Announcements

Assignments:

- HW4
  - Due date Mon, 2/24, 11:59 pm

Midterm

- Monday the 2nd of March from 5:00pm-6:30pm

Midterm Conflicts

- See Piazza post
- Due 11:59pm on Wednesday the 19th of February

Plan

Last time

- Naïve Bayes Assumptions
- Naïve Bayes MLE and MAP
- MLE vs MAP
- Generative vs Discriminative Models

$p(Y|X)$

$p(X|Y)p(Y)$

Today

- Decision Boundaries
- Gaussian Generative Models
- Neural Networks

Introduction to Machine Learning

Generative Models
then
Intro to Neural Networks

Instructor: Pat Virtue

Decision Boundaries

Decision boundary

- The set of points in the domain of the input ($\boldsymbol{x}$) where the predicted classification changes

Two class decision boundary

- So far, we have decided to let the decision boundary be all $\boldsymbol{x}$ such that:

$$p(y = 0 \mid \boldsymbol{x}) = p(y = 1 \mid \boldsymbol{x})$$

- What assumptions are we making here?
  - This assumes that the cost of predicting it wrong is the same for both classes

Piazza Poll 1

Which of the following also define the decision boundary for two classes when we just want $p(Y = 0 \mid \boldsymbol{x}) = p(Y = 1 \mid \boldsymbol{x})$?

A. All $\boldsymbol{x}$, s.t. $p(\boldsymbol{x} \mid Y = 0) = p(\boldsymbol{x} \mid Y = 1)$
B. All $\boldsymbol{x}$, s.t. $p(\boldsymbol{x}, Y = 0) = p(\boldsymbol{x}, Y = 1)$
C. All $\boldsymbol{x}$, s.t. $p(Y = 0) = p(Y = 1)$
D. All $\boldsymbol{x}$, s.t. $p(Y = 1 \mid \boldsymbol{x}) = 0.5$
E. All $\boldsymbol{x}$, s.t. $p(\boldsymbol{x} \mid Y = 1) = 0.5$
F. All $\boldsymbol{x}$, s.t. $p(\boldsymbol{x}, Y = 1) = 0.5$
G. All $\boldsymbol{x}$, s.t. $\log p(\boldsymbol{x}, Y = 1) - \log p(\boldsymbol{x}, Y = 0) = 0$
H. None of the above

$p(Y=0 \mid x) \propto p(x, Y=0) = P(x \mid Y=0)P(Y=0)$

$P(Y=0) = P(Y=1)$

$p(x, Y=0) = p(x, Y=1)$

Piazza Poll 2

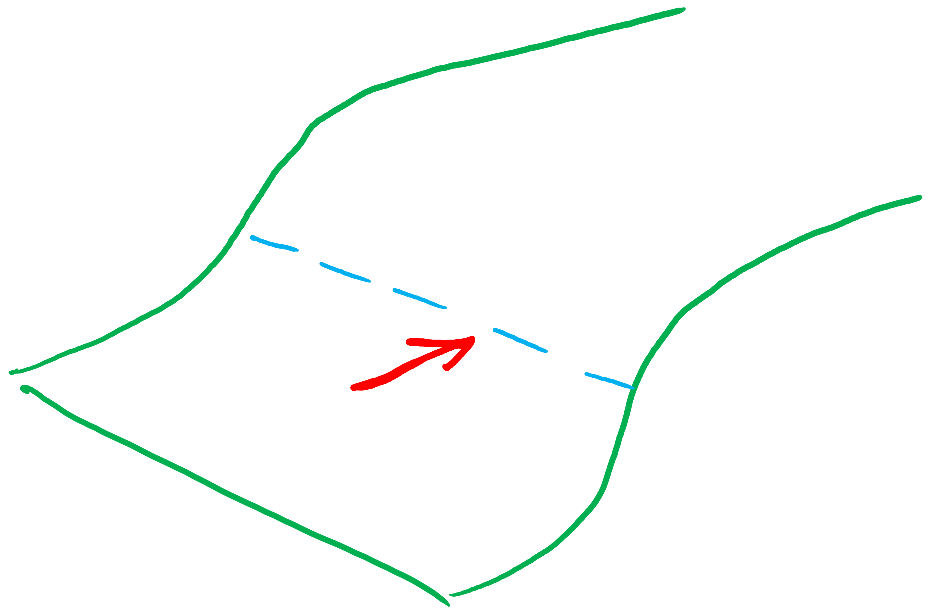
$$\hat{y} = g(w^T x)$$

True/False: Logistic regression always produces a linear decision boundary.

A. I don't know

B. True

C. False



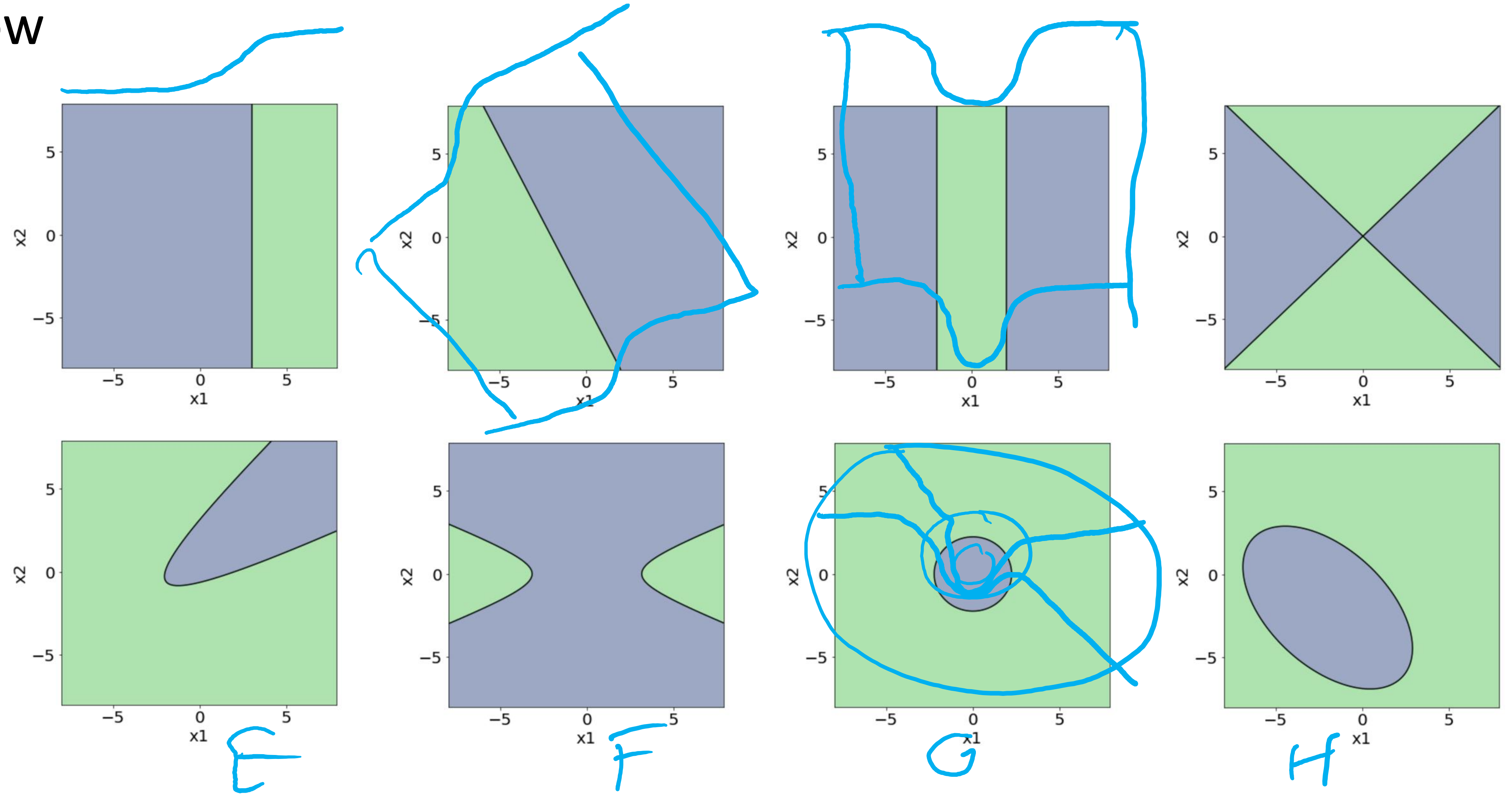
Piazza Poll 2

True/False: Logistic regression always produces a linear decision boundary.

