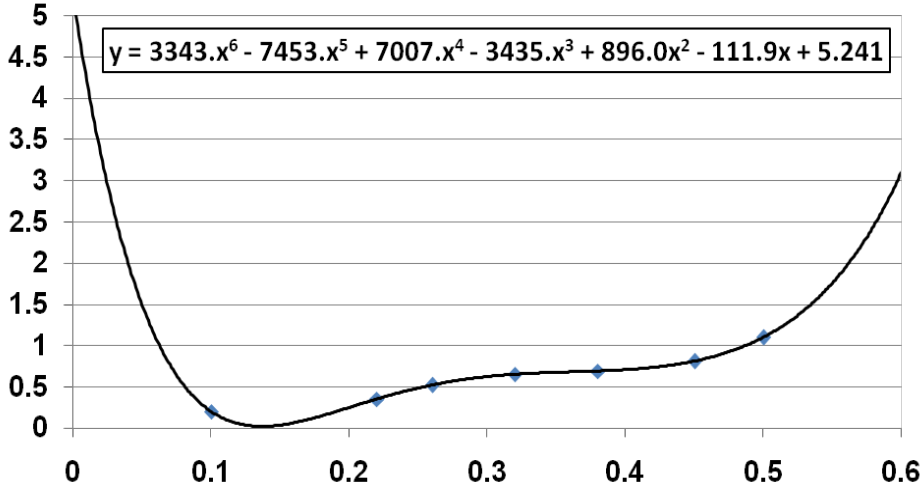
Input: $x = [x_1, x_2, x_3, \ldots x_{243}]$

Piech

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

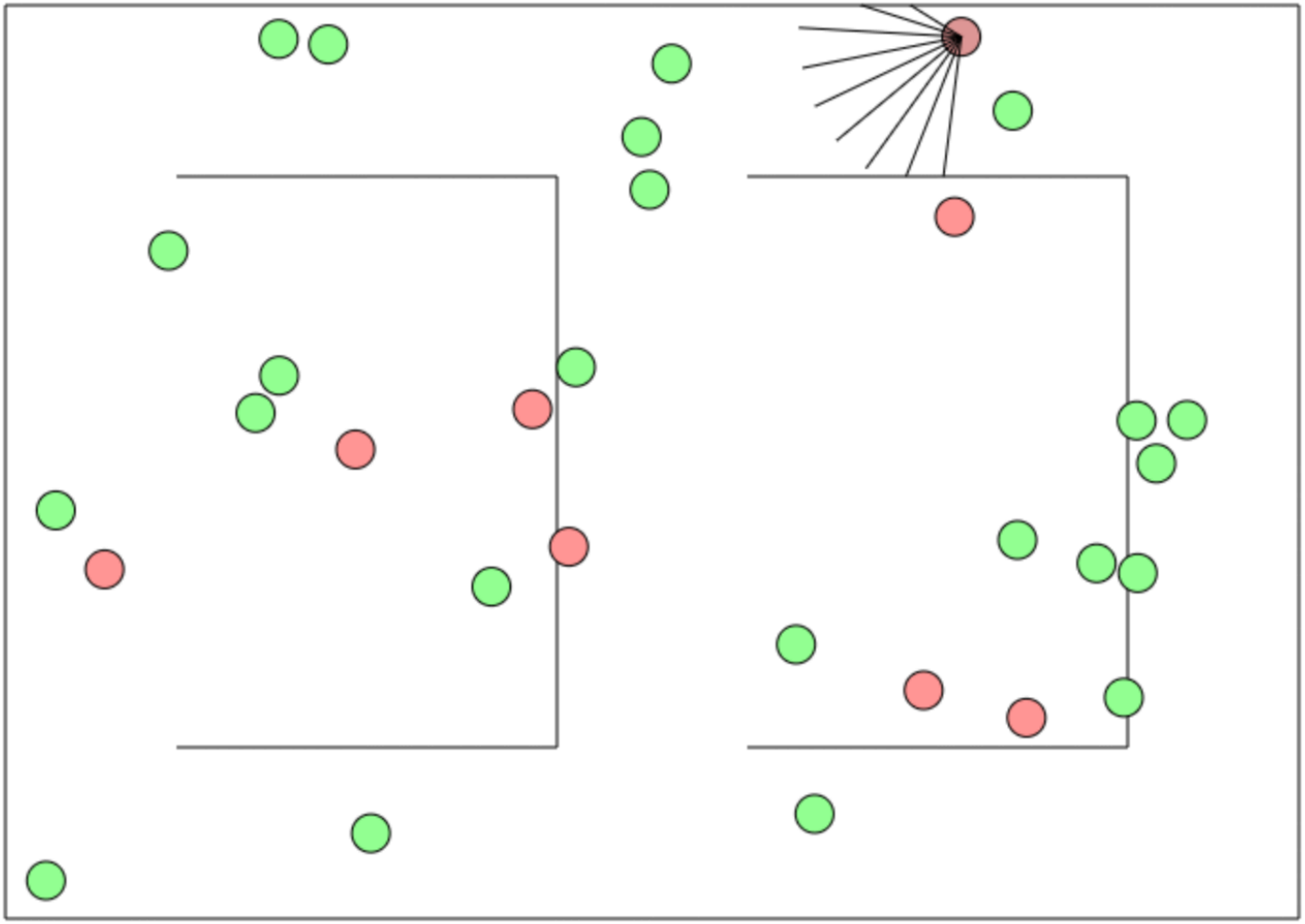




  - Polynomial on the right fits training data perfectly!
  - Which would you rather use to predict a new data point?

Piech

Deep Reinforcement Learning



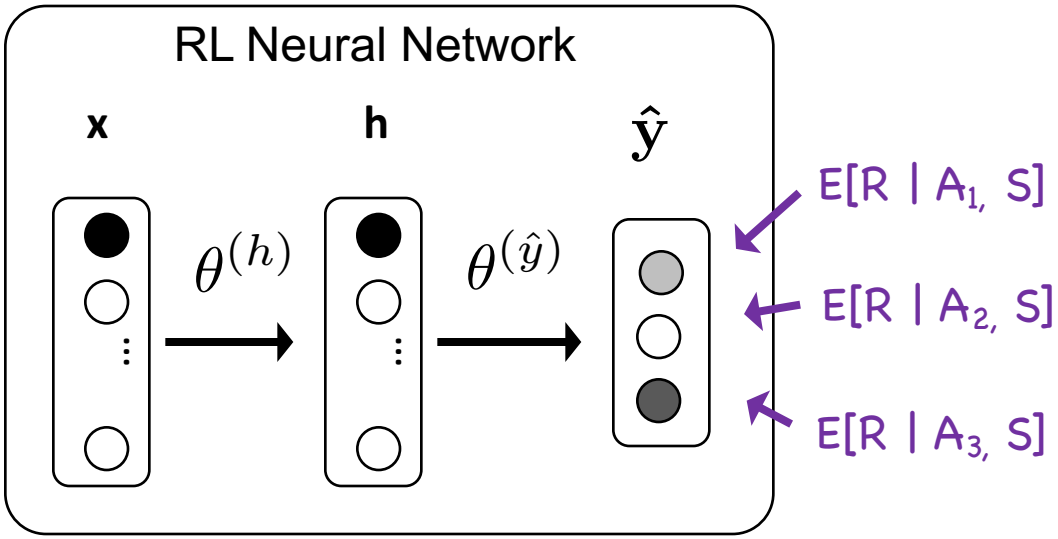
http://cs.stanford.edu/people/karpathy/convnetjs/demo/rldemo.html

Piech

Deep Reinforcement Learning

R is a reward and A_i is a legal action

Input is a representation of current state (S)



Interpret outputs as expected reward for a given action

Piech

Deep Mind Atari Games

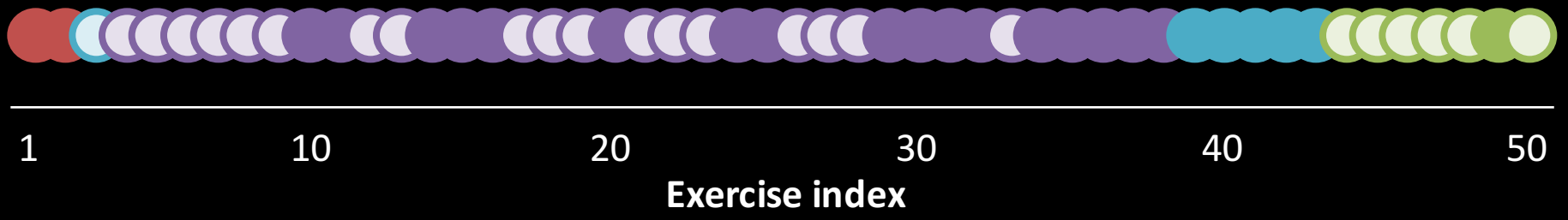
Score compared to best human

Alpha GO

When Do I Get a Robo Tutor?

Piech

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

KHAN ACADEMY

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

KHAN ACADEMY

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

KHAN ACADEMY

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

KHAN ACADEMY

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

KHAN ACADEMY

Story of Riley

Exercise Type:

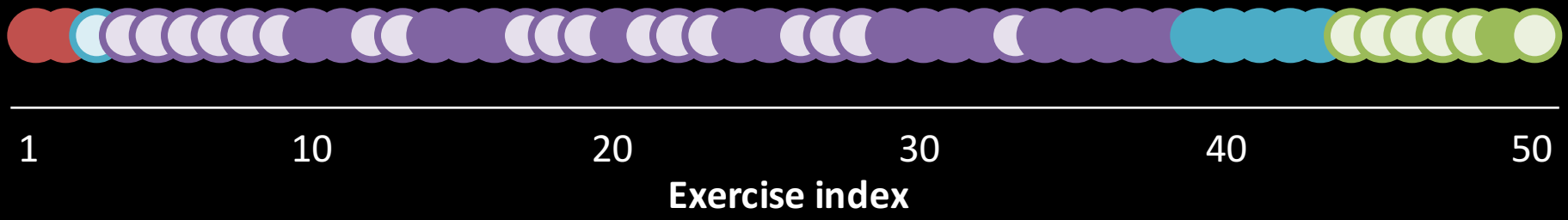
- Solving for x-intercept
- Solving for y-intercept
- Graphing linear equations
- Square roots
- Slope of a line

Answer:

- Correct
- Incorrect

KHAN ACADEMY

Story of Riley



What does Riley know?

