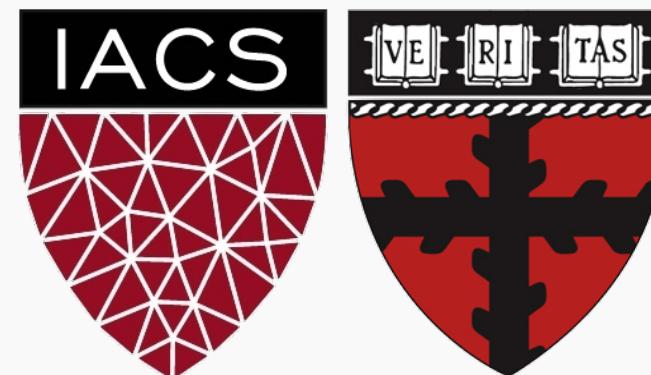


# Lecture 18: Perceptron and Multilayer Perceptron

CS109A Introduction to Data Science  
Pavlos Protopapas, Kevin Rader and Chris Tanner



# ANNOUNCEMENTS

- Homework 5 (209) due on Wednesday 11:59 pm, Nov 6
- Advanced Section on Trees is on Wednesday Nov 13
- Finally



# Outline

---

1. Introduction to Artificial Neural Networks
2. Review of Classification and Logistic Regression
3. Single Neuron Network ('Perceptron')
4. Multi-Layer Perceptron (MLP)

# Outline

---

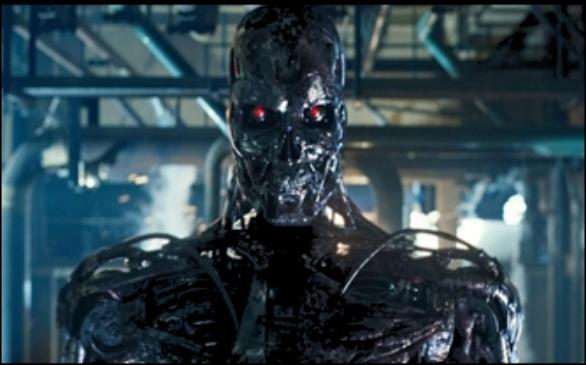
- 1. Introduction to Artificial Neural Networks**
- 2. Review of Classification and Logistic Regression**
- 3. Single Neuron Network ('Perceptron')**
- 4. Multi-Layer Perceptron (MLP)**



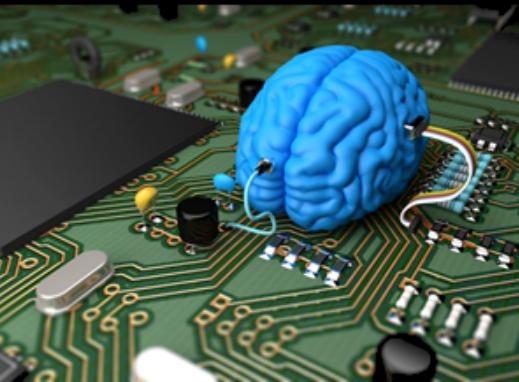
# Artificial Neural Networks



# Deep Learning



What society thinks I do



What my friends think I do



What other computer scientists think I do



What mathematicians think I do



What I think I do

```
In [1]:  
import keras  
Using TensorFlow backend.
```

What I actually do

# Watch this!

---

<http://video.arsTechnica.com/watch/sunspring-sci-fi-short-film>

dialogue written by neural net



# Today's news

## An AI just beat top lawyers at their own game

[Share on F](#) [Share on T](#) [+](#)



WHAT'S THIS?

IMAGE: BOB AL-GREEN/MASHABLE



BY  
**MONICA  
CHIN**

FEB  
26  
2018

The nation's top lawyers recently battled artificial intelligence in a competition to interpret contracts — and they lost.

A new study, conducted by legal AI platform LawGeex in consultation with scholars from Stanford University, Duke University School of Law, and University of Southern California, pitted twenty experienced lawyers against an AI trained to evaluate legal contracts.

Competitors were given four hours to review five non-disclosure agreements (NDAs) and identify 30 legal issues, including arbitration, confidentiality of relationship, and indemnification. They were scored by how accurately they identified each issue.

**SEE ALSO:** [Google's new AI can predict heart disease by simply scanning your eyes](#)

# Today's news

## Google's new AI can predict heart disease by simply scanning your eyes

[Share on Facebook](#) [Share on Twitter](#) [Share on LinkedIn](#) [More](#)



IMAGE: BEN BRAIN/DIGITAL CAMERA MAGAZINE  
VIA GETTY IMAGES

The secret to identifying certain health conditions may be hidden in our eyes.

BY

MONICA

CHIN

FEB  
2018

Researchers from Google and its health-tech subsidiary Verily announced on Monday that they have successfully created algorithms to predict whether someone has high blood pressure or is at risk of a heart attack or stroke simply by scanning a person's eyes, the *Washington Post* reports.

SEE ALSO: [This fork helps you stay healthy](#)

Google's researchers trained the algorithm with images of scanned retinas from more than 280,000 patients. By reviewing this massive database, Google's algorithm trained itself to recognize the patterns that designated people as at-risk.

This algorithm's success is a sign of exciting developments in healthcare on the horizon. As Google fine-tunes the technology, it could one day

# AlphaGo (2015)

First program to beat a professional Go player



# AlphaZero (2017)

DeepMind

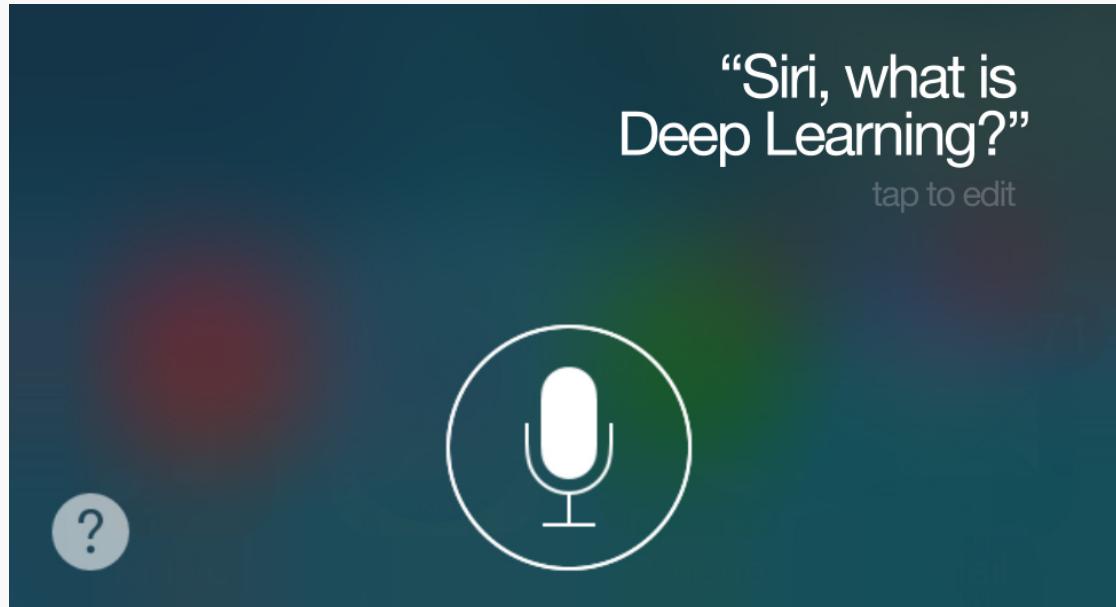
## AlphaZero AI beats champion chess program after teaching itself in four hours

Google's artificial intelligence sibling DeepMind repurposes Go-playing AI to conquer chess and shogi without aid of human knowledge



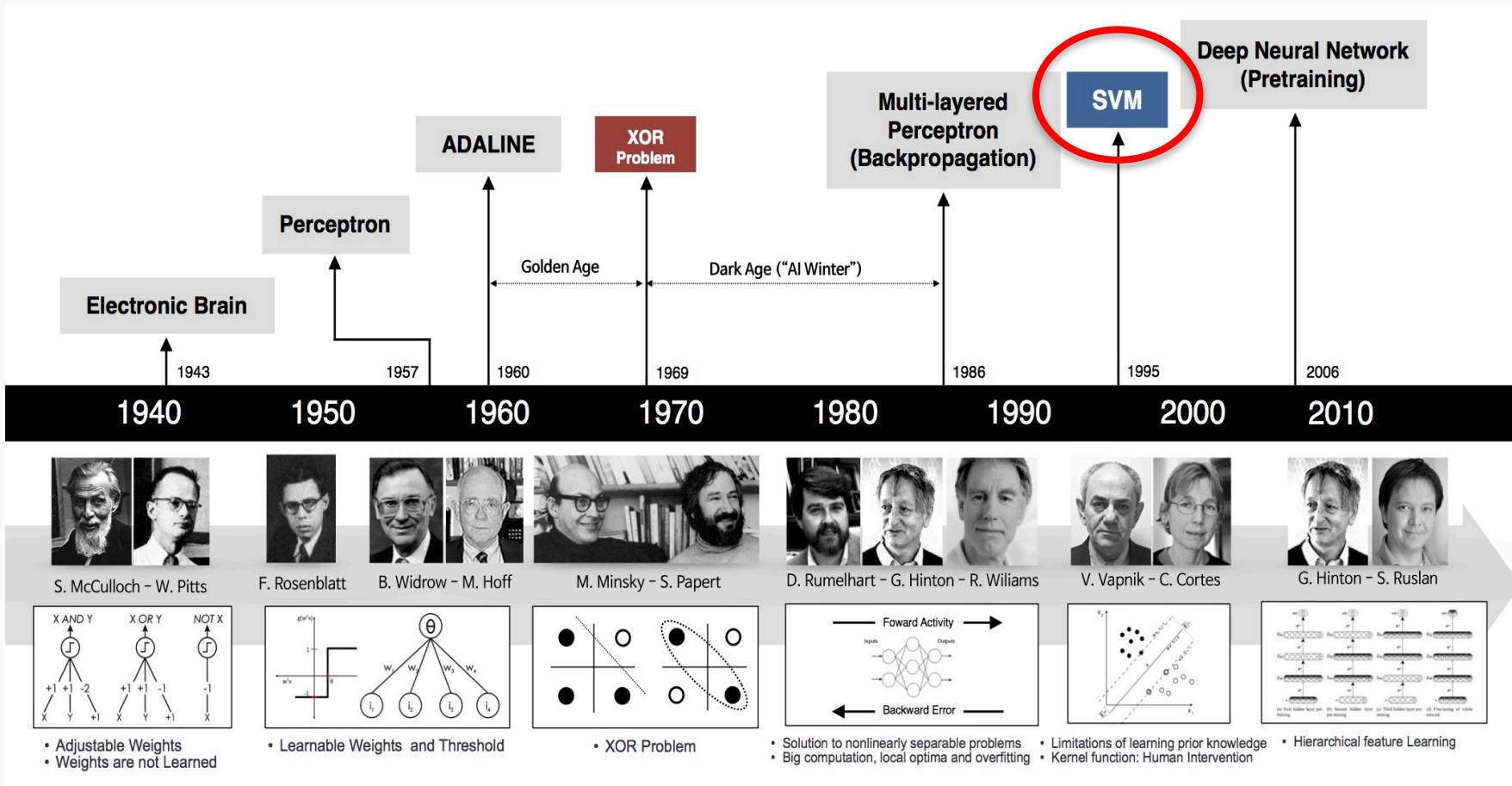
# iOS Speech Synthesis (2016-)