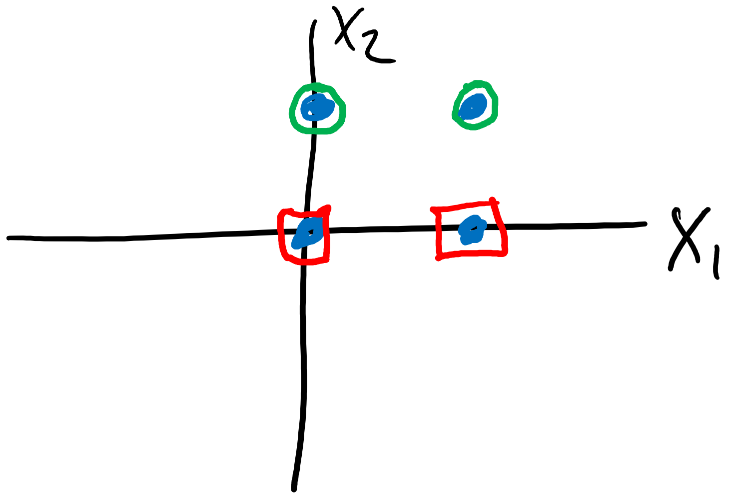
A. I don’t know

B. True

C. False



Generative Models



SPAM:

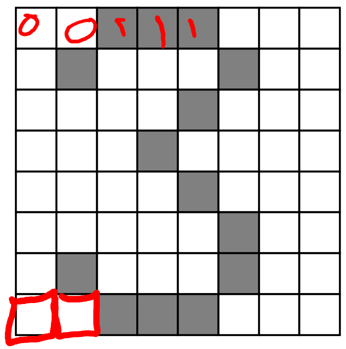
- Class distribution: $Y \sim \ Bern(\phi)$
- Class conditional distribution: $X_m \sim \ Bern(\theta_{m,y})$
- Naïve Bayes X_i conditionally independent X_j given Y for all $i \neq j$

$$p(X_i, X_j \mid Y) \ = \ p(X_i \mid Y) \mid p(X_j \mid Y)$$

Digits:

- Class distribution: $Y \sim \ Multinomial(\phi, 1)$
- Class conditional distribution: $X_m \sim \ Bern(\theta_{m,y})$
- Naïve Bayes X_i conditionally independent X_j given Y for all $i \neq j$

$$p(X_i, X_j \mid Y) \ = \ p(X_i \mid Y) \mid p(X_j \mid Y)$$



Recitation?

Fisher Iris Dataset

https://en.wikipedia.org/wiki/Iris_flower_data_set

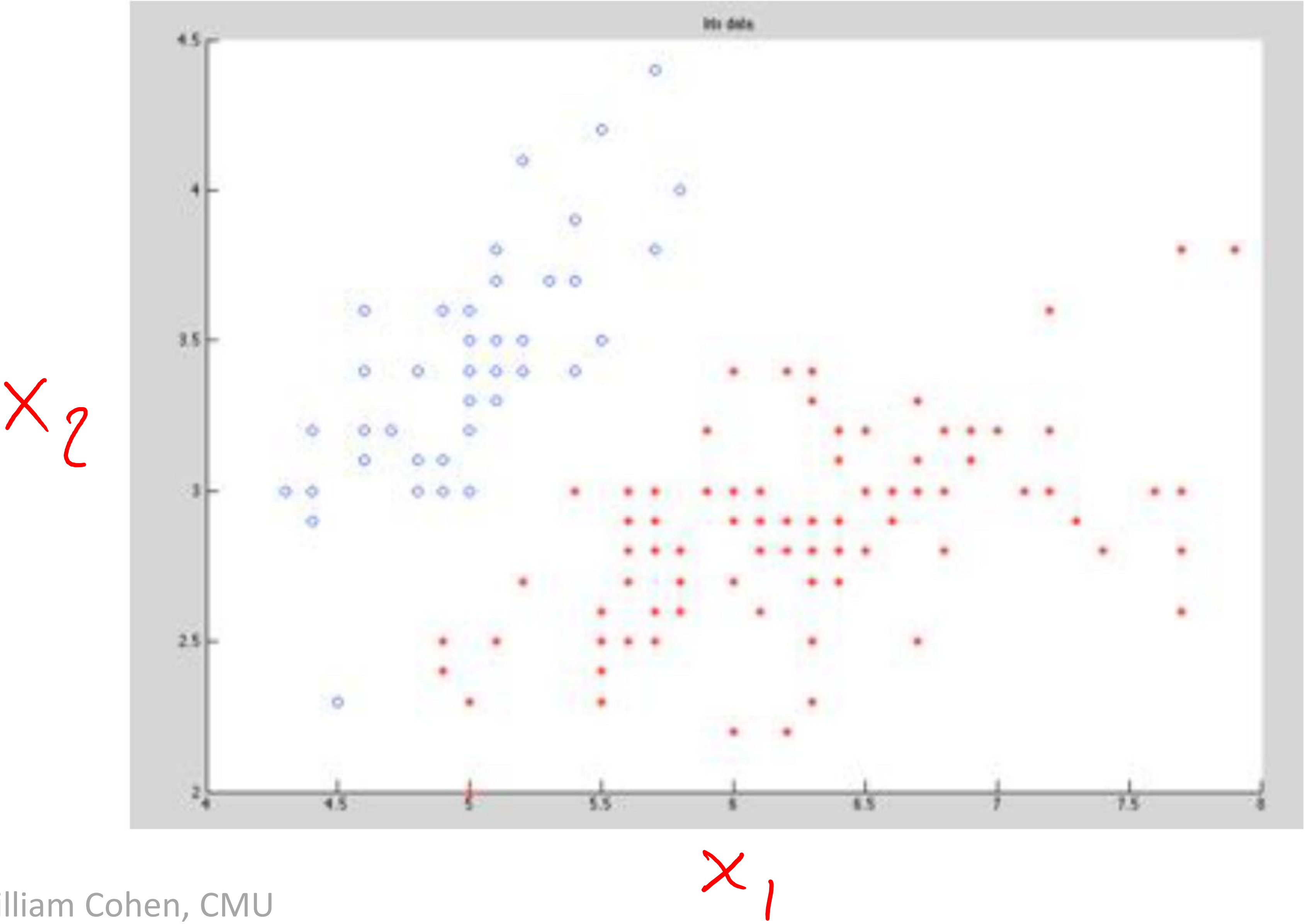
Fisher Iris Dataset

https://en.wikipedia.org/wiki/Iris_flower_data_set

Fisher (1936) used 150 measurements of flowers from 3 different species: Iris setosa (0), Iris virginica (1), Iris versicolor (2) collected by Anderson (1936)