Exercise Type:

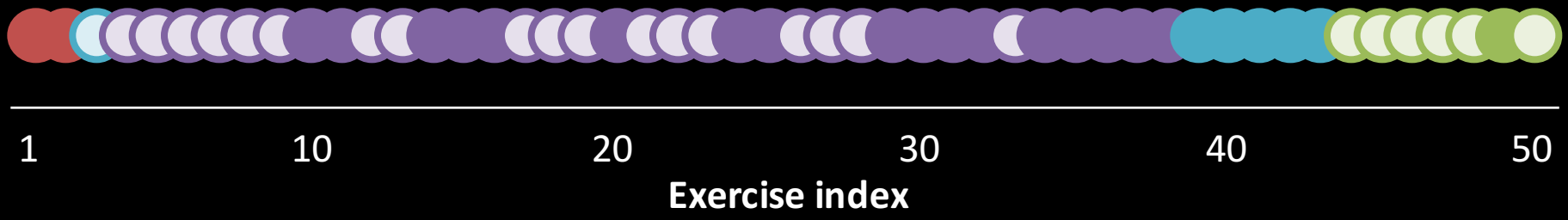
- Solving for x-intercept
- Solving for y-intercept
- Graphing linear equations
- Square roots
- Slope of a line

Answer:

- Correct
- Incorrect

KHAN ACADEMY

Story of Riley



What should Riley do next?

Exercise Type:

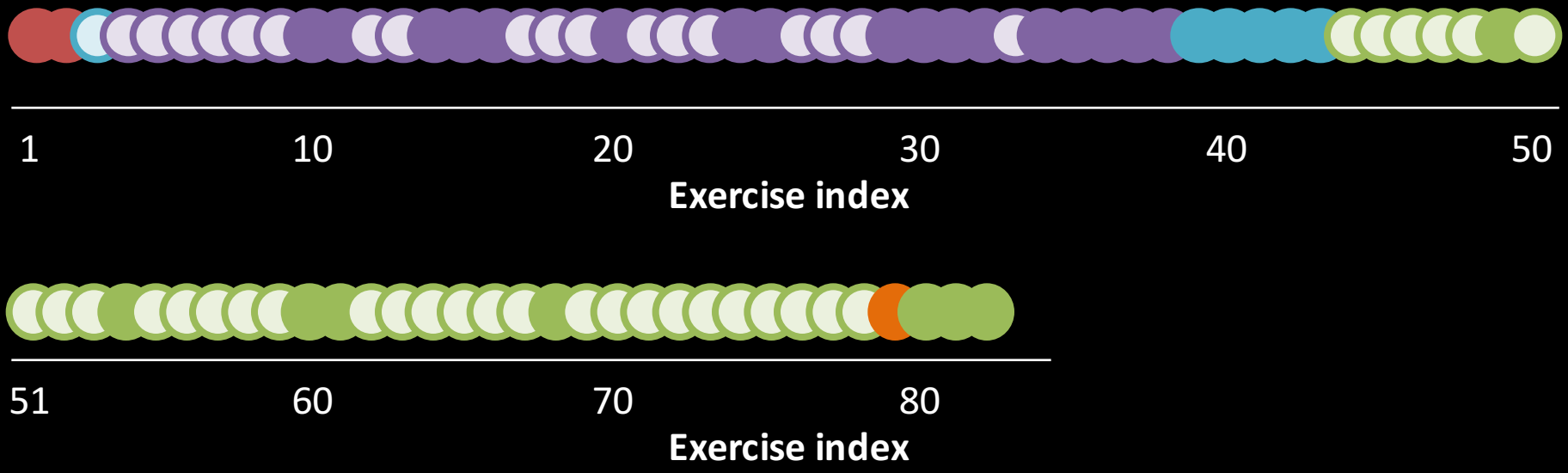
- Solving for x-intercept
- Solving for y-intercept
- Graphing linear equations
- Square roots
- Slope of a line

Answer:

- Correct
- Incorrect

KHAN ACADEMY

Story of Riley



Exercise Type:

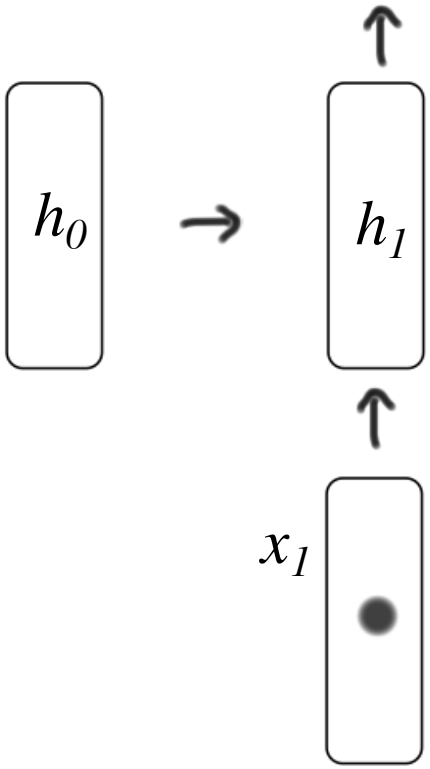
- Solving for x-intercept
- Solving for y-intercept
- Graphing linear equations
- Square roots
- Slope of a line

Answer:

- Correct
- Incorrect

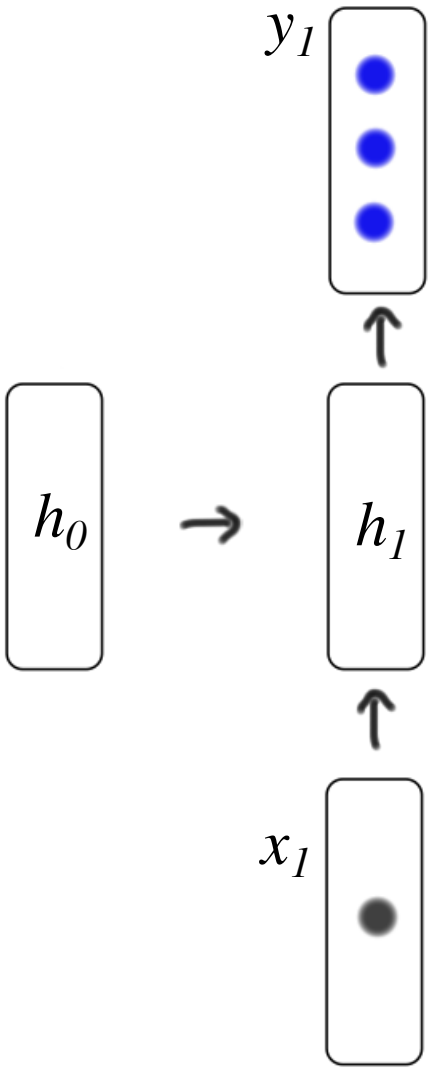
KHAN ACADEMY

Recurrent Neural Network



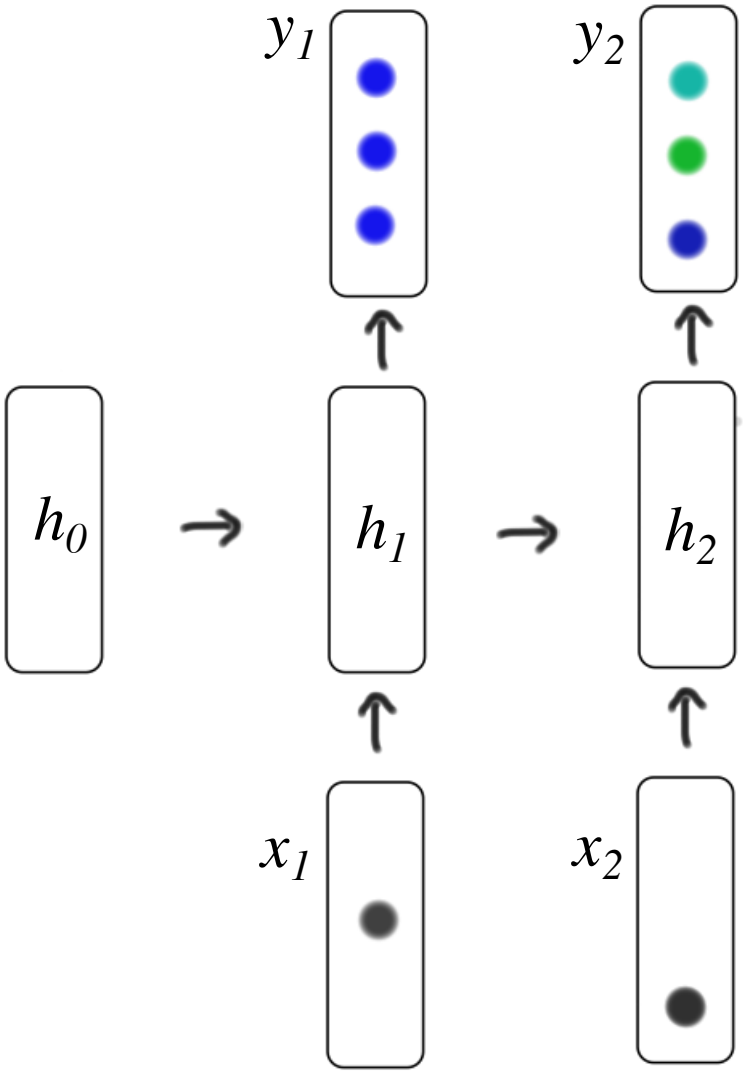
Piech

Recurrent Neural Network



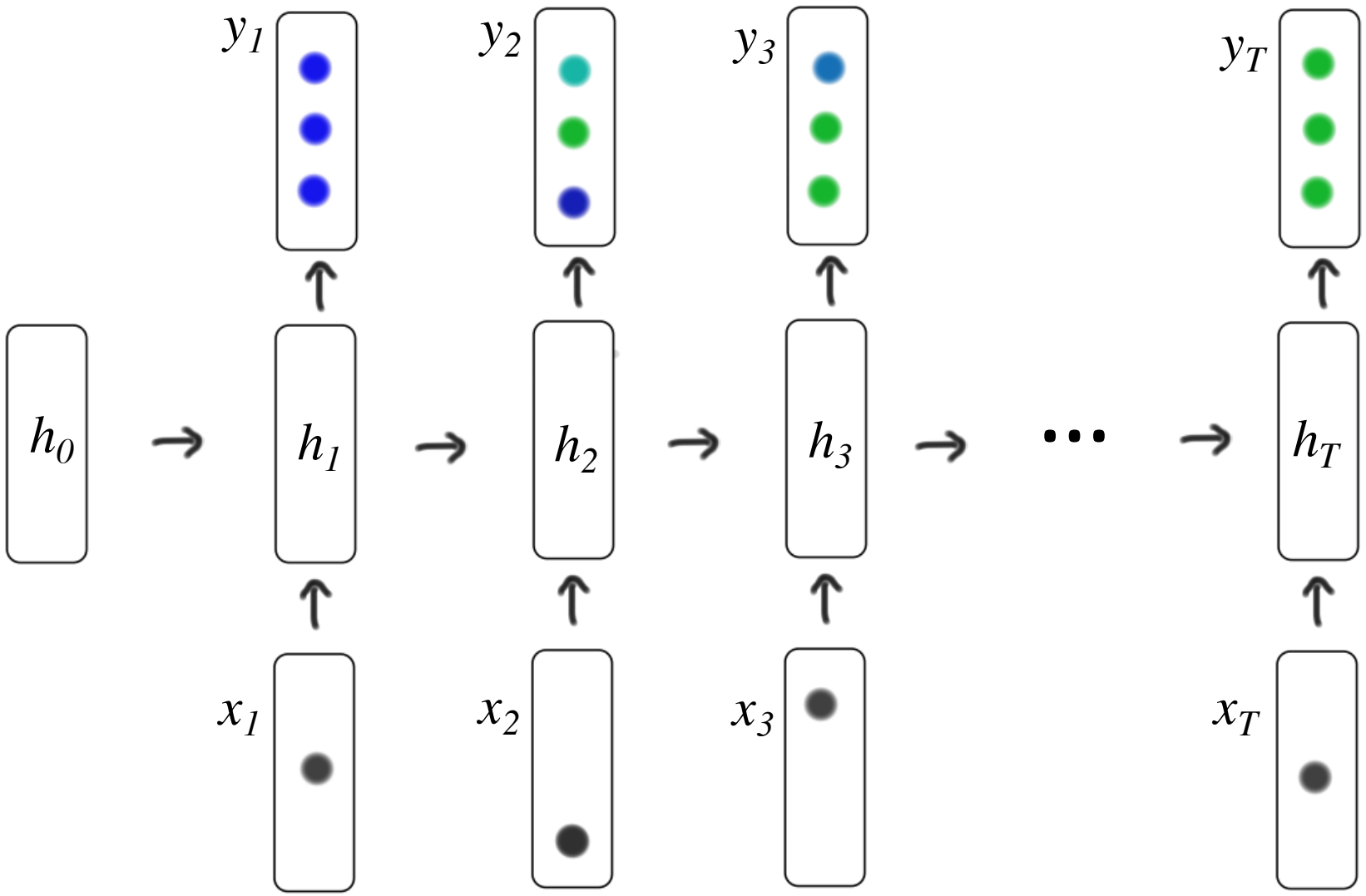
Piech