Trained from 20 hours of high quality speech



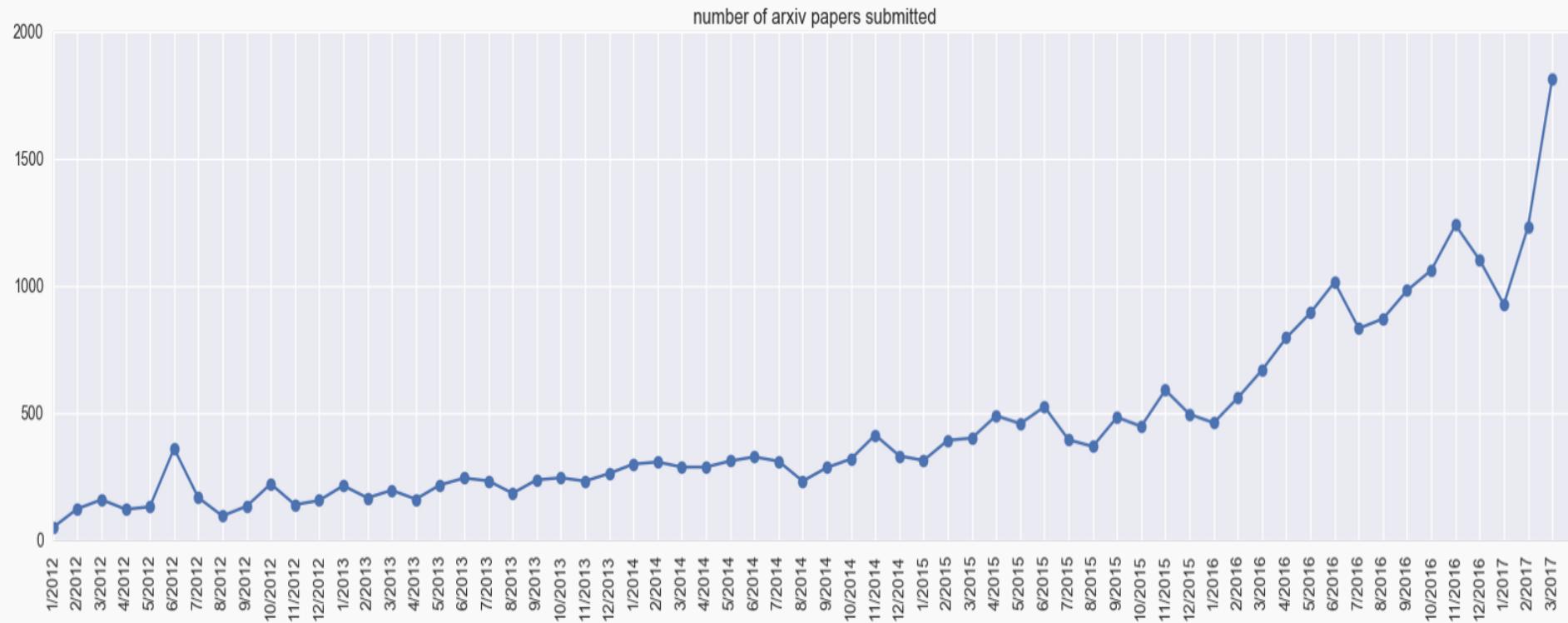
[machinelearning.apple.com](https://machinelearning.apple.com)

# Historical Trends



# Historical Trends

## ArXiv papers on deep learning: 2012-2017



# Outline

---

1. Introduction to Artificial Neural Networks
2. **Review of Classification and Logistic Regression**
3. Single Neuron Network ('Perceptron')
4. Multi-Layer Perceptron (MLP)

# Classification and Logistic Regression



# Regression and Classification

---

Methods that are centered around modeling and prediction of a **quantitative** response variable (ex, number of taxi pickups, number of bike rentals, etc) are called **regressions** (and Ridge, LASSO, etc).

When the response variable is **categorical**, then the problem is no longer called a regression problem but is instead labeled as a **classification problem**.

The goal is to attempt to classify each observation into a category (aka, class or cluster) defined by Y, based on a set of predictor variables X.

# Typical Classification Examples

---

The motivating examples for this lecture(s), homeworks and labs are based on classification. Classification problems are common in these domains:

- Trying to determine where to set the cut-off for some diagnostic test (pregnancy tests, prostate or breast cancer screening tests, etc...)
- Trying to determine if cancer has gone into remission based on treatment and various other indicators
- Trying to classify patients into types or classes of disease based on various genomic markers



# Data: Response vs. Predictor Variables

The diagram illustrates a data matrix with two main components: predictors (X) and observations (n). The predictors are represented by three columns: TV, radio, and newspaper. The observations are represented by five rows of data points. The last column, sales, represents the response variable. Brackets on the left indicate the number of observations (n), and brackets at the bottom indicate the number of predictors (p).

	TV	radio	newspaper	sales
230.1	37.8	69.2	22.1	
44.5	39.3	45.1	10.4	
17.2	45.9	69.3	9.3	
151.5	41.3	58.5	18.5	
180.8	10.8	58.4	12.9	

**X**  
**predictors**  
features  
covariates

**Y**  
outcome  
**response** variable  
dependent variable

n observations

p predictors

# Response vs. Predictor Variables

$$X = X_1, \dots, X_p$$

$$X_j = x_{1j}, \dots, x_{ij}, \dots, x_{nj}$$

**predictors**

features

covariates

$$Y = y_1, \dots, y_n$$

outcome

**response** variable

dependent variable

**n** observations

TV	radio	newspaper	sales
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	9.3
151.5	41.3	58.5	18.5
180.8	10.8	58.4	12.9

**p** predictors



# Heart Data

response variable Y  
is Yes/No

Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak	Slope	Ca	Thal	AHD
63	1	typical	145	233	1	2	150	0	2.3	3	0.0	fixed	No
67	1	asymptomatic	160	286	0	2	108	1	1.5	2	3.0	normal	Yes
67	1	asymptomatic	120	229	0	2	129	1	2.6	2	2.0	reversible	Yes
37	1	nonanginal	130	250	0	0	187	0	3.5	3	0.0	normal	No
41	0	nontypical	130	204	0	2	172	0	1.4	1	0.0	normal	No



# Heart Data

---

These data contain a binary outcome HD for 303 patients who presented with chest pain. An outcome value of:

- **Yes** indicates the presence of heart disease based on an angiographic test,
- **No** means no heart disease.

There are 13 predictors including:

- Age
- Sex
- Chol (a cholesterol measurement),
- MaxHR
- RestBP

and other heart and lung function measurements.



# Logistic Regression

---

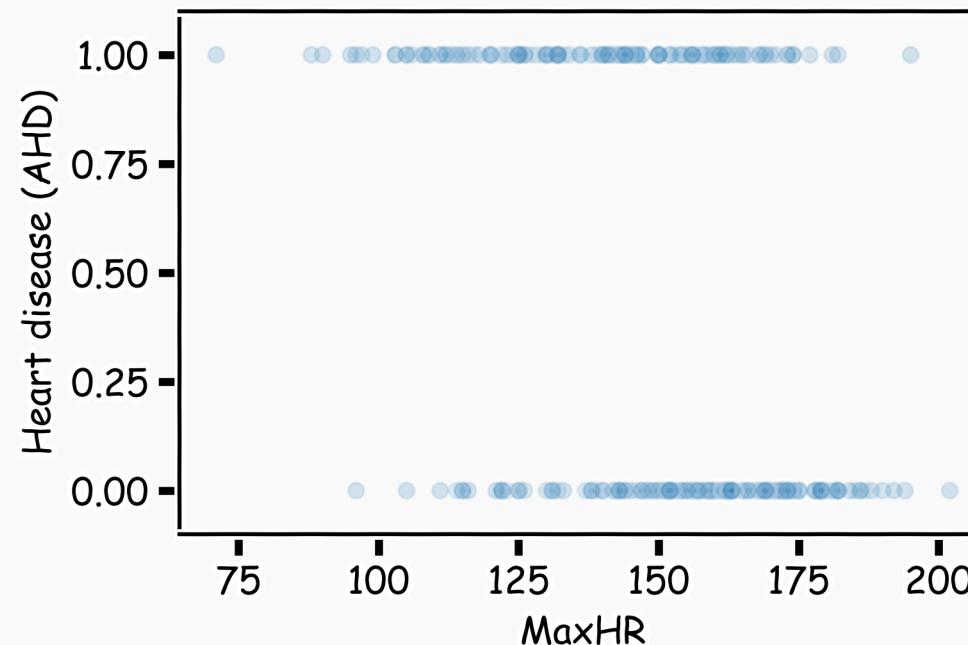
Logistic Regression addresses the problem of estimating a probability,  $P(y = 1)$ , given an input  $X$ . The logistic regression model uses a function, called the **logistic** function, to model  $P(y = 1)$ :

$$P(Y = 1) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}} = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$



# Heart Data: logistic estimation

We'd like to predict whether or not a person has a heart disease. And we'd like to make this prediction, for now, just based on the MaxHR.