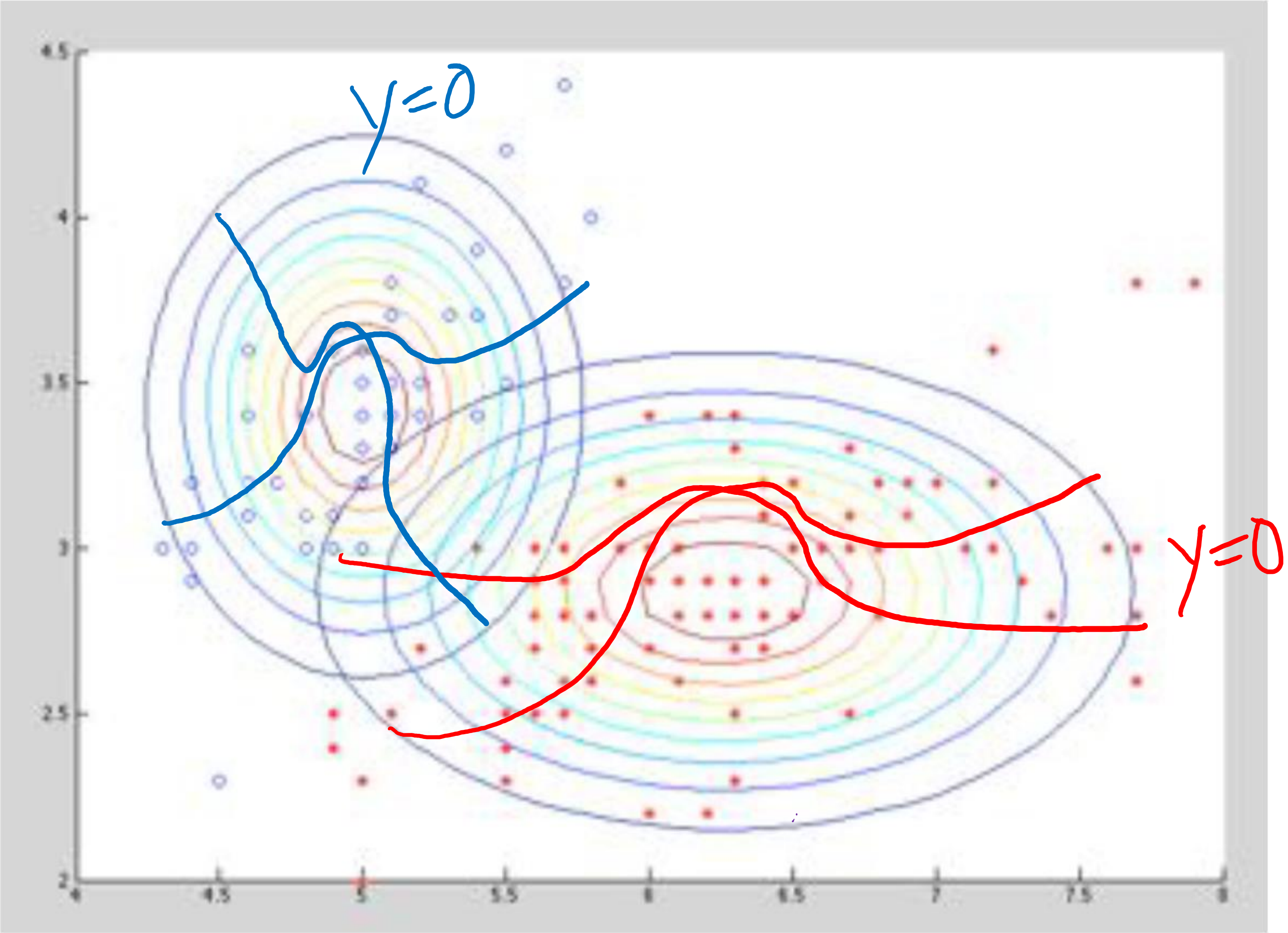
Species	Sepal Length	Sepal Width	Petal Length	Petal Width
0	4.3	3.0	1.1	0.1
0	4.9	3.6	1.4	0.1
0	5.3	3.7	1.5	0.2
1	4.9	2.4	3.3	1.0
1	5.7	2.8	4.1	1.3
1	6.3	3.3	4.7	1.6
1	6.7	3.0	5.0	1.7

Fisher Iris Dataset



Slides from William Cohen, CMU

Fisher Iris Dataset



$p(x|y)$
$f(x|y)$
$f(x|y=0)$
$f(x|y=1)$

Generative Models with Continuous Features

Iris dataset:

- Class distribution: $Y \sim \ Bern(\phi)$
- Class conditional distribution: $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y)$
- Naïve Bayes assumption?

Piazza Poll 3

Iris dataset:

- Class distribution: $Y \sim \ Bern(\phi)$
- Class conditional distribution: $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y)$
- Naïve Bayes assumption?

Which of the following pairs of Gaussian class conditional distributions satisfy the Naïve Bayes assumptions? Select ALL that apply.

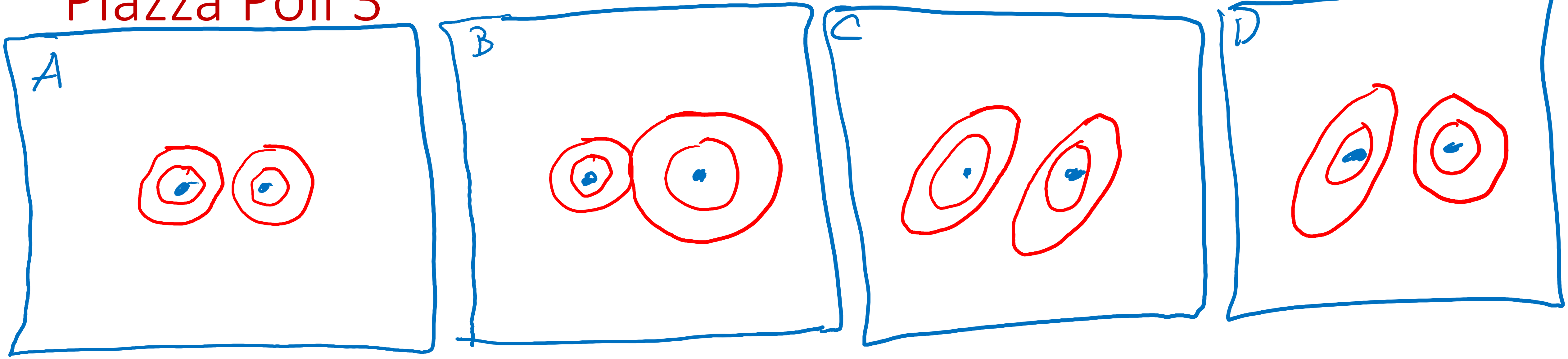
$A.\ \ \boldsymbol{\mu}_{y=0} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \boldsymbol{\Sigma}_{y=0} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \boldsymbol{\mu}_{y=1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \boldsymbol{\Sigma}_{y=1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

$B.\ \ \boldsymbol{\mu}_{y=0} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \boldsymbol{\Sigma}_{y=0} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \boldsymbol{\mu}_{y=1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \boldsymbol{\Sigma}_{y=1} = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}$

$C.\ \ \boldsymbol{\mu}_{y=0} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \boldsymbol{\Sigma}_{y=0} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}, \qquad \boldsymbol{\mu}_{y=1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \boldsymbol{\Sigma}_{y=1} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$

$D.\ \ \boldsymbol{\mu}_{y=0} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \boldsymbol{\Sigma}_{y=0} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}, \qquad \boldsymbol{\mu}_{y=1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \boldsymbol{\Sigma}_{y=1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

Piazza Poll 3



A. $$\boldsymbol{\mu}_{y=0} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \boldsymbol{\Sigma}_{y=0} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \boldsymbol{\mu}_{y=1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \boldsymbol{\Sigma}_{y=1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

B. $$\boldsymbol{\mu}_{y=0} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \boldsymbol{\Sigma}_{y=0} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \boldsymbol{\mu}_{y=1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \boldsymbol{\Sigma}_{y=1} = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}$$

C. $$\boldsymbol{\mu}_{y=0} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \boldsymbol{\Sigma}_{y=0} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}, \quad \boldsymbol{\mu}_{y=1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \boldsymbol{\Sigma}_{y=1} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$

D. $$\boldsymbol{\mu}_{y=0} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \boldsymbol{\Sigma}_{y=0} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}, \quad \boldsymbol{\mu}_{y=1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \boldsymbol{\Sigma}_{y=1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Decision Boundaries

Iris dataset:

- Class distribution: $Y \sim \ Bern(\phi)$
- Class conditional distribution: $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y)$

- Naïve Bayes assumption: diagonal Σ_y
- Linear Decision Boundary: $\Sigma_{y=0} = \Sigma_{y=1}$
- Quadradic Decision Boundary:

$$\Sigma_{y=0} \neq \Sigma_{y=1}$$

$p(Y|X)$

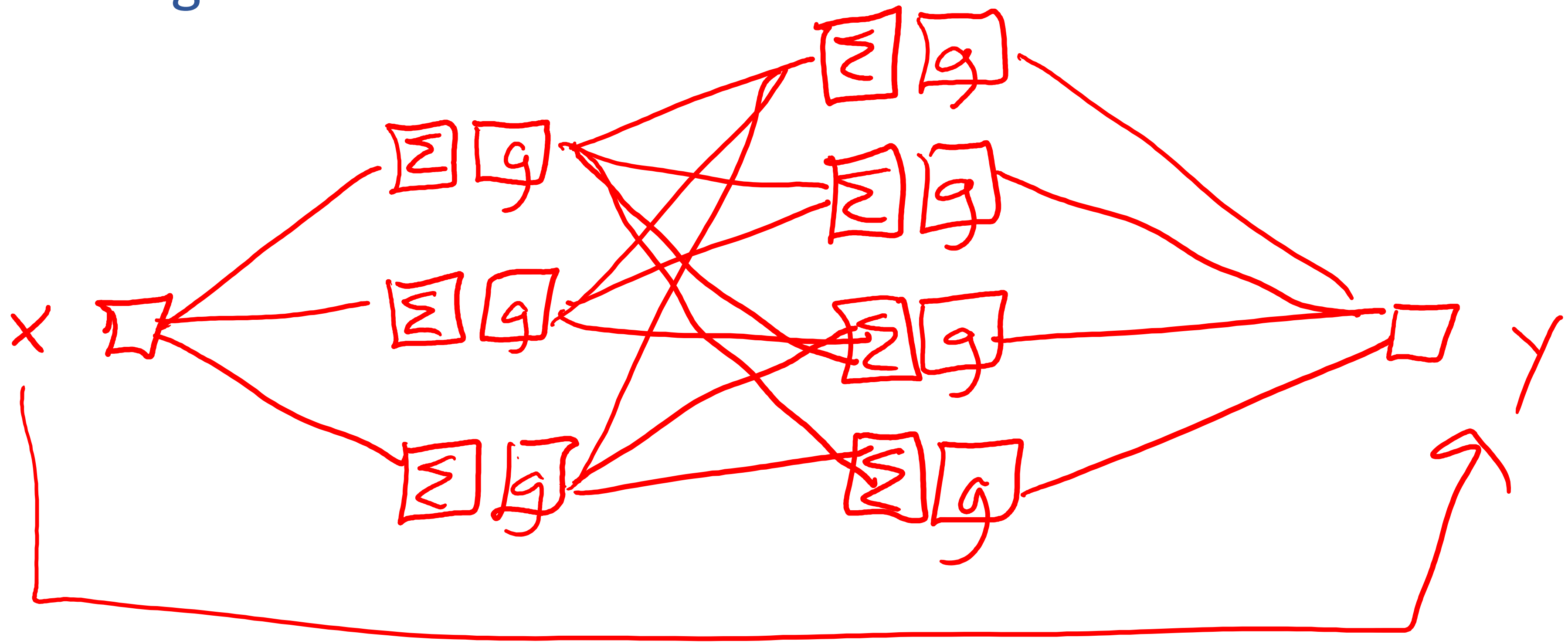
Introduction to Machine Learning

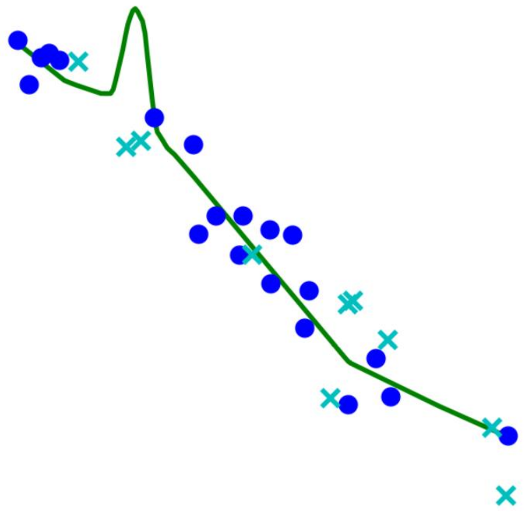
Intro to Neural Networks

Instructor: Pat Virtue

Neural Networks from HW2

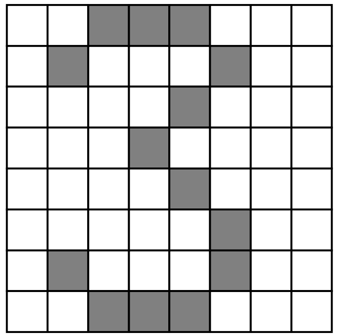
1-D Regression

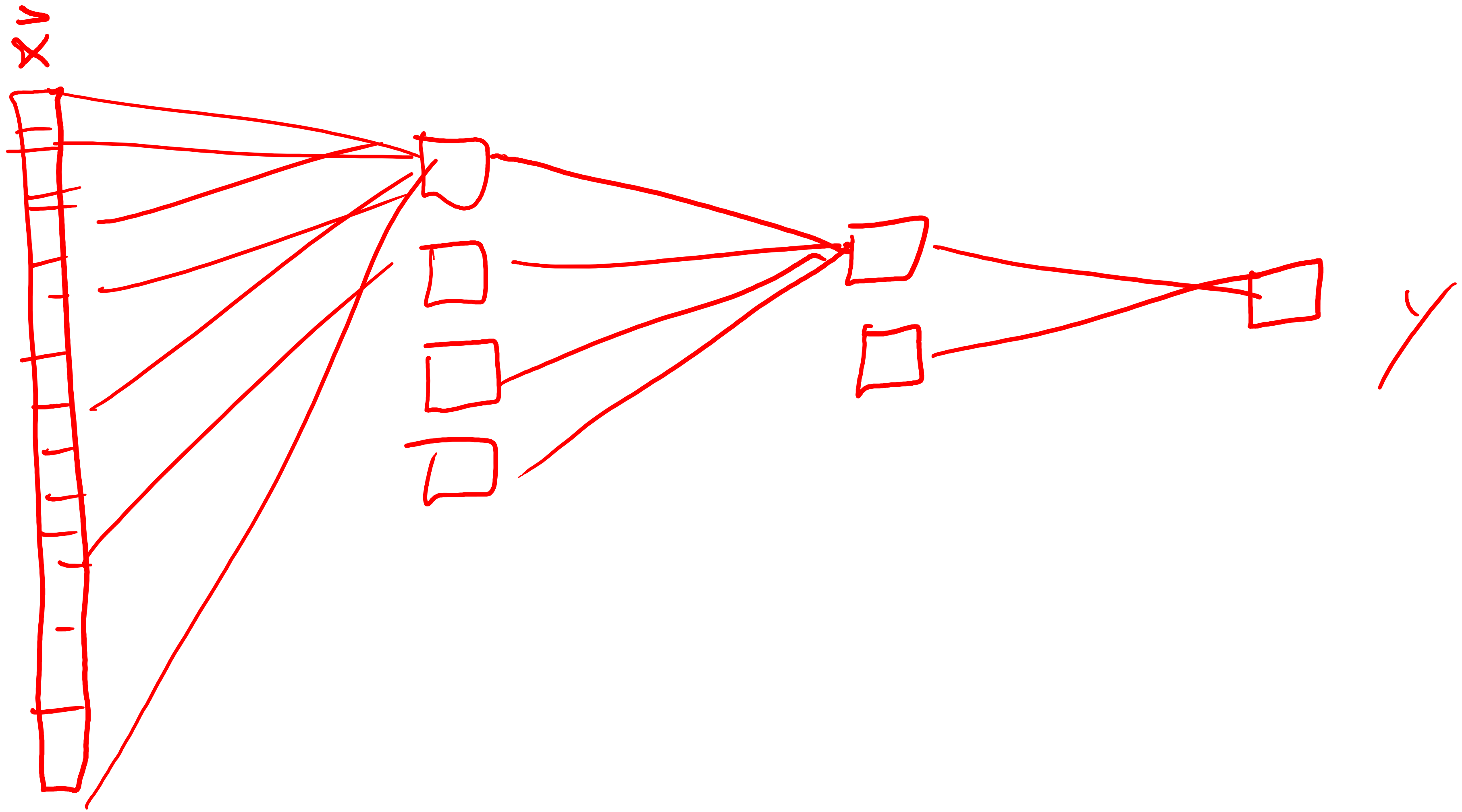