Recurrent Neural Network



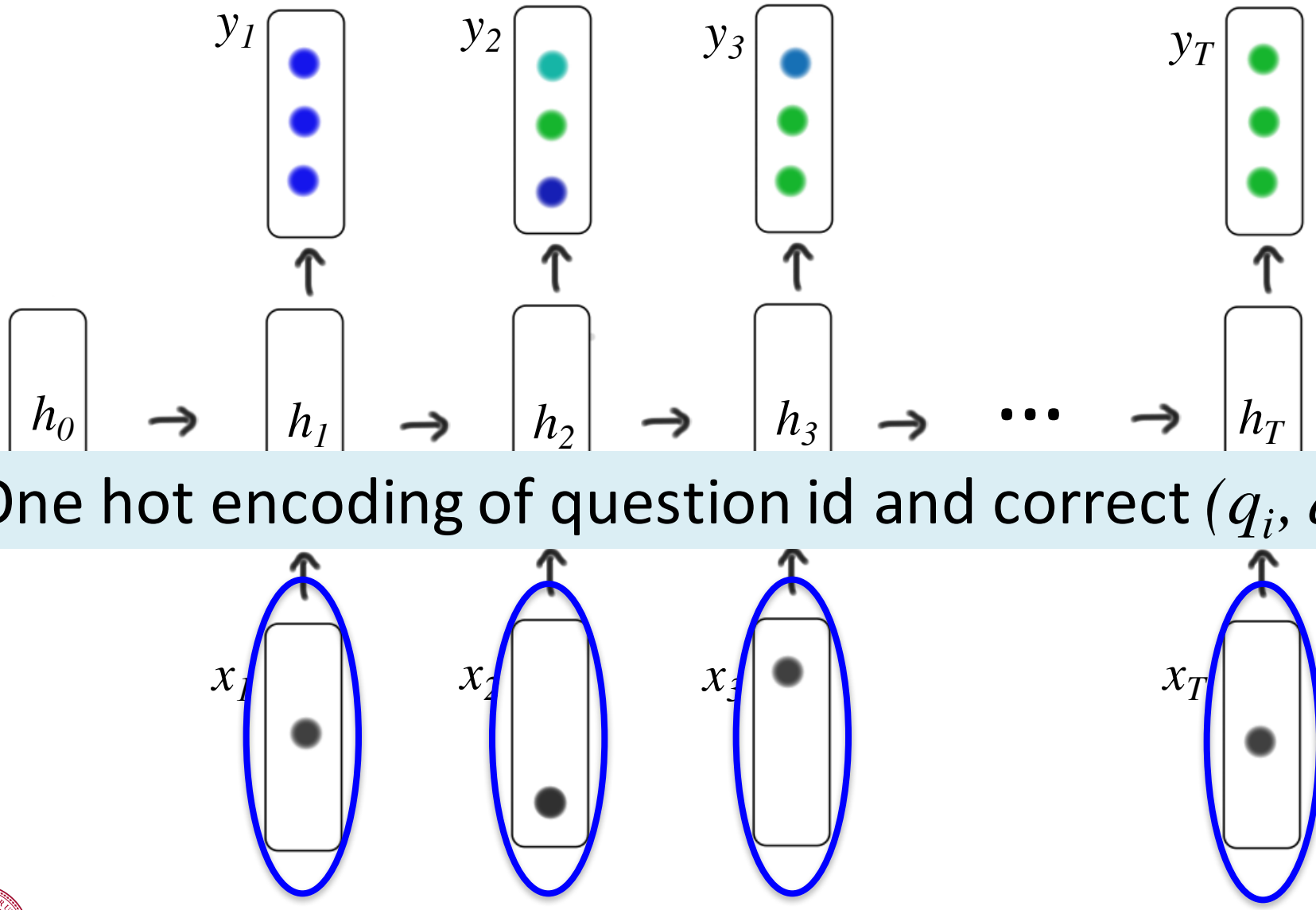
Piech

Recurrent Neural Network



Piech

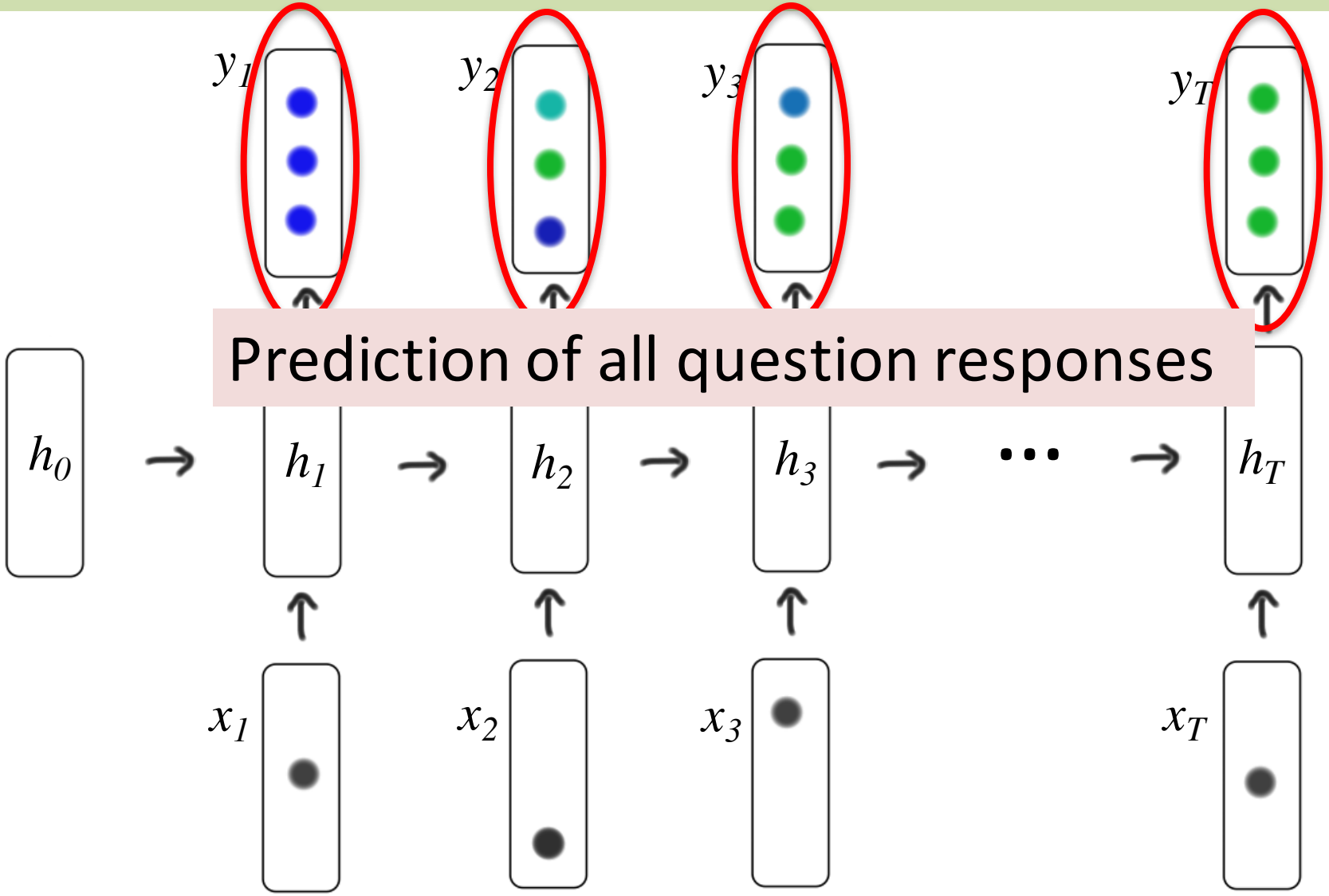
Deep Knowledge Tracing



One hot encoding of question id and correct (q_i, a_i)