# Logistic Regression

---

As a result the model will predict  $P(y = 1)$  with an *S*-shaped curve, which is the general shape of the logistic function.

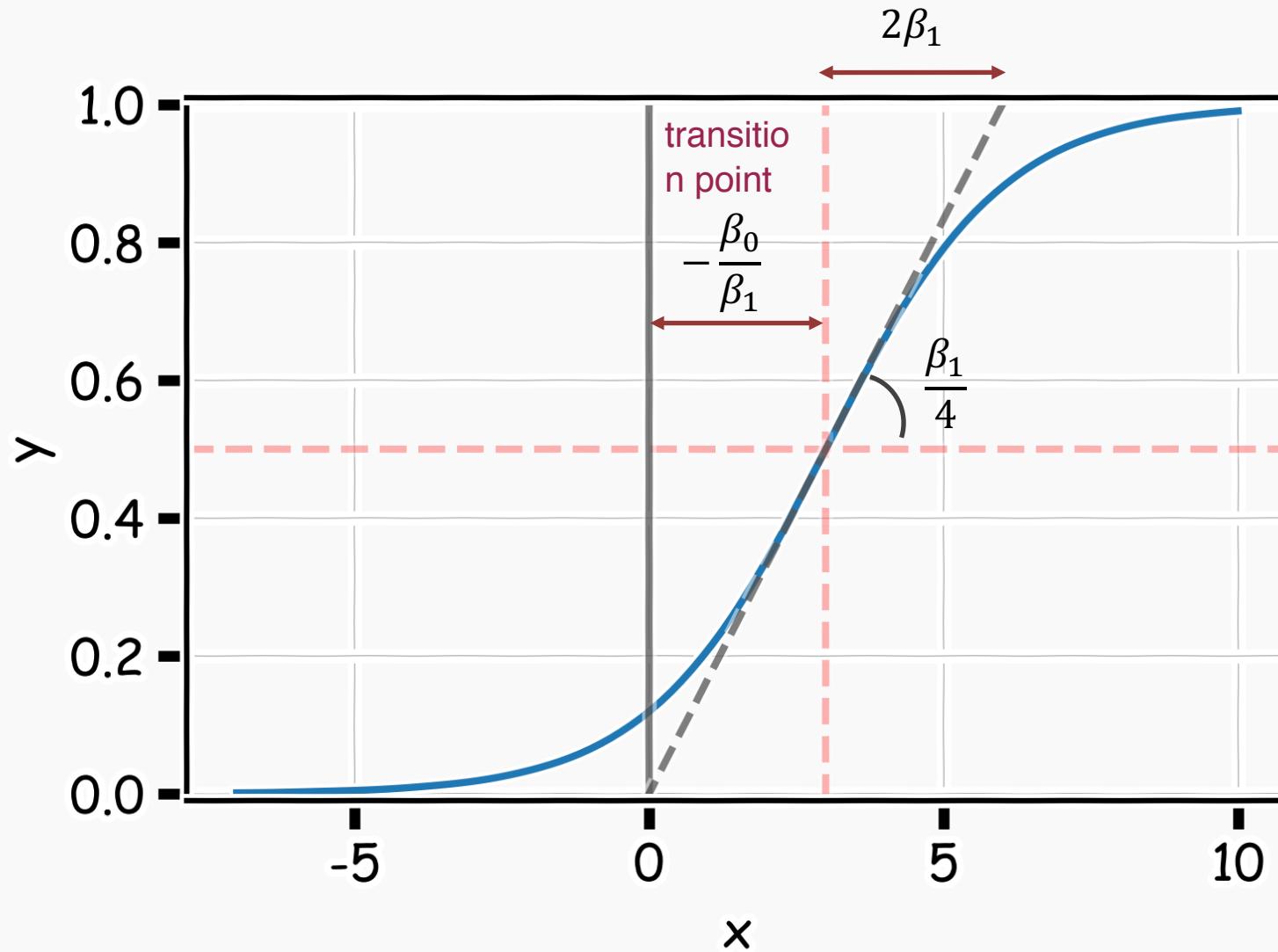
$\beta_0$  shifts the curve right or left by  $c = -\frac{\beta_0}{\beta_1}$ .

$\beta_1$  controls how steep the *S*-shaped curve is distance from  $\frac{1}{2}$  to  $\sim 1$  or  $\frac{1}{2}$  to  $\sim 0$  to  $\frac{1}{2}$  is  $\frac{2}{\beta_1}$

Note: if  $\beta_1$  is positive, then the predicted  $P(y = 1)$  goes from zero for small values of  $X$  to one for large values of  $X$  and if  $\beta_1$  is negative, then has the  $P(y = 1)$  opposite association.

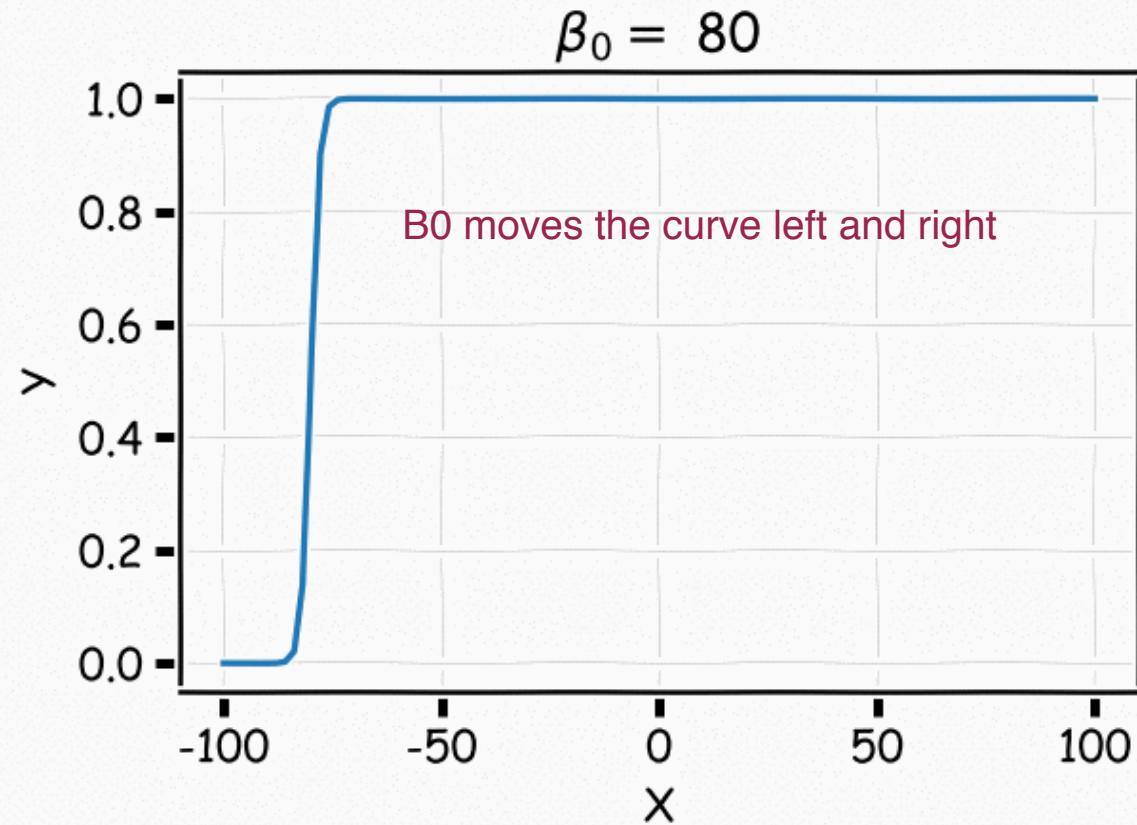


# Logistic Regression



# Logistic Regression

$$P(Y = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$

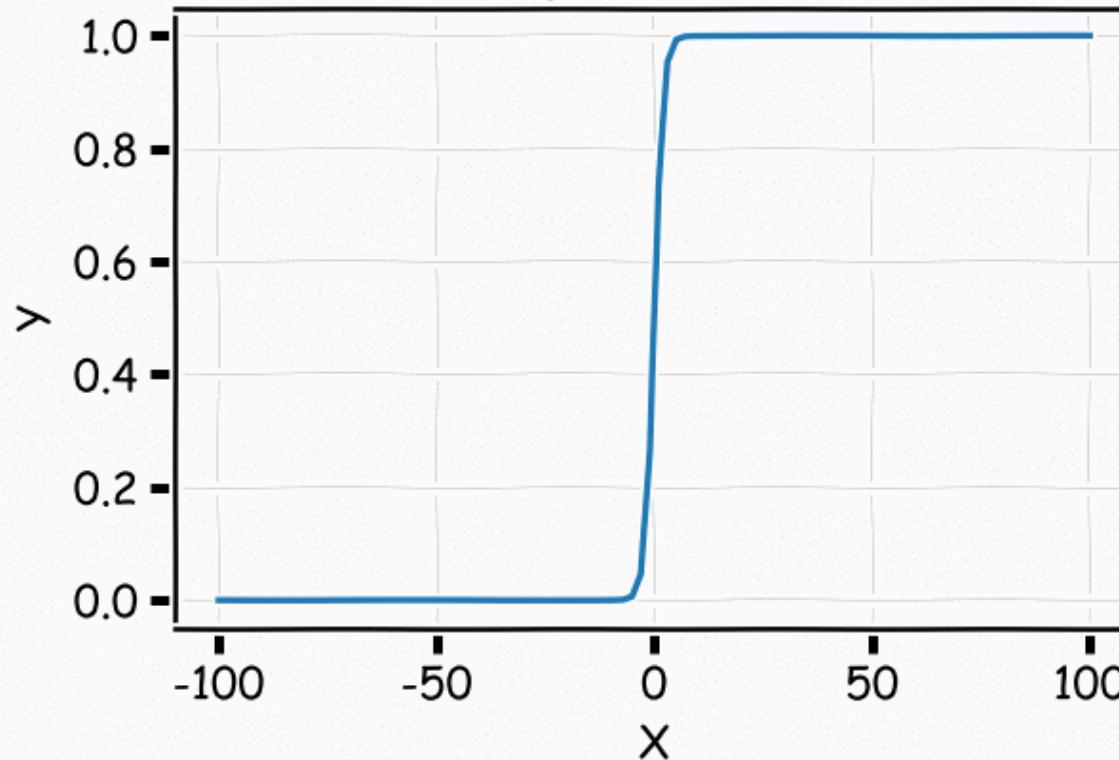


# Logistic Regression

$$P(Y = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$

B1 changes slope (steepness and direction)