Neural Networks from HW2

Digit Classification

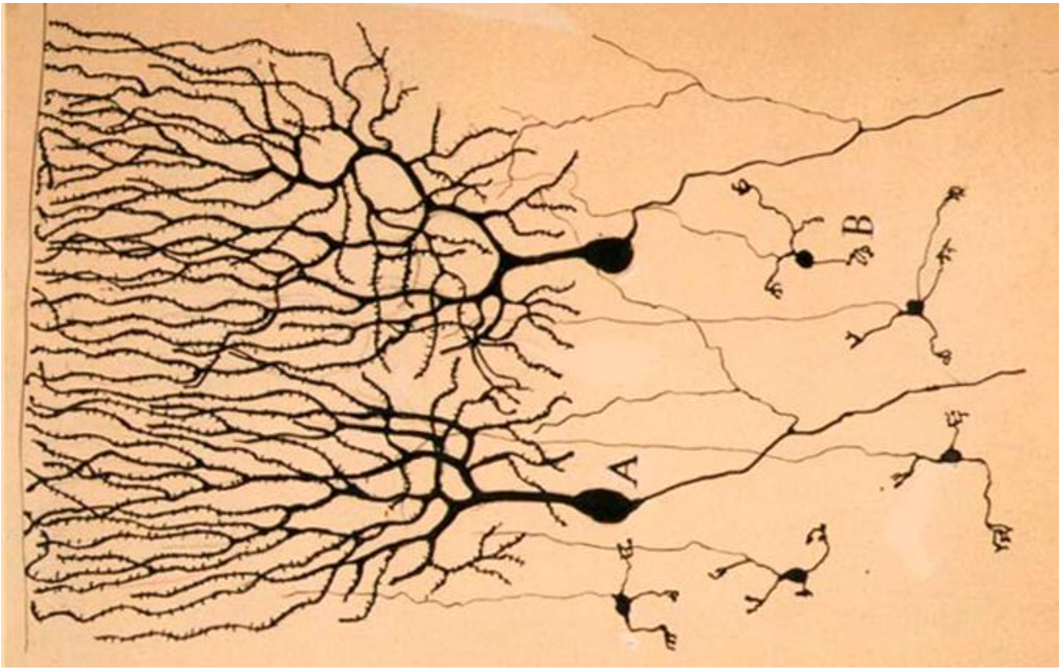




Neural Networks

Inspired by actual human brain

Input Signal



Output Signal

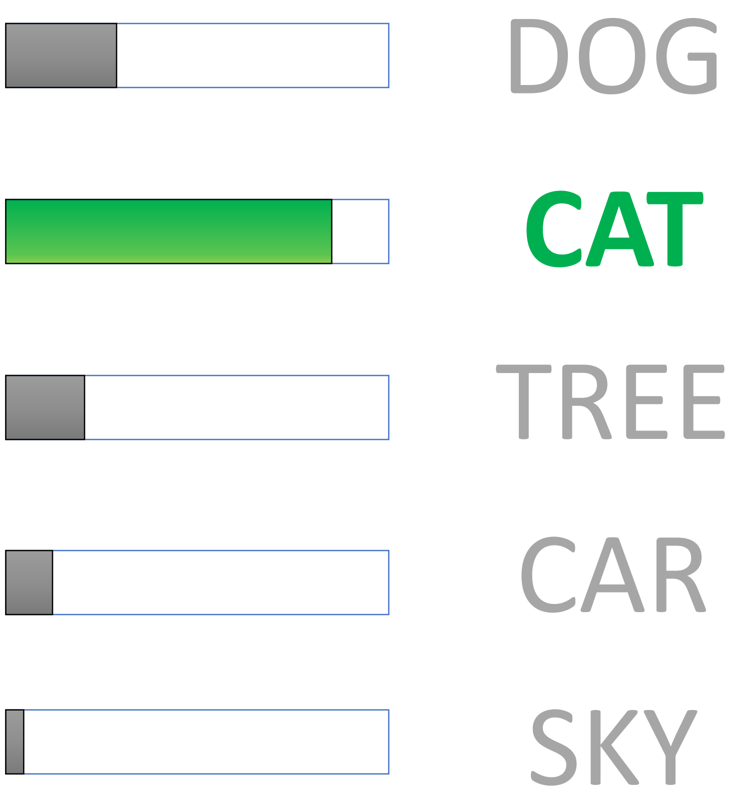


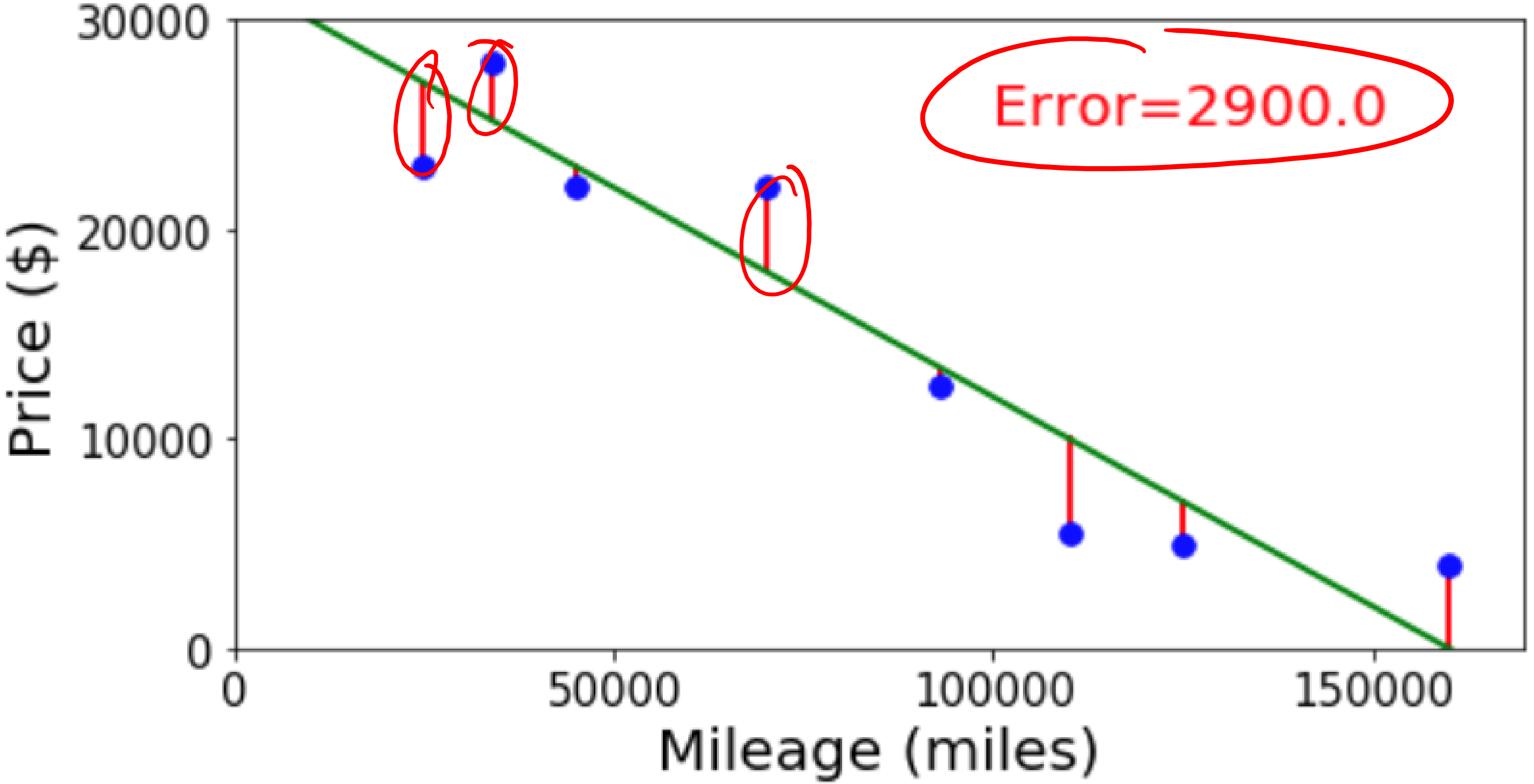
Image: https://en.wikipedia.org/wiki/Neuron

Neural Networks

Simple single neuron example:

- Selling my car

$\begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$

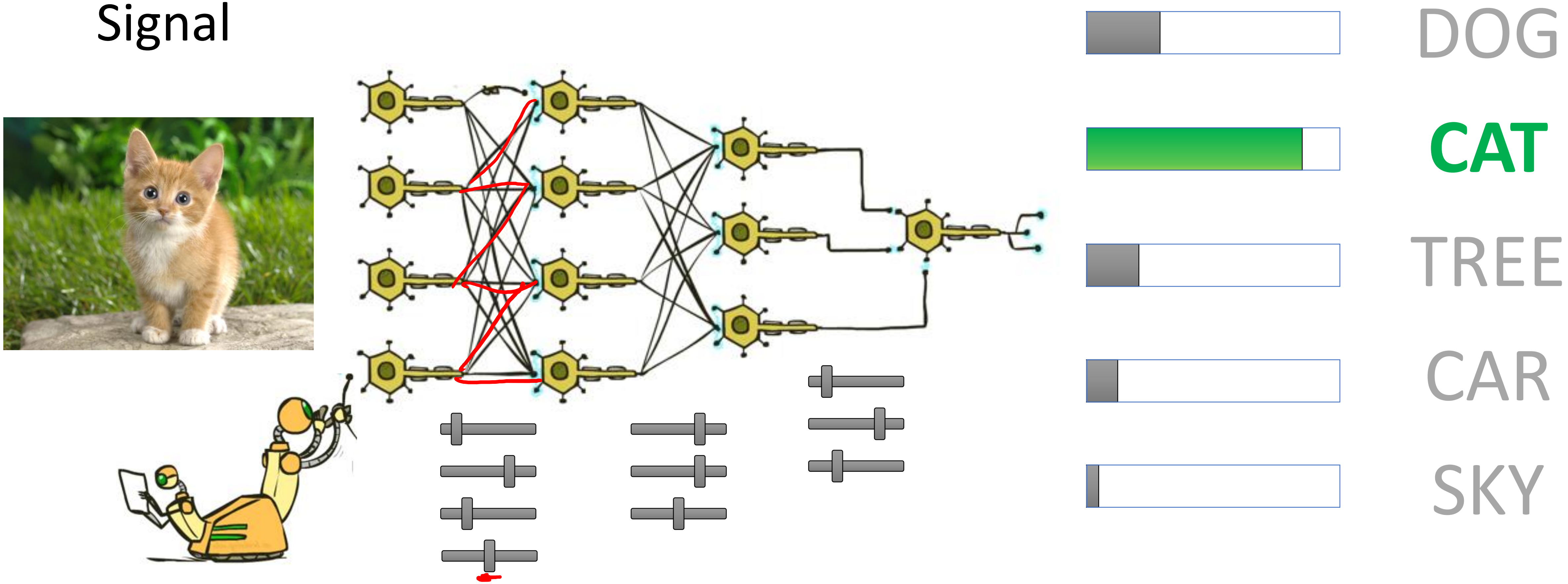


Neural Networks

Many layers of neurons, millions of parameters

Output Signal

Input Signal

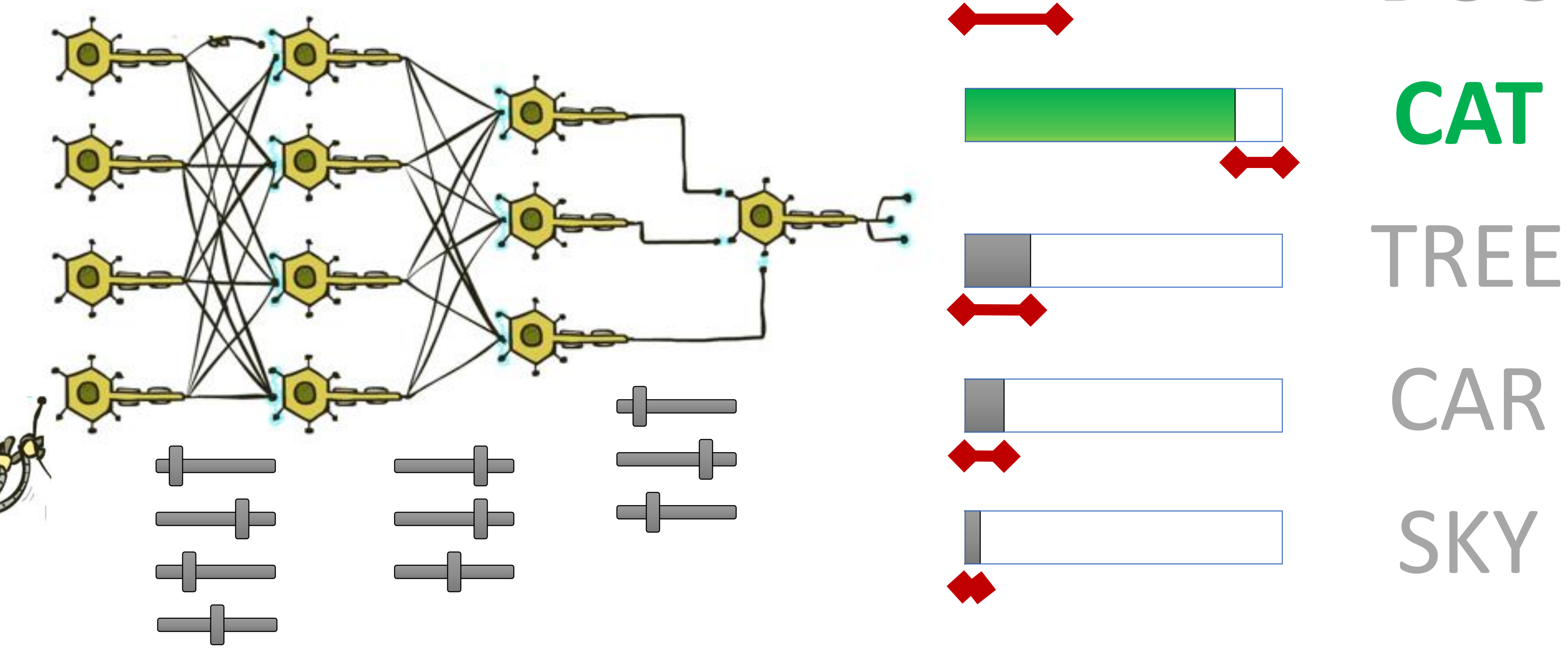


Neural Networks

Many layers of neurons, millions of parameters

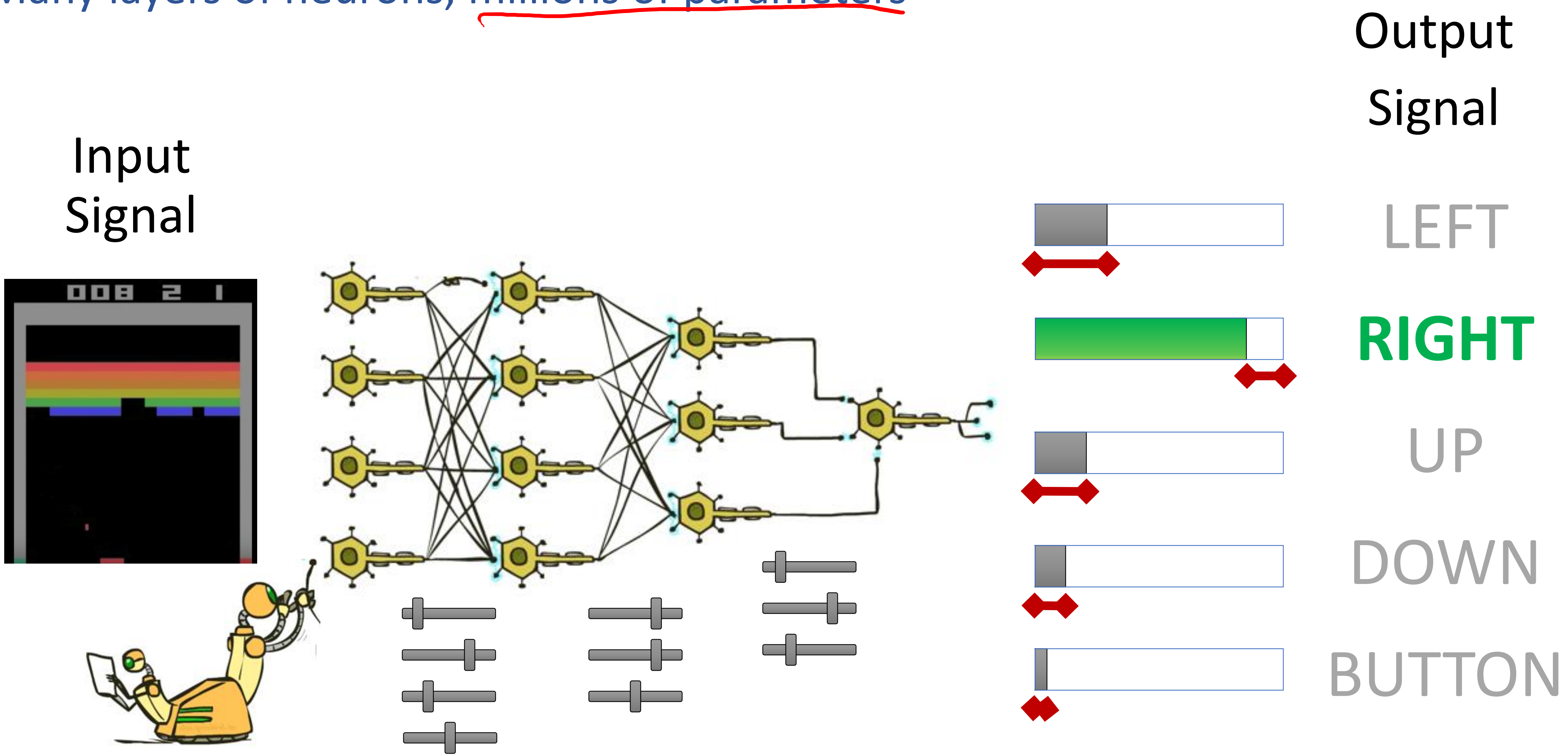
Output Signal

Input Signal