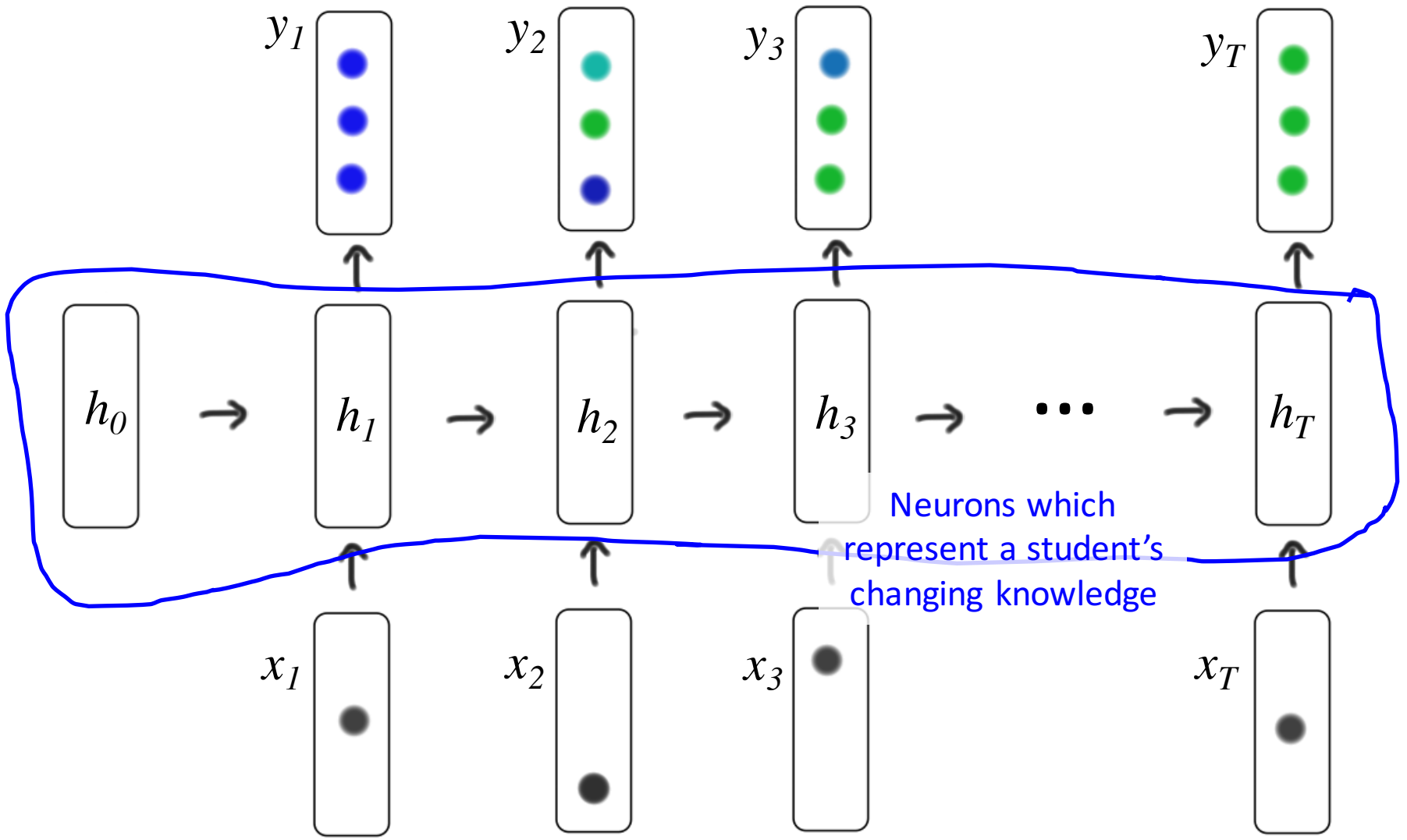
Piech

Deep Knowledge Tracing



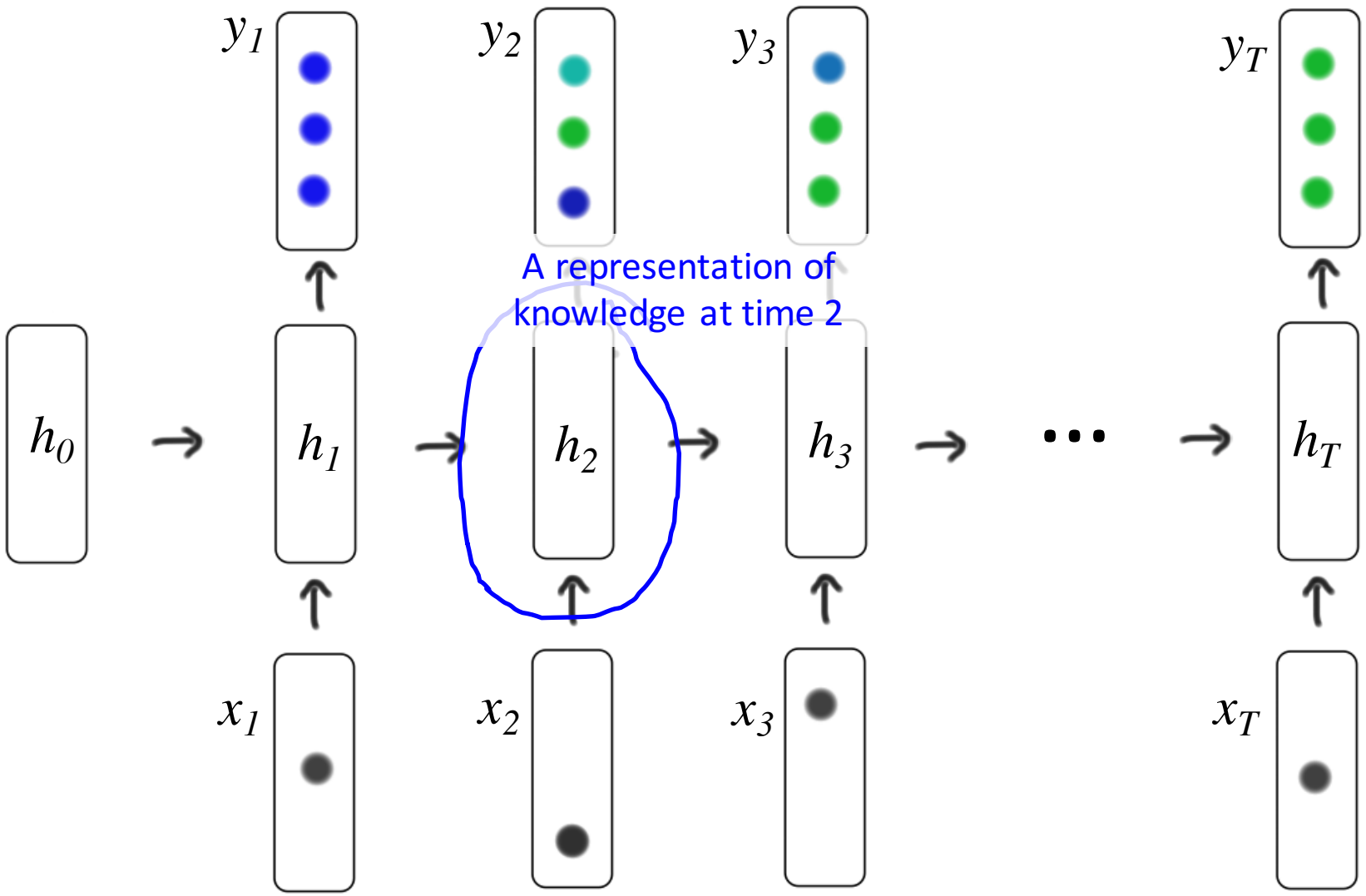
Piech

Recurrent Neural Network



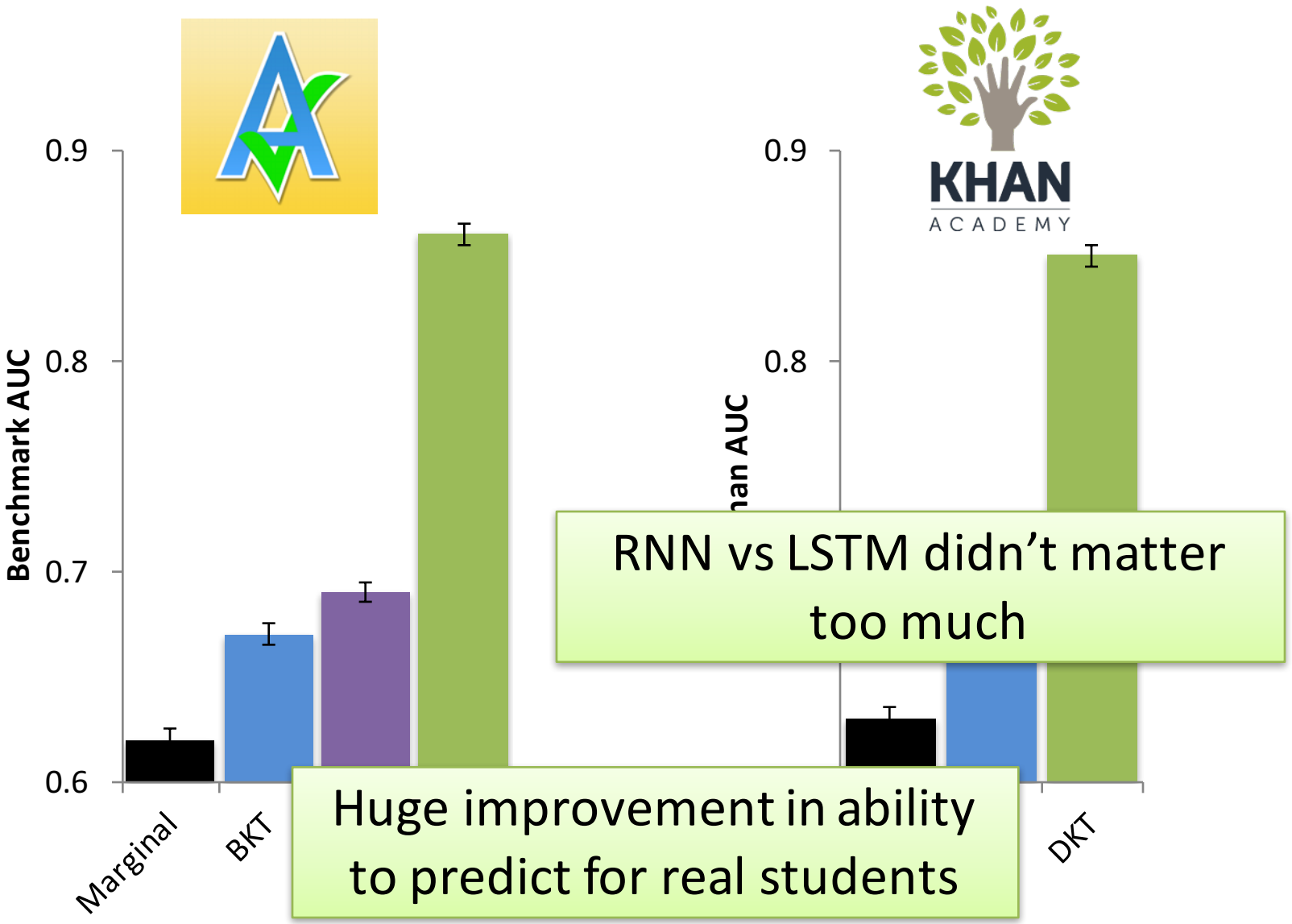
Piech

Recurrent Neural Network



Piech

Prediction Results



Piech

Interpretation

If you can't do problem *X*

Then you can't do problem *Y* either.

But, if you can do problem *X*

Then you can do problem *Y* too.

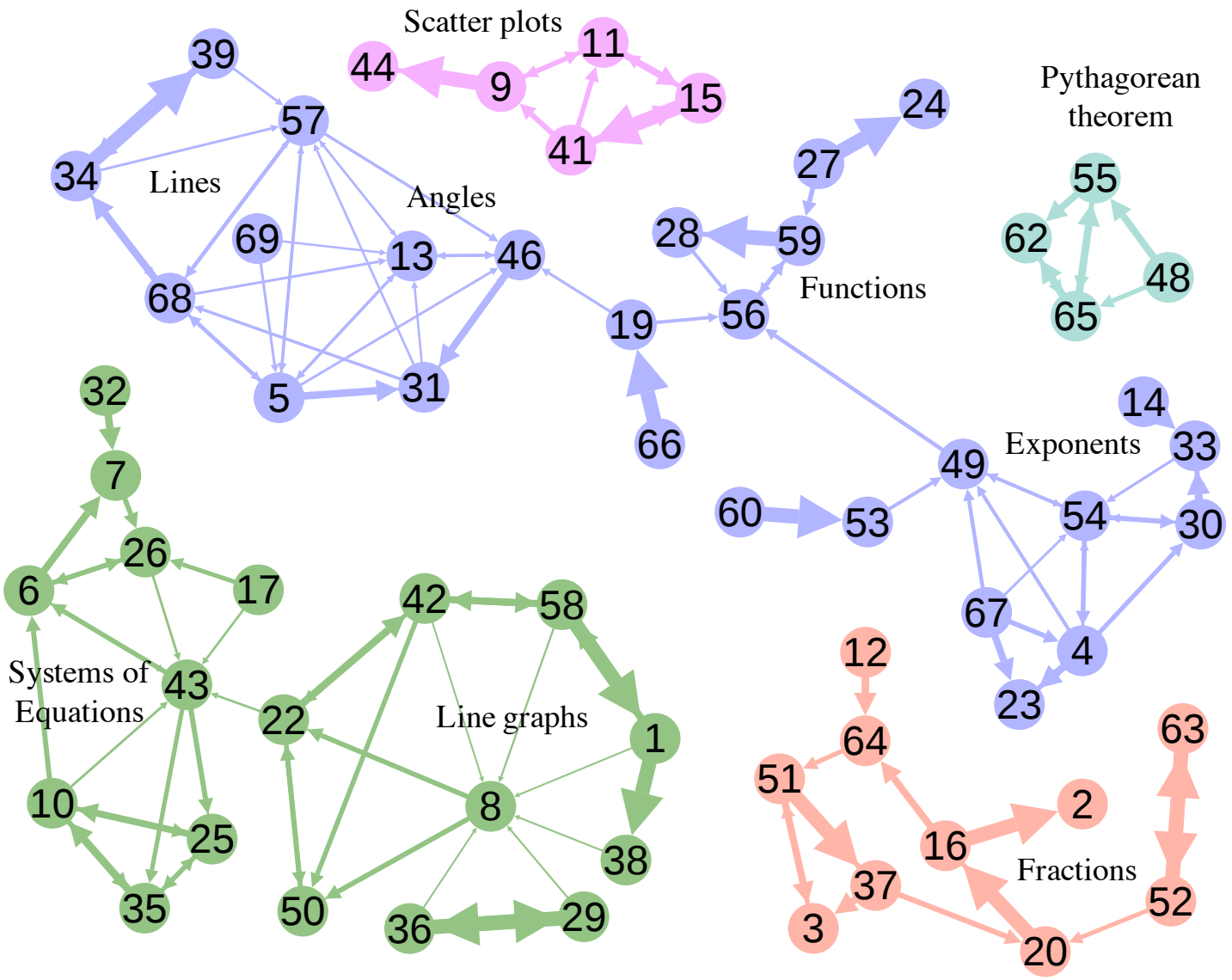
$$\rho(X,Y) = \frac{\mathrm{Cov}(X,Y)}{\sqrt{\mathrm{Var(X)Var(Y)}}}$$