$$\beta_1 = 1.0$$

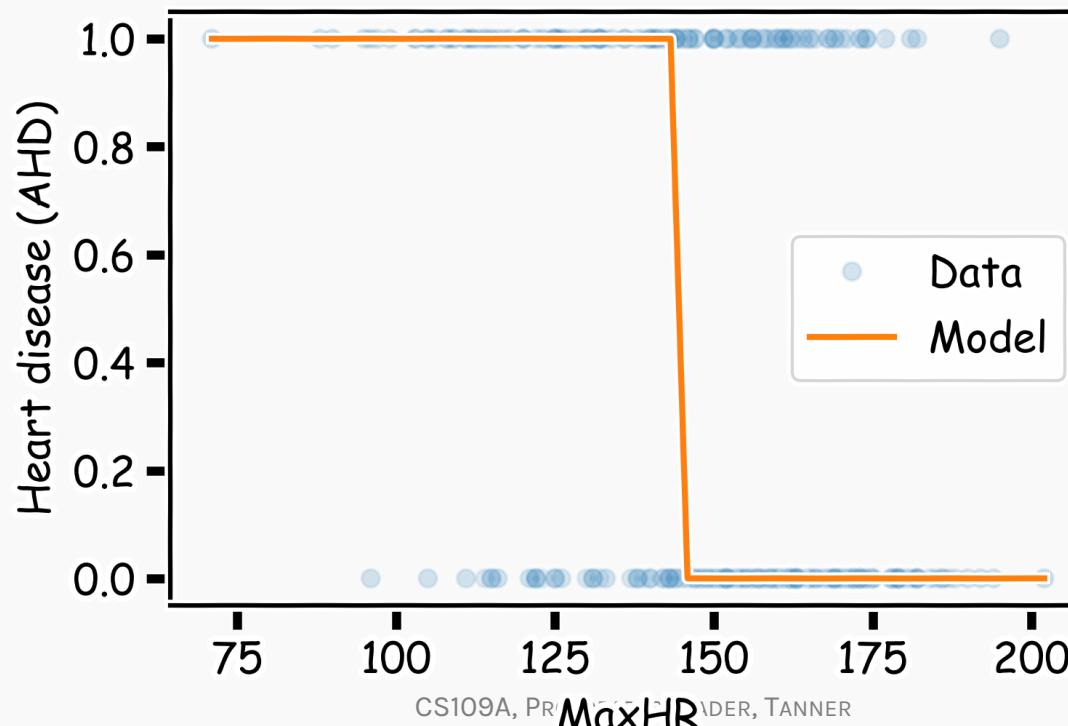


# Estimating the coefficients for Logistic Regression

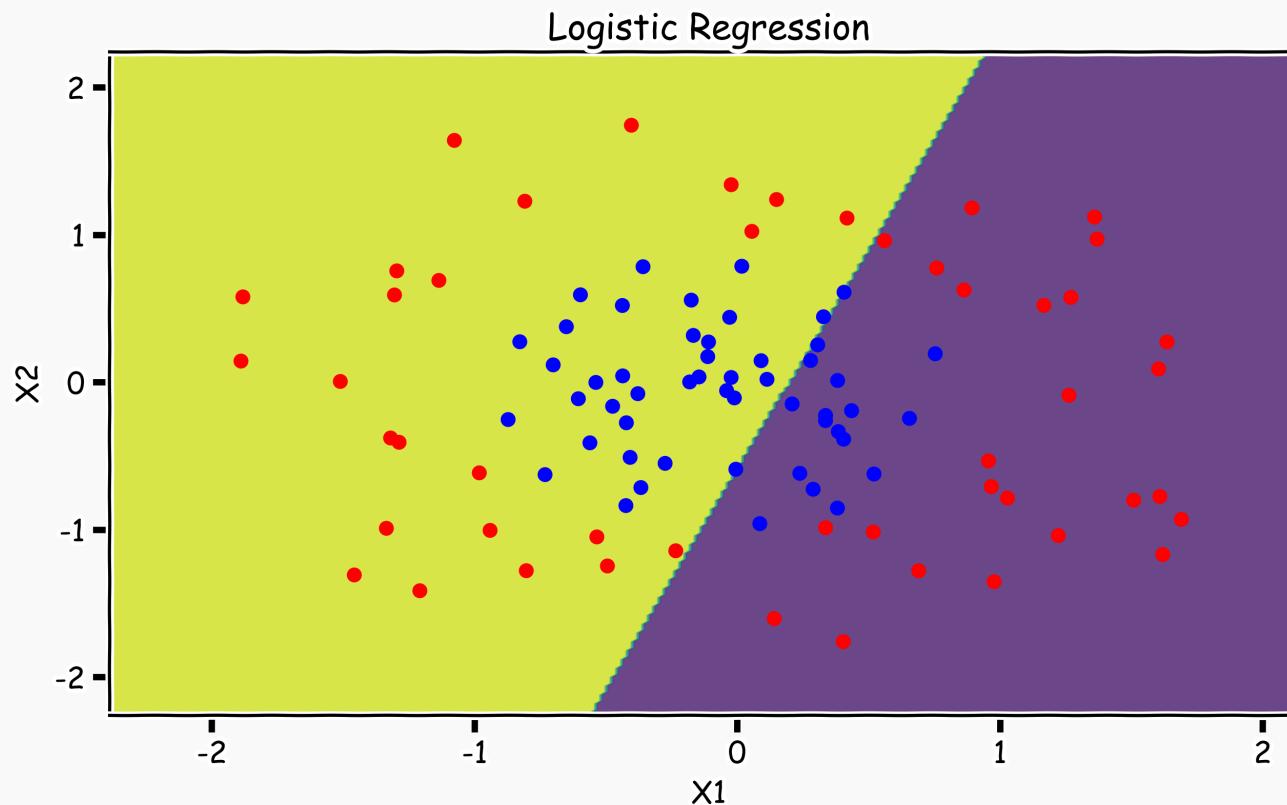
Find the coefficients that minimize the loss function

$$\mathcal{L}(\beta_0, \beta_1) = - \sum_i [y_i \log p_i + (1 - y_i) \log(1 - p_i)]$$