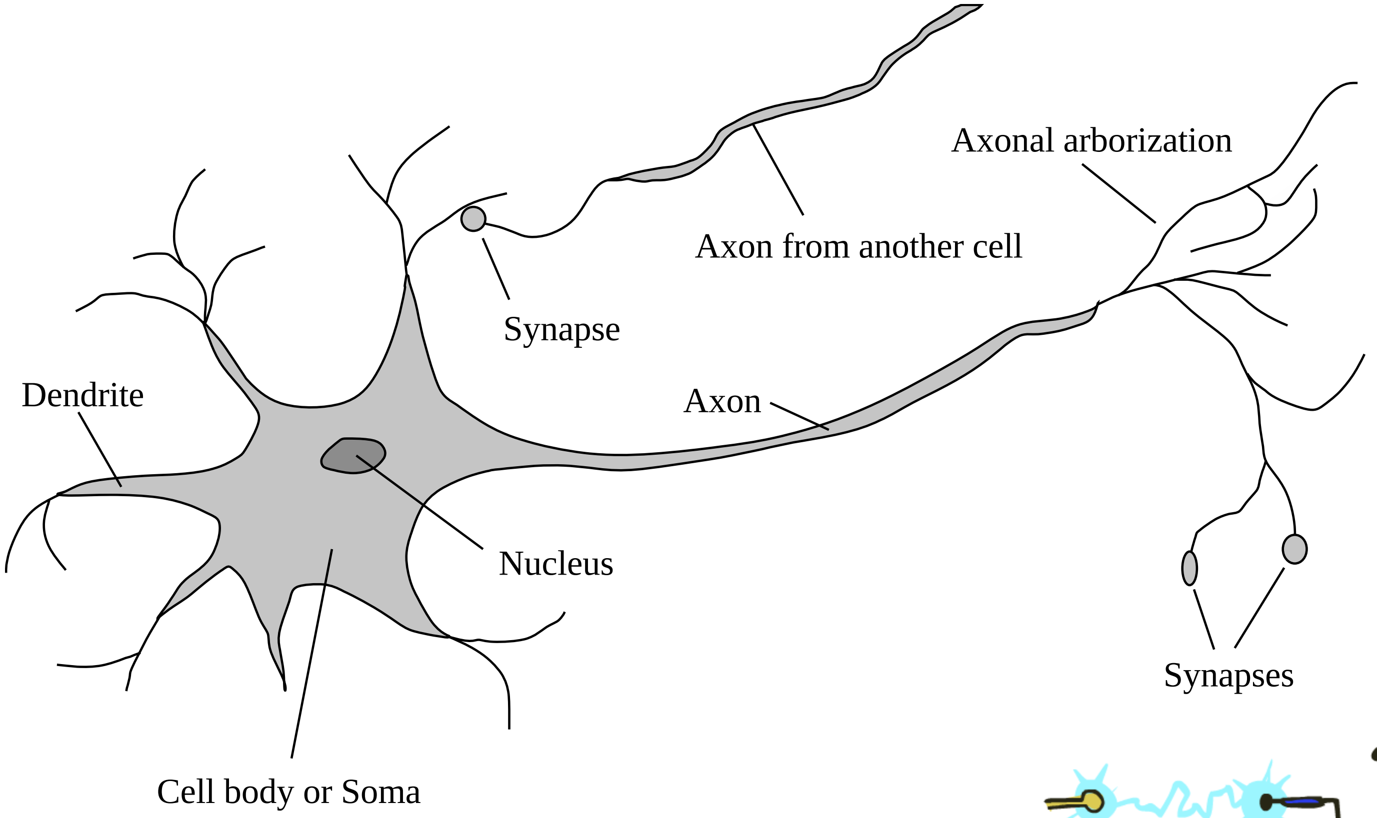


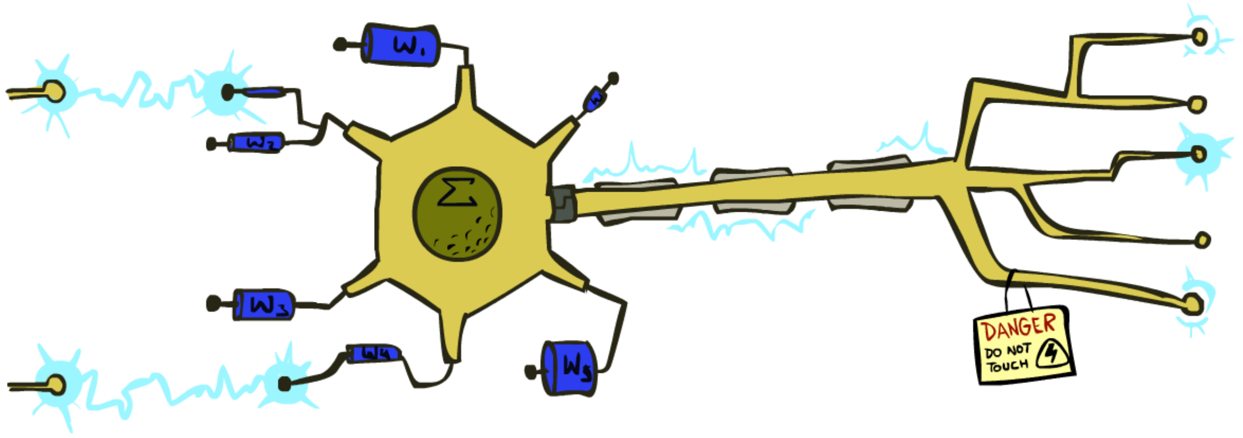
Neural Networks

Many layers of neurons, millions of parameters



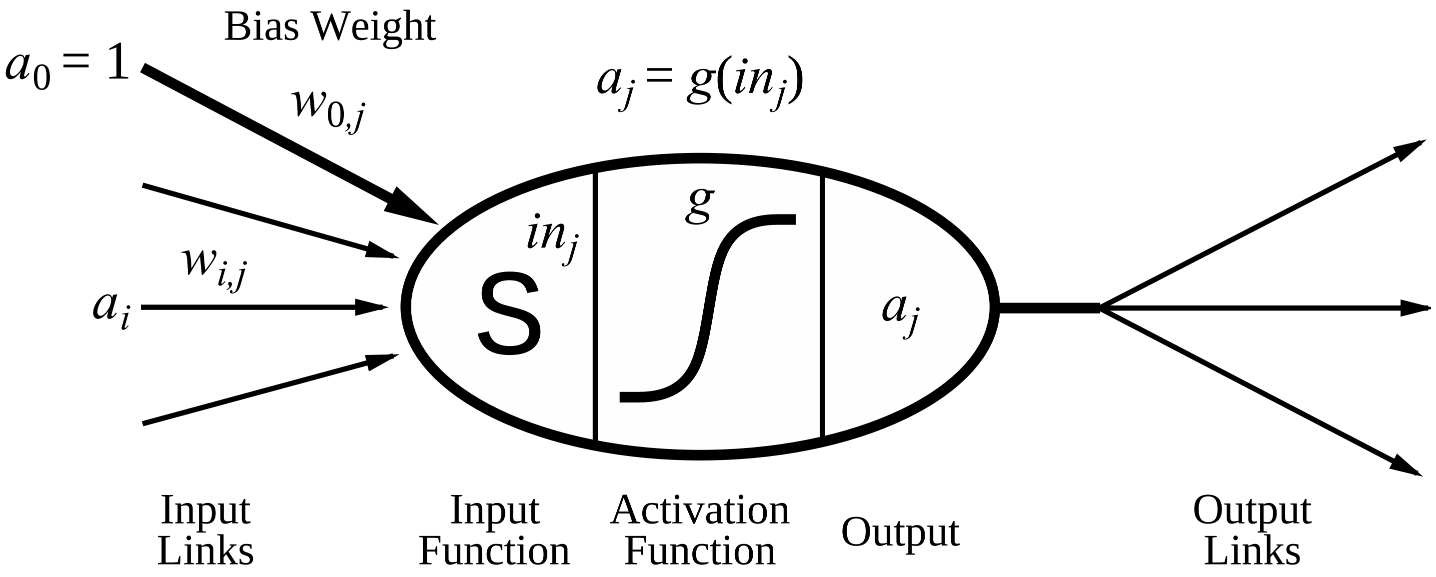
Very Loose Inspiration: Human Neurons





Slide from UC Berkeley AI

Simple Model of a Neuron (McCulloch & Pitts, 1943)



Inputs a_i come from the output of node i to this node j (or from "outside")