$$\mathrm{Cov}(X,Y) = E[XY] - E[X]E[Y]$$

$$= P(XY) - P(X)P(Y)$$

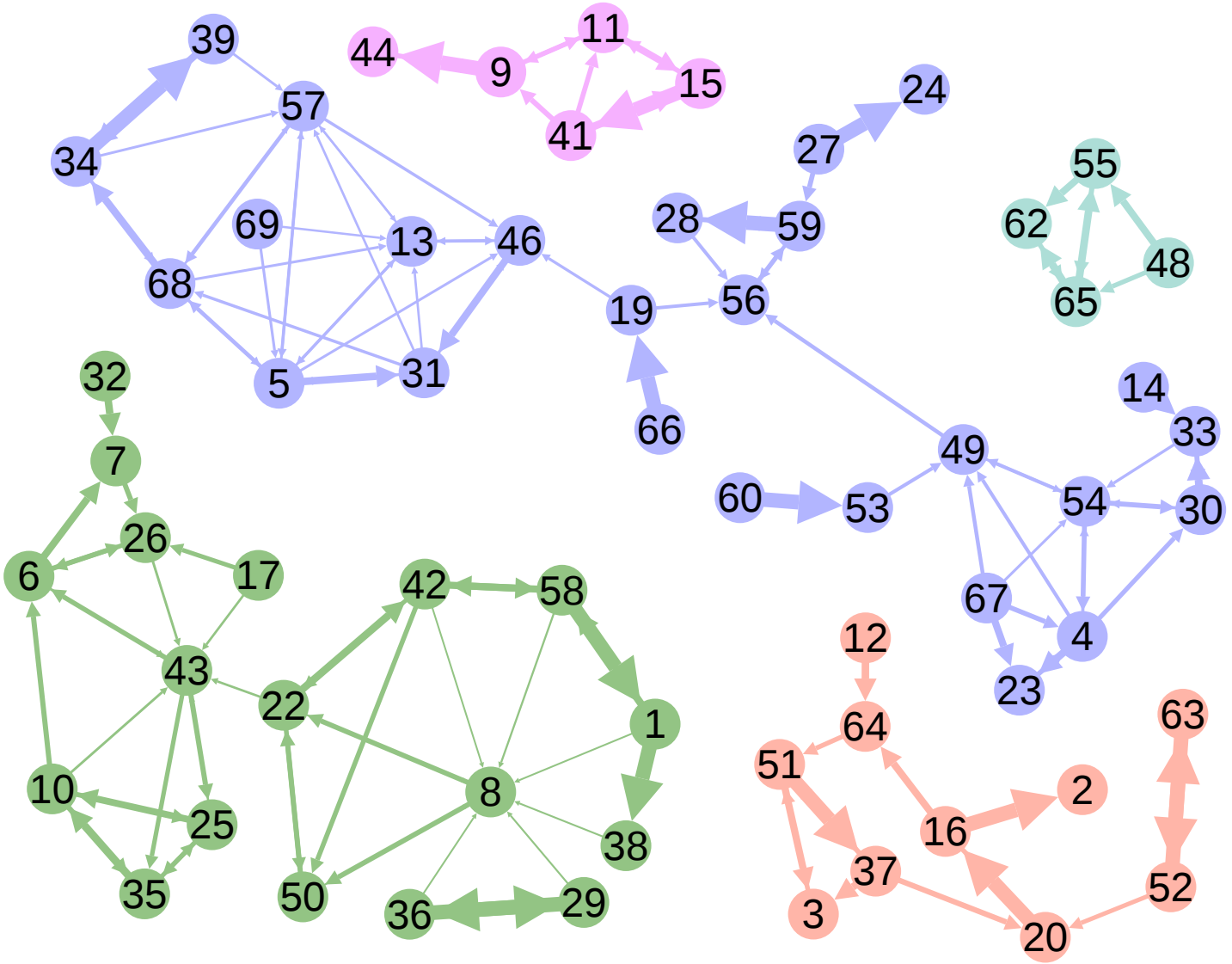
Piech

Learns Concept Relationships



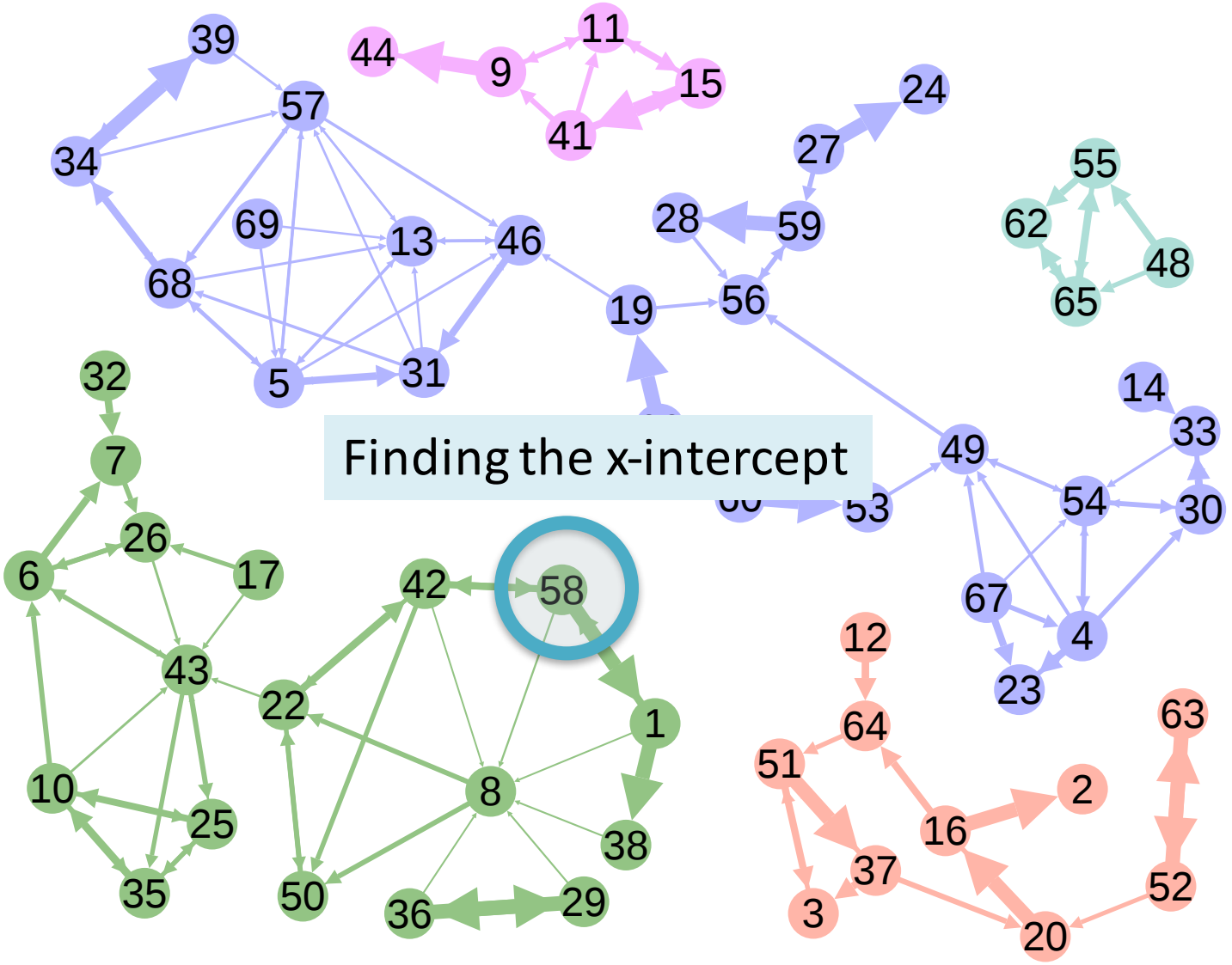
Piech

Learns Concept Relationships



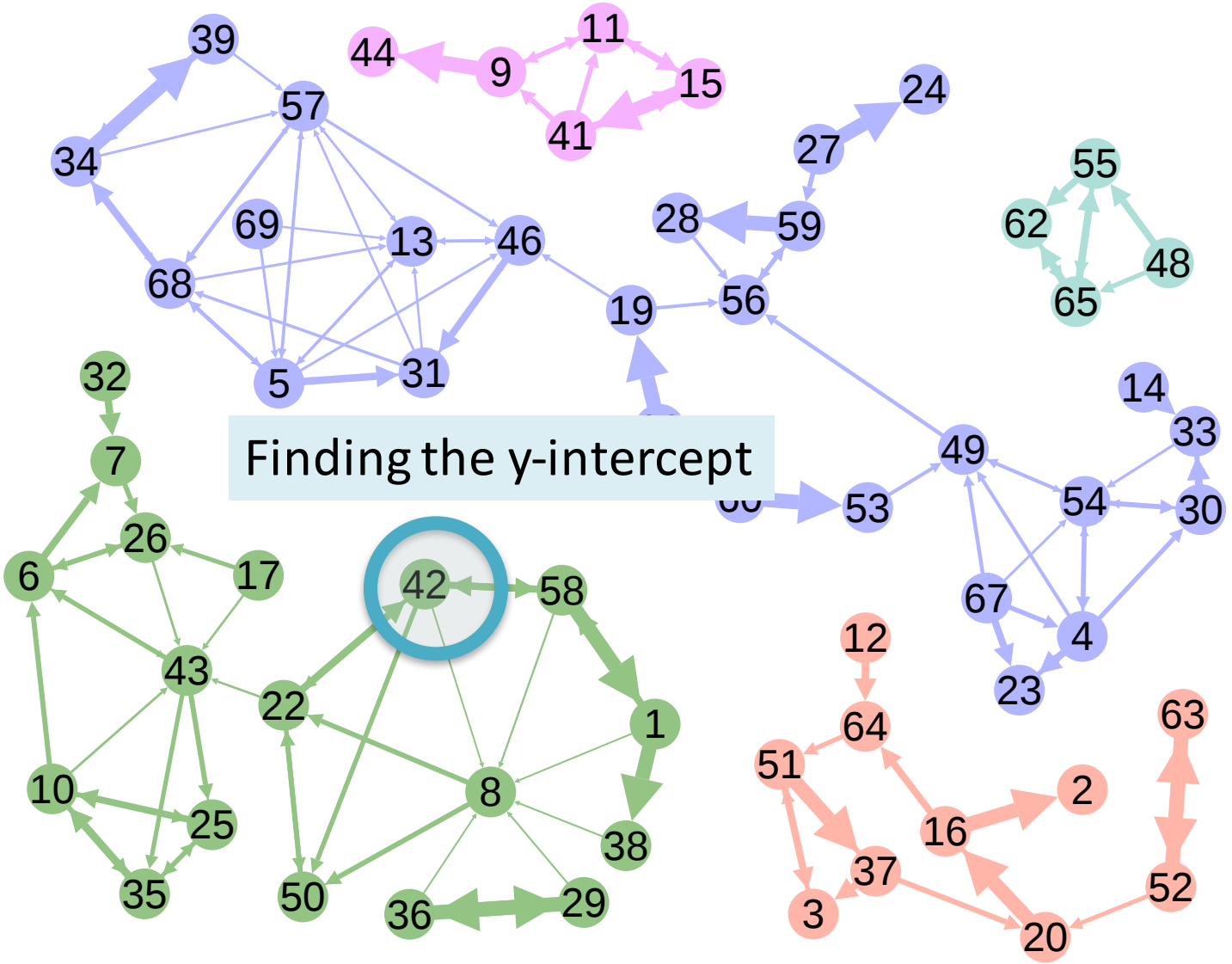
Piech

Learns Concept Relationships



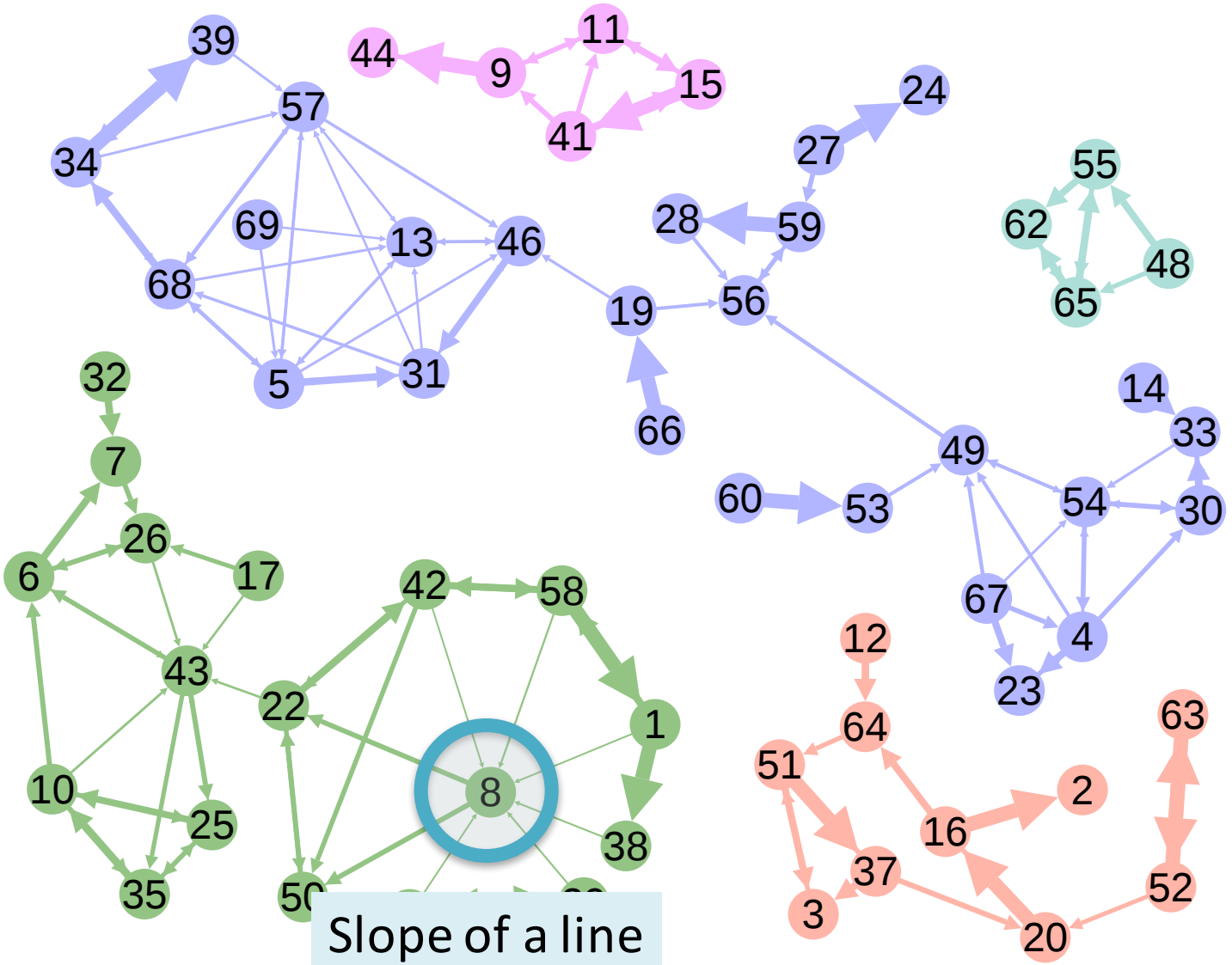
Piech

Learns Concept Relationships



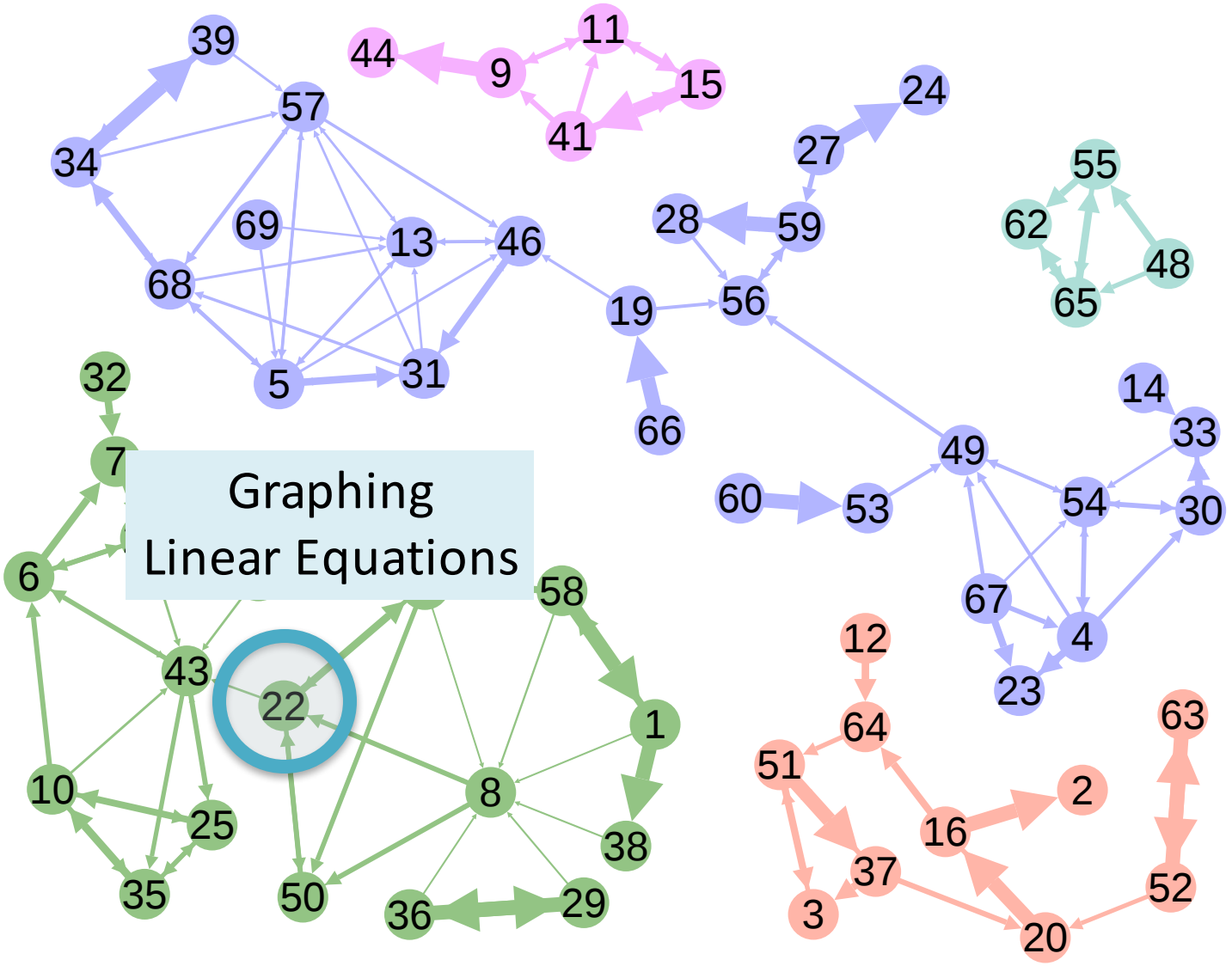
Piech

Learns Concept Relationships



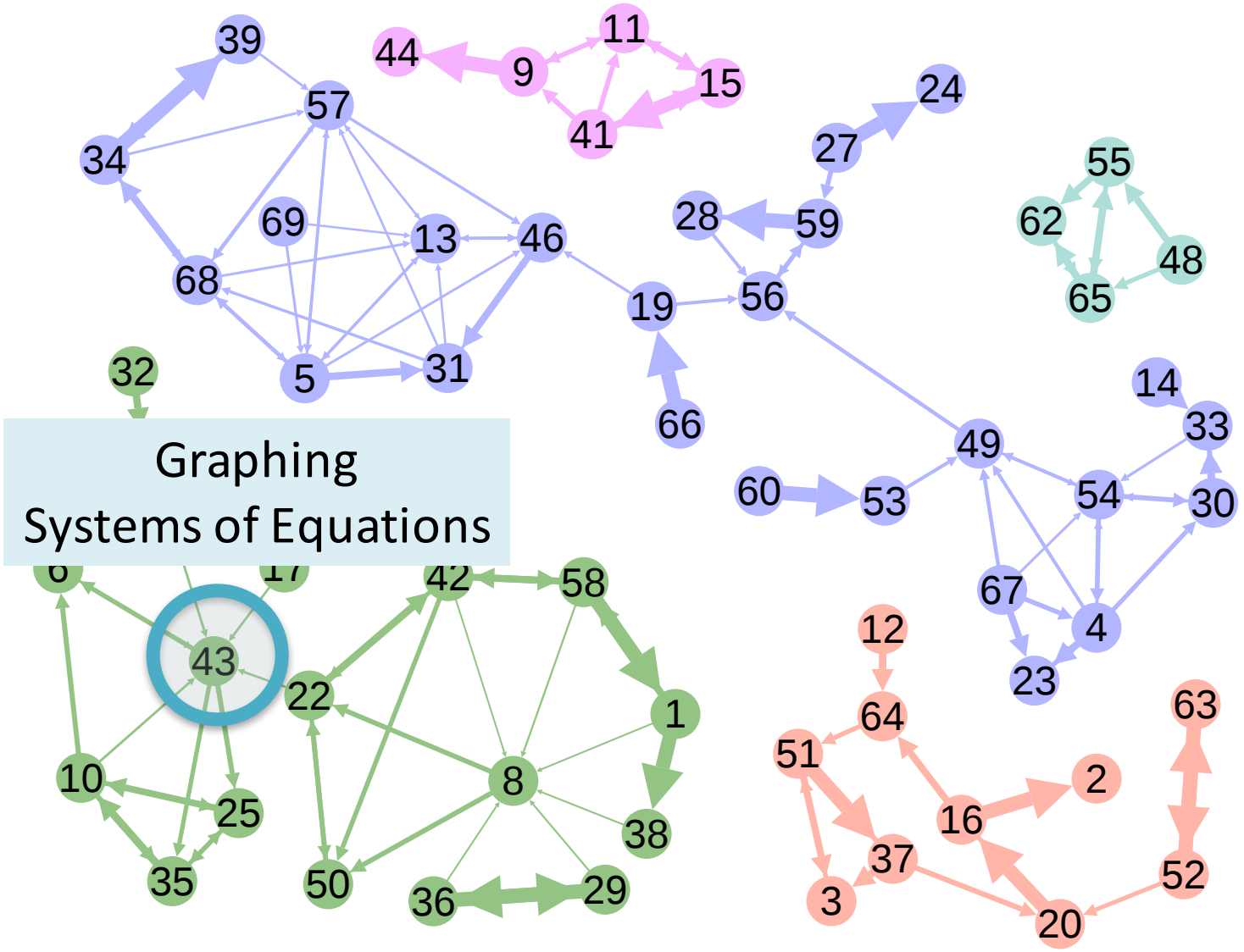
Piech

Learns Concept Relationships



Piech

Learns Concept Relationships



Graphing
Systems of Equations