cross entropy  
loss fxn

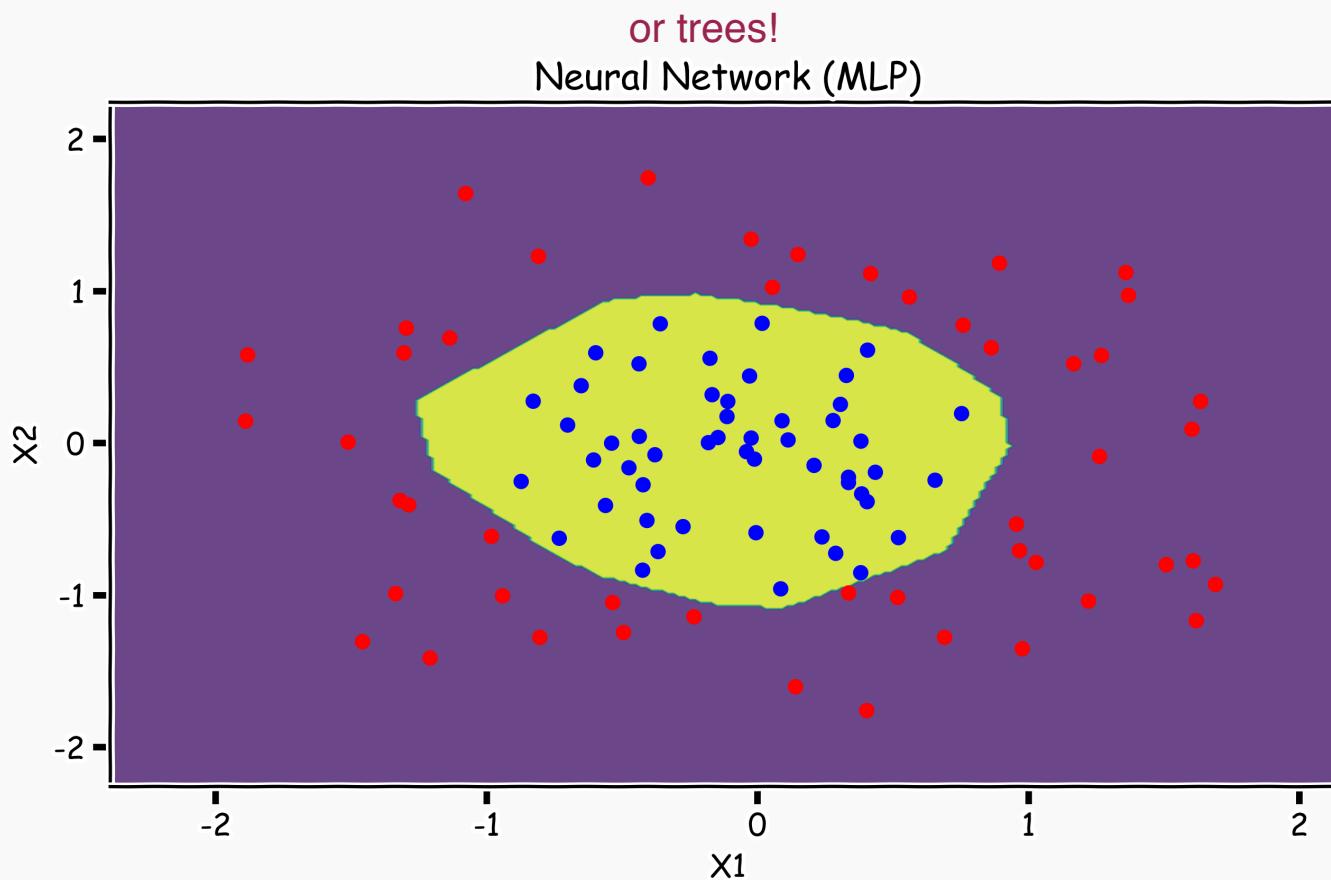


# Need for Non-Linearity



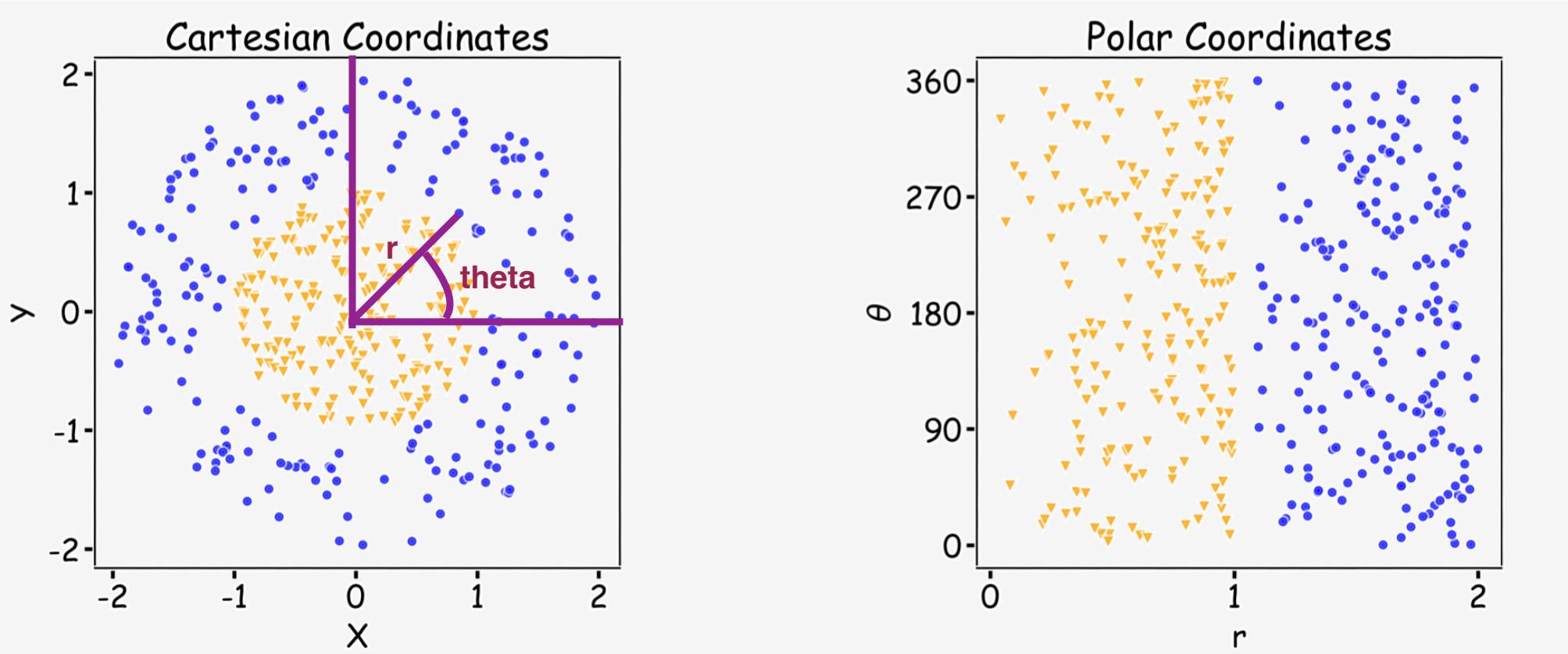
Without **augmenting the features** (i.e. without adding  $X_1^2$  or  $X_2^2$  non-linear features), **Logistic Regression** is incapable of modeling the correct decision boundary.

# Neural Networks to The Rescue



A neural network is a powerful non-linear model that can easily model the non-linear decision boundary correctly.

# Representation Matters



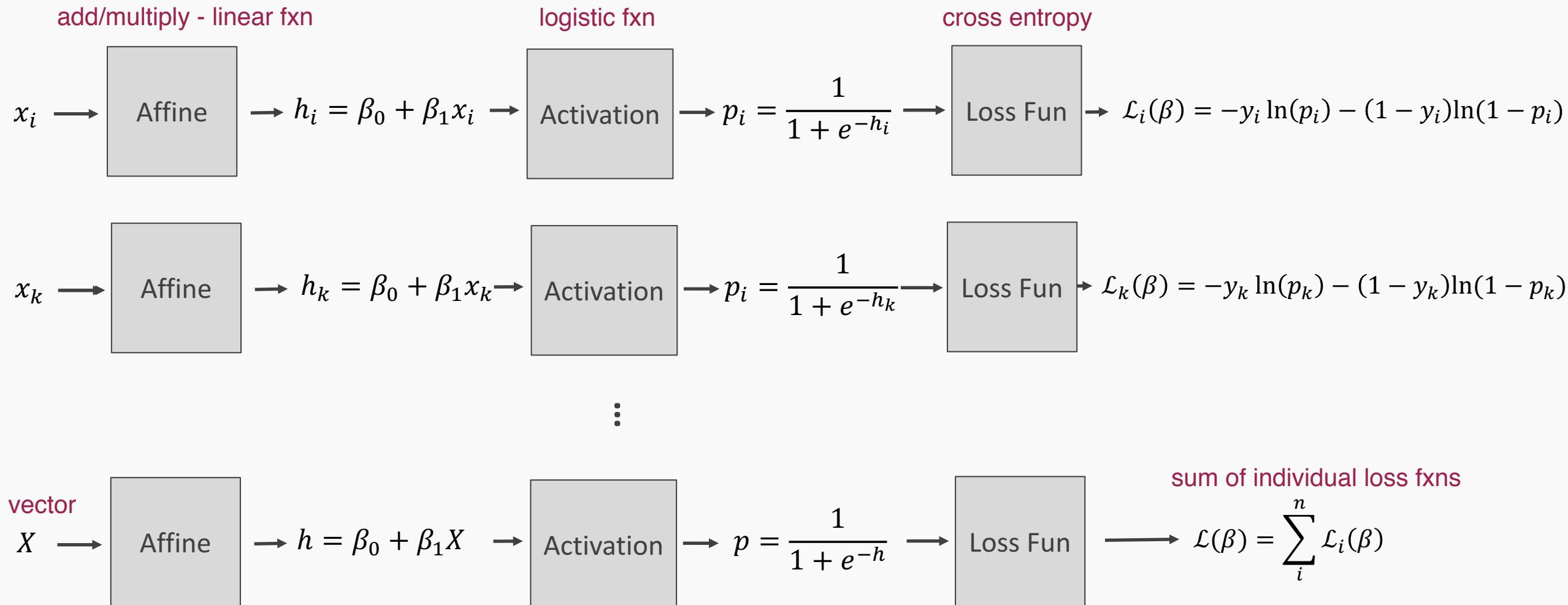
Neural networks can **learn useful representations** for the problem. This is another reason why they can be so powerful!

# Outline

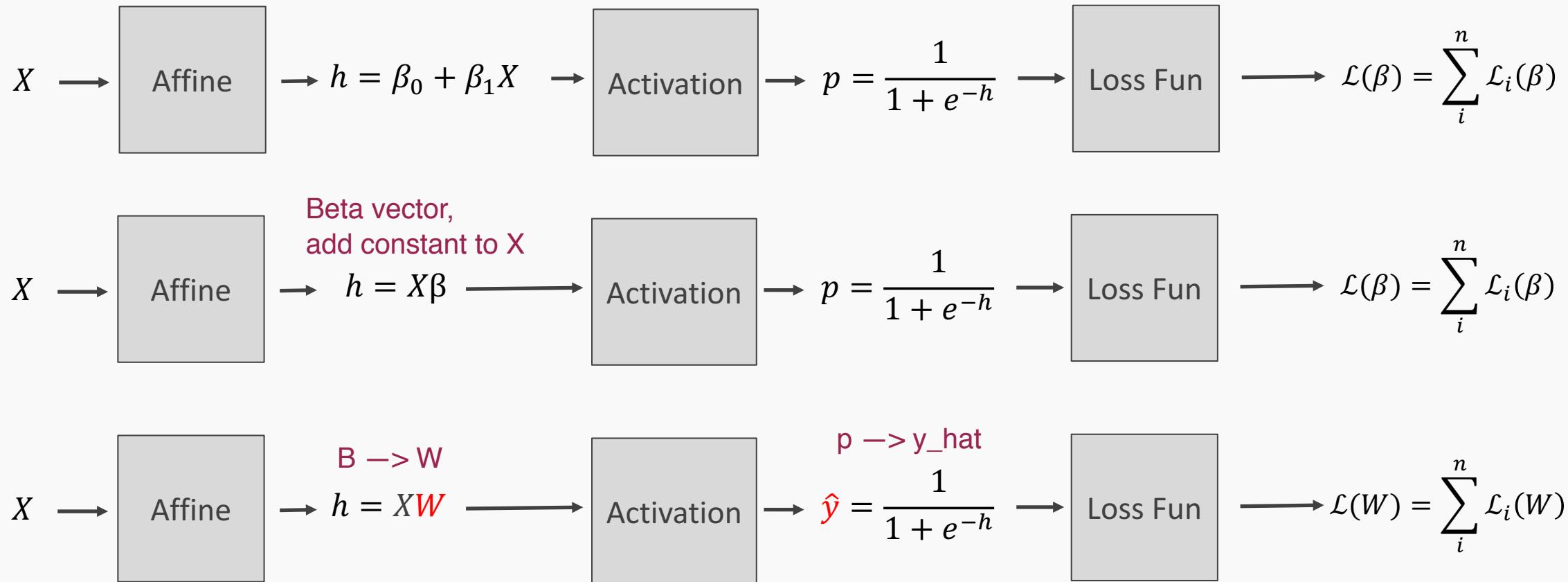
---

1. Introduction to Artificial Neural Networks
2. Review of Classification and Logistic Regression
- 3. Single Neuron Network ('Perceptron')**
4. Multi-Layer Perceptron (MLP)