Each input link has a ***weight*** $w_{i,j}$

There is an additional fixed input a_0 with ***bias*** weight $w_{0,j}$

The total input is $in_j = \Sigma_i \, w_{i,j} \, a_i$

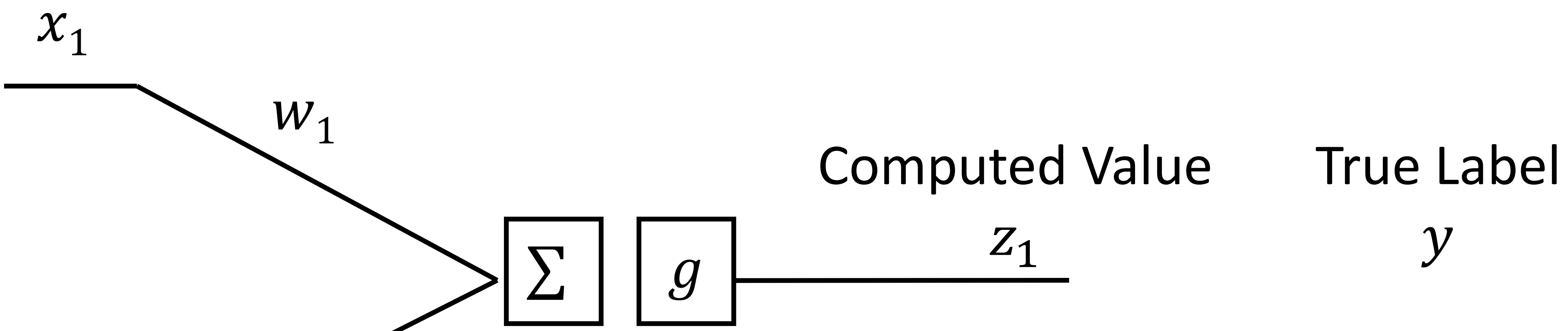
The output is $a_j = g(in_j) = g(\Sigma_i \, w_{i,j} \, a_i) = g(\mathbf{w.a})$

Slide from UC Berkeley AI

Single Neuron

Single neuron system

- Perceptron (if g is step function)
- Logistic regression (if g is sigmoid)



$$h_{\boldsymbol{w}}(\boldsymbol{x}) = z_1$$

$$h_{\boldsymbol{w}}(\boldsymbol{x}) = g\left(\sum_i w_i x_i\right)$$

Optimizing

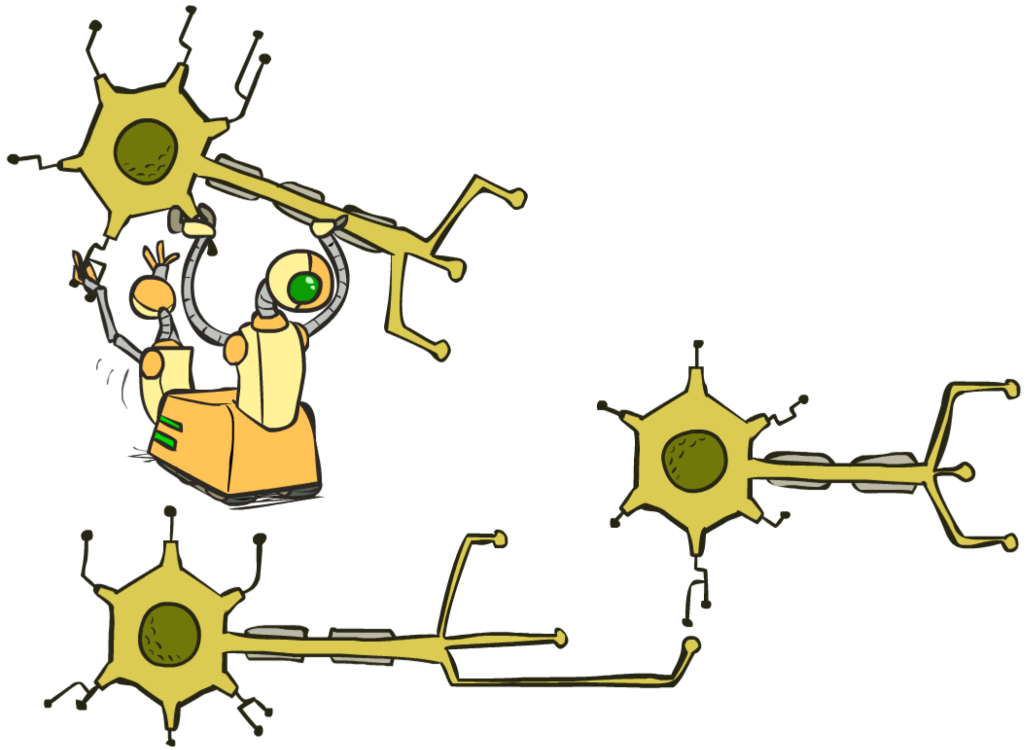
How do we find the "best" set of weights?

$$h_{\boldsymbol{w}}(\boldsymbol{x}) = g\left(\sum_i w_i x_i\right)$$

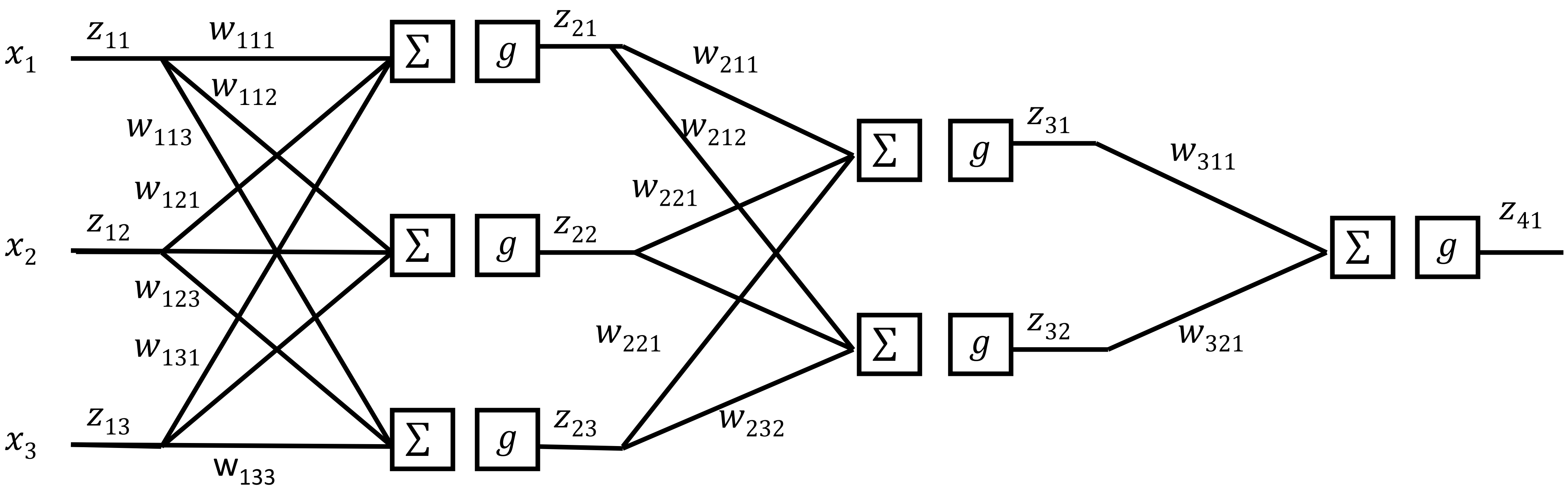
Multilayer Perceptrons

A ***multilayer perceptron*** is a feedforward neural network with at least one ***hidden layer*** (nodes that are neither inputs nor outputs)

MLPs with enough hidden nodes can represent any function



Neural Network Equations



$$h_w(\boldsymbol{x}) = z_{4,1}$$

$$z_{1,1} = x_1$$

$$z_{4,1} = g\left(\sum_i w_{3,i,1}\, z_{3,i}\right)$$

$$z_{3,1} = g\left(\sum_i w_{2,i,1}\, z_{2,i}\right)$$

$$z_{d,1} = g\left(\sum_i w_{d-1,i,1}\, z_{d-1,i}\right)$$

$$h_w(x) = g\left(\sum_k w_{3,k,1}\; g\left(\sum_j w_{2,j,k}\; g\left(\sum_i w_{1,i,j}\; x_i\right)\right)\right)$$

Optimizing

How do we find the "best" set of weights?

$$h_w(x) = g\left(\sum_k w_{3,k,1}\ g\left(\sum_j w_{2,j,k}\ g\left(\sum_i w_{1,i,j}\ x_i\right)\right)\right)$$

Neural Networks Properties

Practical considerations

- Large number of neurons
  - Danger for overfitting
- Modelling assumptions vs data assumptions trade-off
- Gradient descent can get stuck in bad local optima

What if there are no non-linear activations?

- A deep neural network with only linear layers can be reduced to an exactly equivalent single linear layer

Universal Approximation Theorem:

- A two-layer neural network with a sufficient number of neurons can approximate any continuous function to any desired accuracy.