Piech

Let’s try it!

Piech

Only type of AI worth knowing?

No!

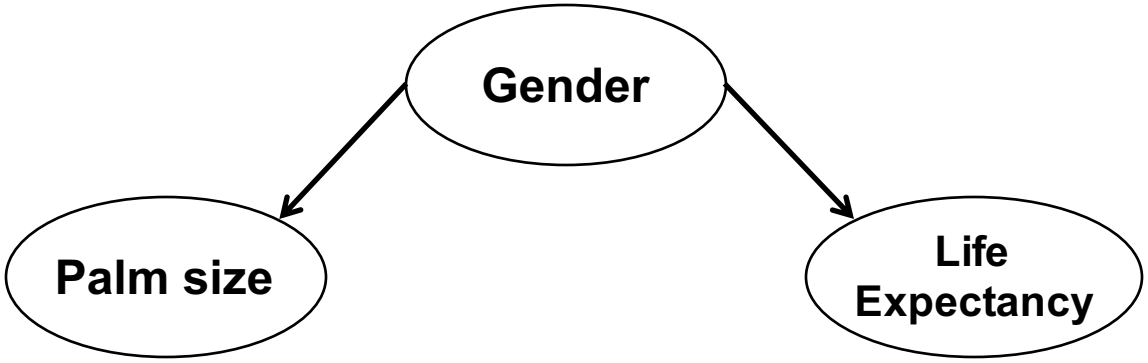
Predictions vs Understanding

- In machine learning, maintain critical perspective
  - Making predictions is only part of the story
  - Humans solve many different tasks. Don't just predict.
- Example
  - True statement: palm size negatively correlates with life expectancy
    - The larger your palm size, the shorter your life (on average)
  - Why?
    - Women have smaller palms than men on average
    - Women live 5 years longer than men on average
  - Sometimes you need better model of your domain!