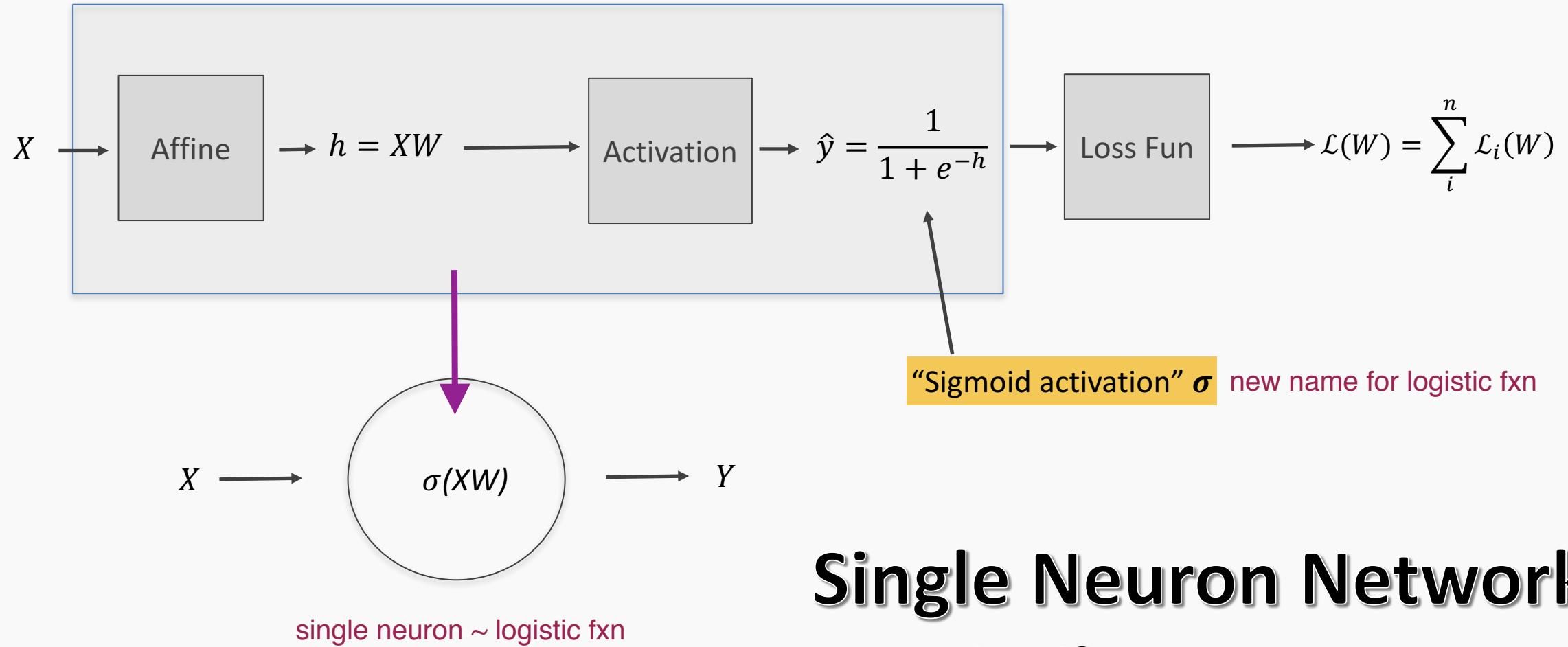
# Logistic Regression Revisited



# Build our first ANN



# Build our first ANN



**Single Neuron Network  
Very similar to Perceptron**

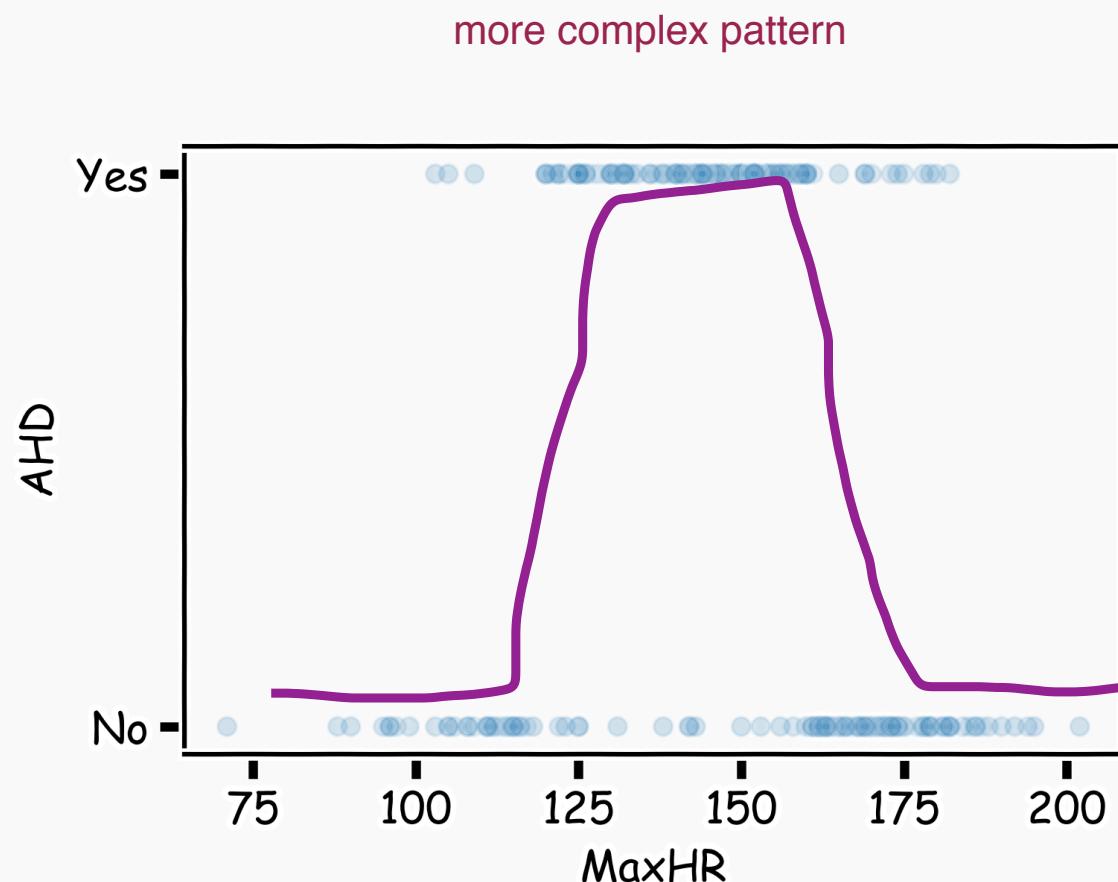
# Outline

---

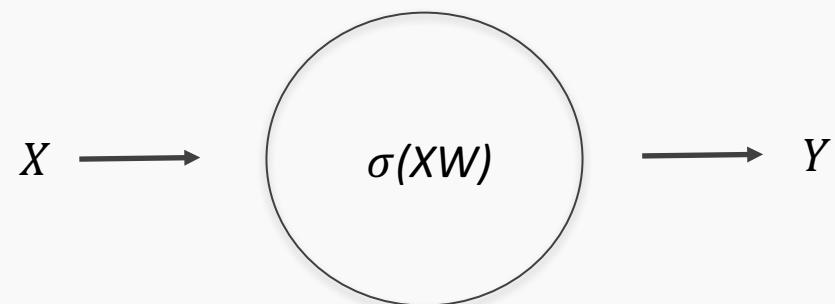
1. Introduction to Artificial Neural Networks
2. Review of Classification and Logistic Regression
3. Single Neuron Network ('Perceptron')
4. Multi-Layer Perceptron (MLP)

# Example Using Heart Data

Slightly modified data to illustrate concepts.

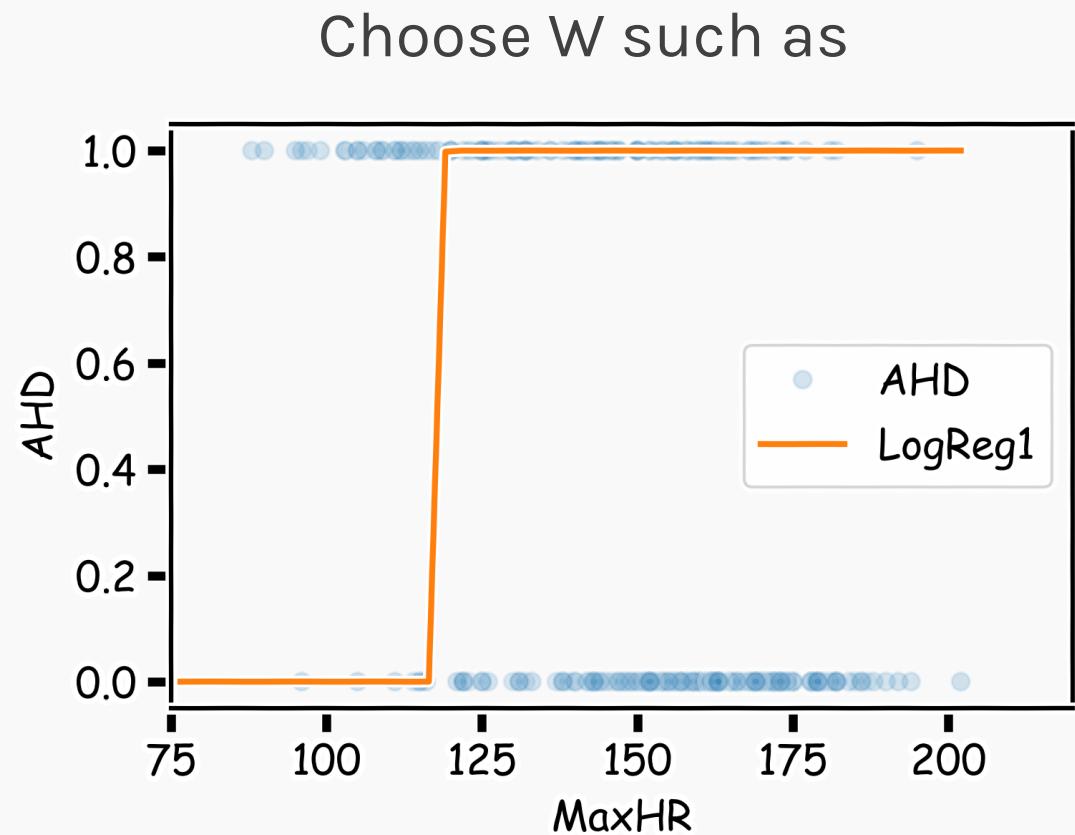
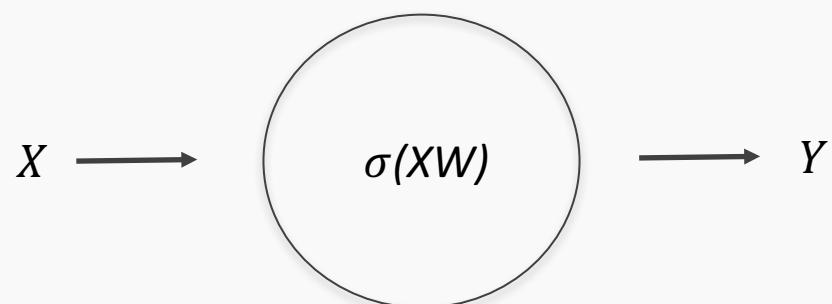


# Example Using Heart Data



does ok job on half the data

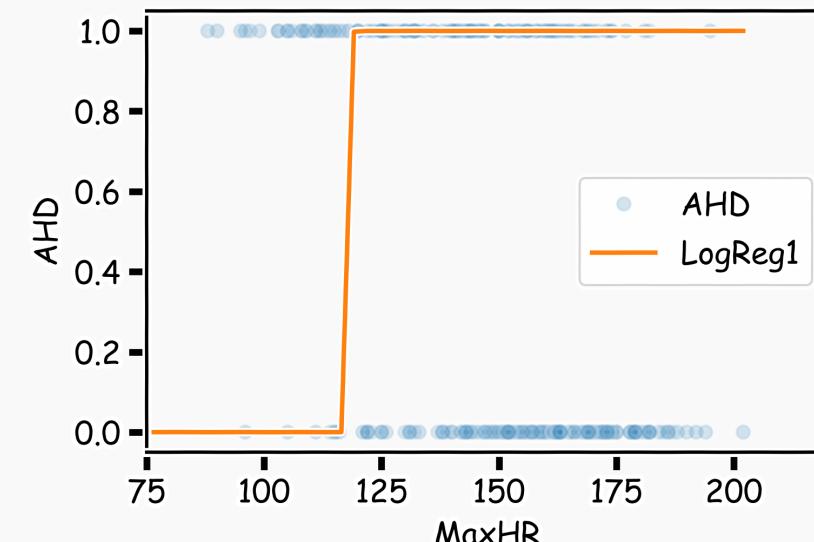
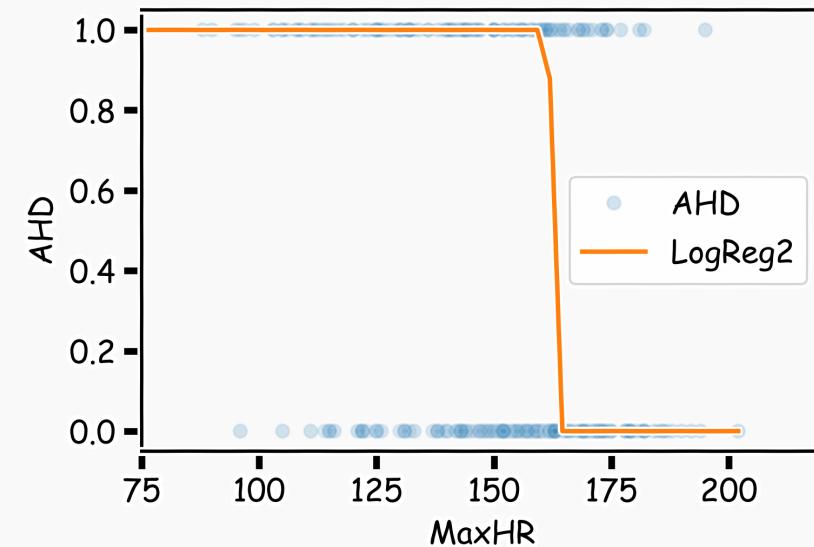
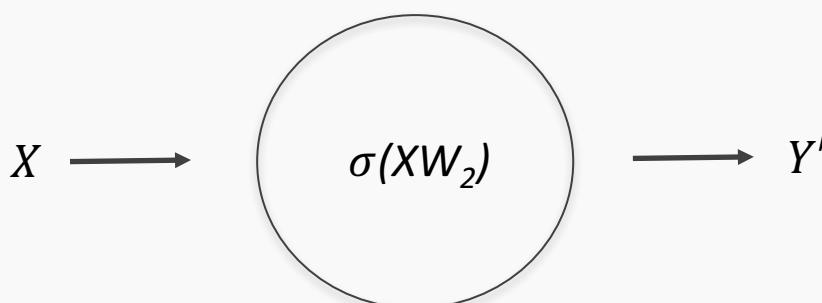
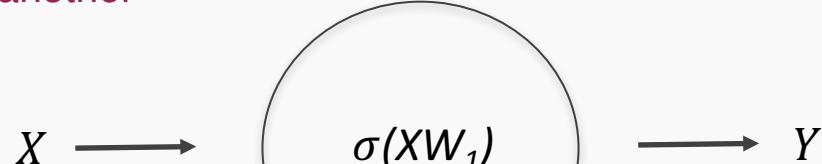
# Example Using Heart Data



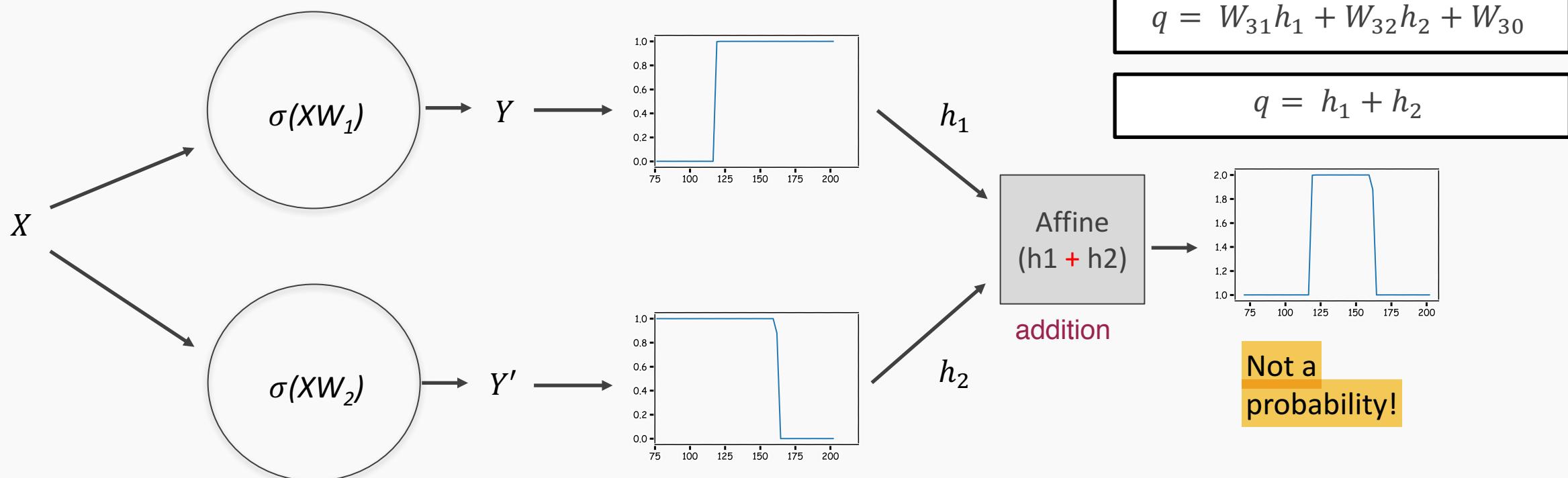
now do the other half of the data

# Example

create two models - one that does well in one area and one that does well in another