Piech

Bayesian Networks

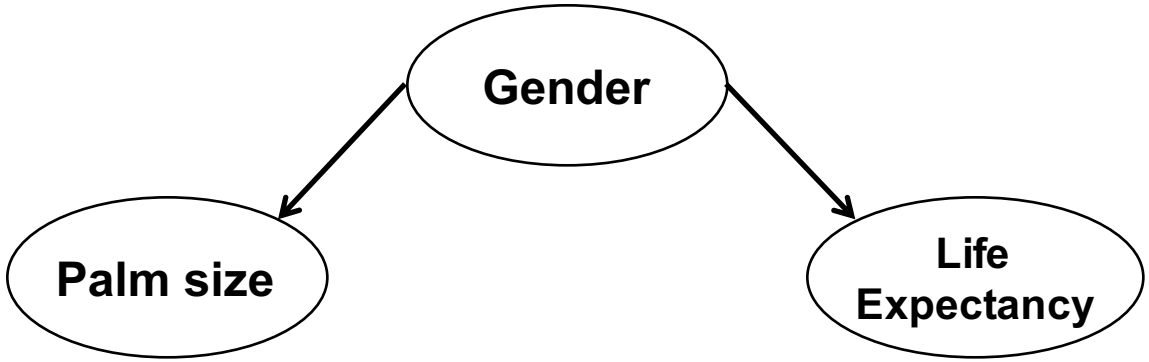
- Graphical representation of joint probability distribution



  - Node: random variable
  - Arc (X, Y): variable X has direct influence on variable Y
    - Call X a "parent" of Y
  - Each node X has conditional probability: P(X | parents(X))
  - Graph has no cycles (loops by following arcs)
    - Called "Directed Acyclic Graph" (DAG)

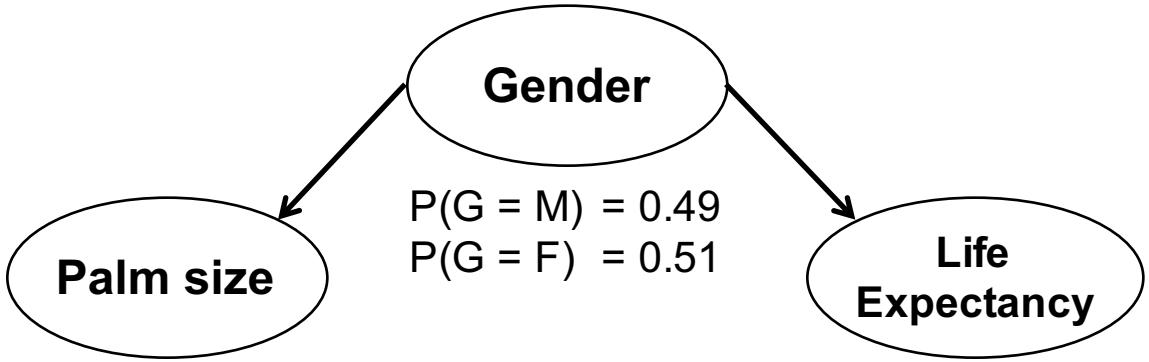
Piech

Shows Conditional Independence



- Conditional independence encoded in network
  - Each node (variable) is conditionally independent of its non-descendants, given its parents
  - In network above, Palm Size (PS) and Life Expectancy (LE) are conditionally independent, given Gender (G)
    - Formally: P(PS, LE | G) = P(PS | G) P(LE | G)
- Network structure provides insight about domain

Conditional Probability Tables



P(PS = L | G = M) = 0.45
P(PS = M | G = M) = 0.35
P(PS = S | G = M) = 0.20
P(PS = L | G = F) = 0.10
P(PS = M | G = F) = 0.30
P(PS = S | G = F) = 0.60

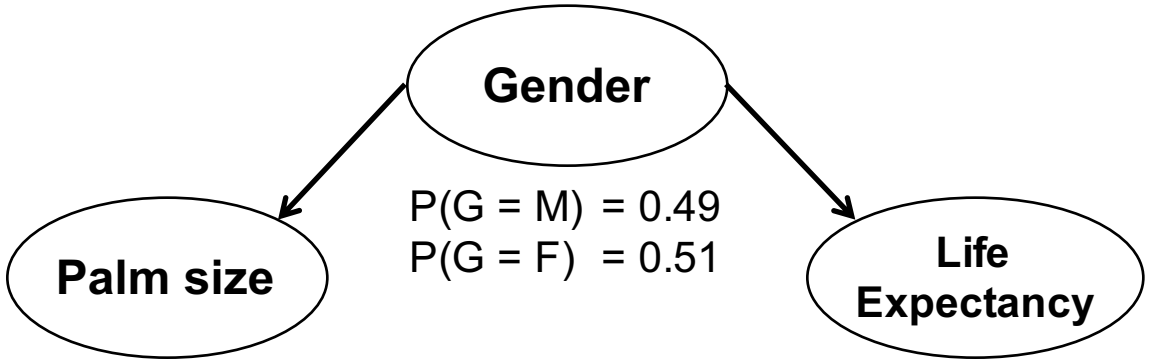
P(LE = < 70 | G = M) = 0.20
P(LE = 70-80 | G = M) = 0.50
P(LE = > 80 | G = M) = 0.30
P(LE = < 70 | G = F) = 0.10
P(LE = 70-80 | G = F) = 0.35
P(LE = > 80 | G = F) = 0.55

- Each node has conditional probability table (CPT)
  - For node X: P(X | Parents(X))
  - Conditional independence modularizes joint probability:

$$P(X_1, X_2, \ldots, X_m) = \prod_{i=1}^{m} P(X_i \mid \text{Parents}(X_i))$$

Piech

Full Joint, Fewer Parameters



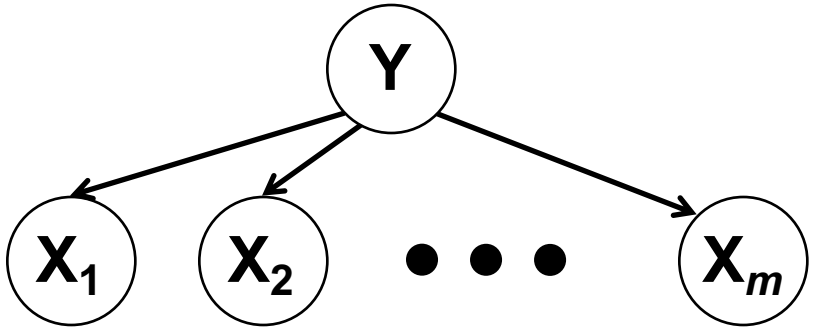
P(PS = L | G = M) = 0.45
P(PS = M | G = M) = 0.35
P(PS = S | G = M) = 0.20
P(PS = L | G = F) = 0.10
P(PS = M | G = F) = 0.30
P(PS = S | G = F) = 0.60

P(LE = < 70 | G = M) = 0.20
P(LE = 70-80 | G = M) = 0.50
P(LE = > 80 | G = M) = 0.30
P(LE = < 70 | G = F) = 0.10
P(LE = 70-80 | G = F) = 0.35
P(LE = > 80 | G = F) = 0.55

- Each node has conditional probability table (CPT)
  - Reduces number of parameters needed in model
  - Normally, need 2 x 3 x 3 – 1 = 18 – 1 = 17 parameters
  - Here, need (2 – 1) + (6 – 2) + (6 – 2) = 9 parameters

Naïve Bayes is a Bayesian Network

- Welcome back, Naïve Bayes…
  - Now with new and improved "Bayesian Network" flavor!



  - Network structure encodes assumption:

$$P(\boldsymbol{X} \mid Y) = P(X_1, X_2, \ldots, X_m \mid Y) = \prod_{i=1}^{m} P(X_i \mid Y)$$

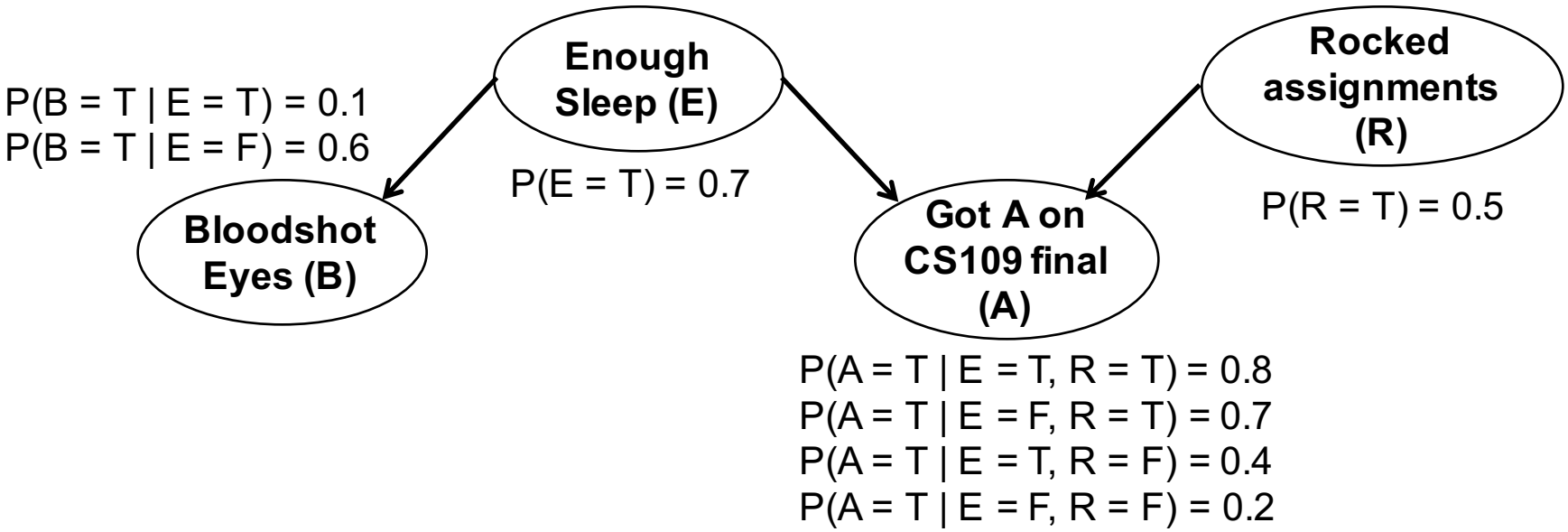
  - Full joint distribution can be computed as:

$$P(\boldsymbol{X}, Y) = P(Y)P(\boldsymbol{X} \mid Y) = P(Y)\prod_{i=1}^{m} P(X_i \mid Y)$$

Piech

Answer Many Questions

- Consider the following Bayes Net:



- Determine $P(A = T \mid B = T, R = T)$

The future of artificial intelligence is uncertain

Open questions:

Natural Language Processing?

One Shot Learning?

General AI?

How will these ideas help improve the quality of human life?

You tell me.

Probability is a phenomenal toolset