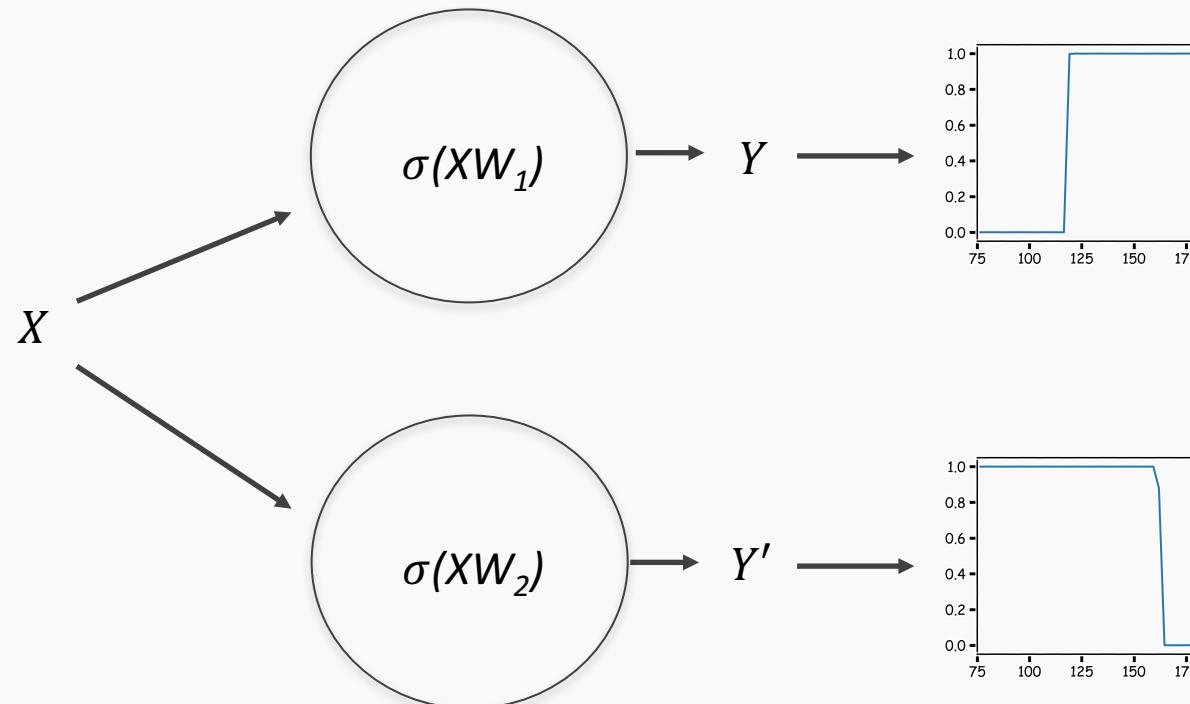


# Pavlos game #232



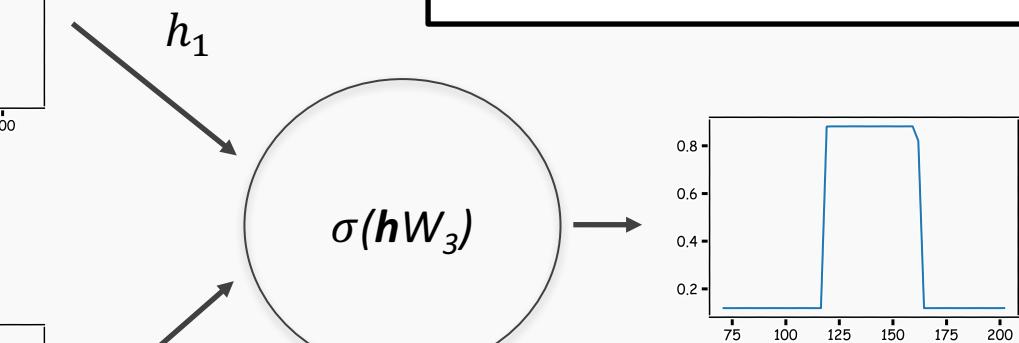
# Pavlos game #232

ta-dah it's a neural network



Need to learn  $W1$ ,  $W2$  and  $W3$ .

$$q = W_{31}h_1 + W_{32}h_2 + W_{30}$$
$$p = \frac{1}{1 + e^{-q}}$$



pass thru  
logistic  
(sigmoid) fxn;  
another neuron

Passing through sigmoid  
yields probability

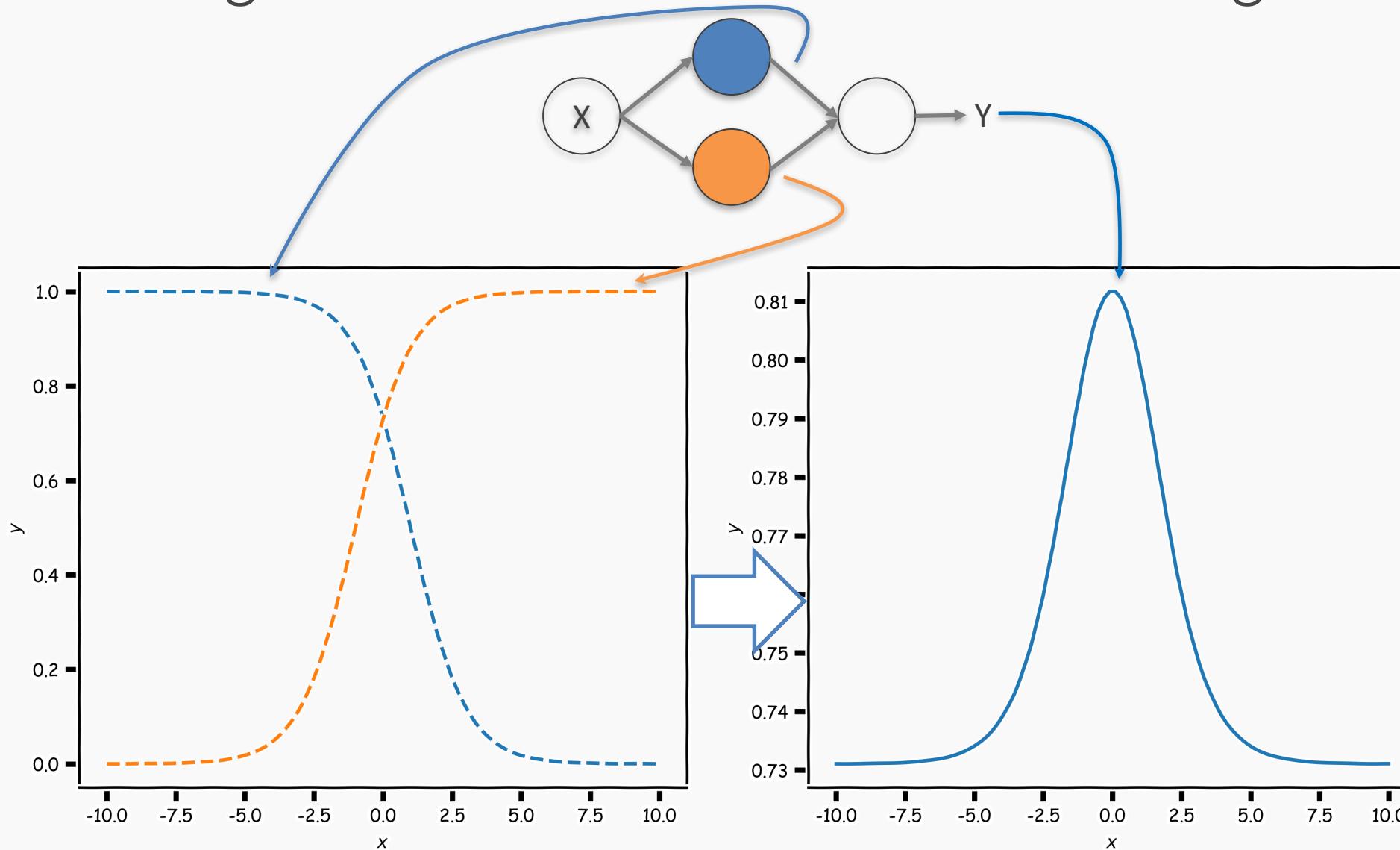
$$L = -y \ln(p) - (1 - y) \ln(1 - p)$$

loss function - use gradient descent to optimize weights by minimizing loss fxn

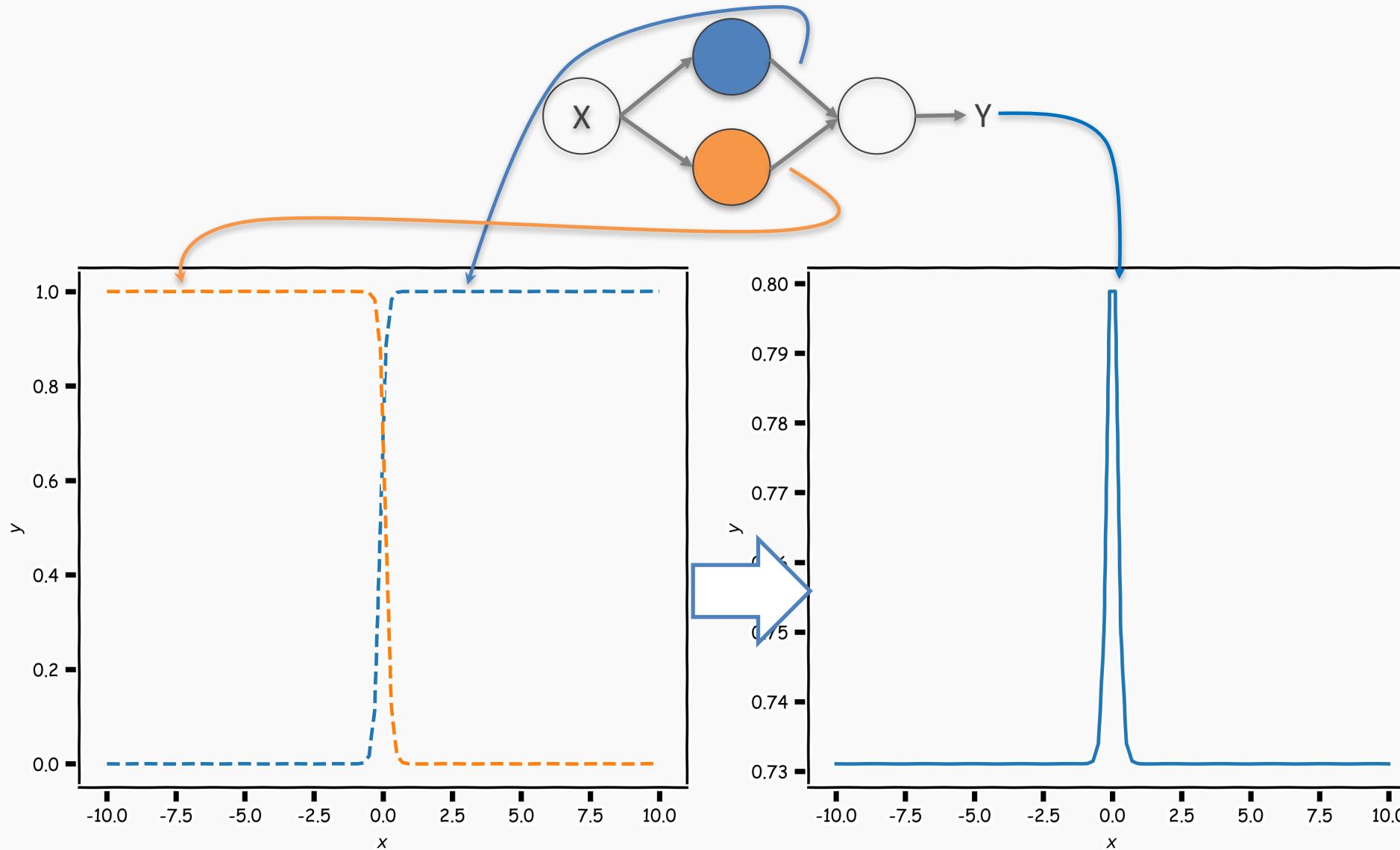
when network becomes complex, finding weights is tough



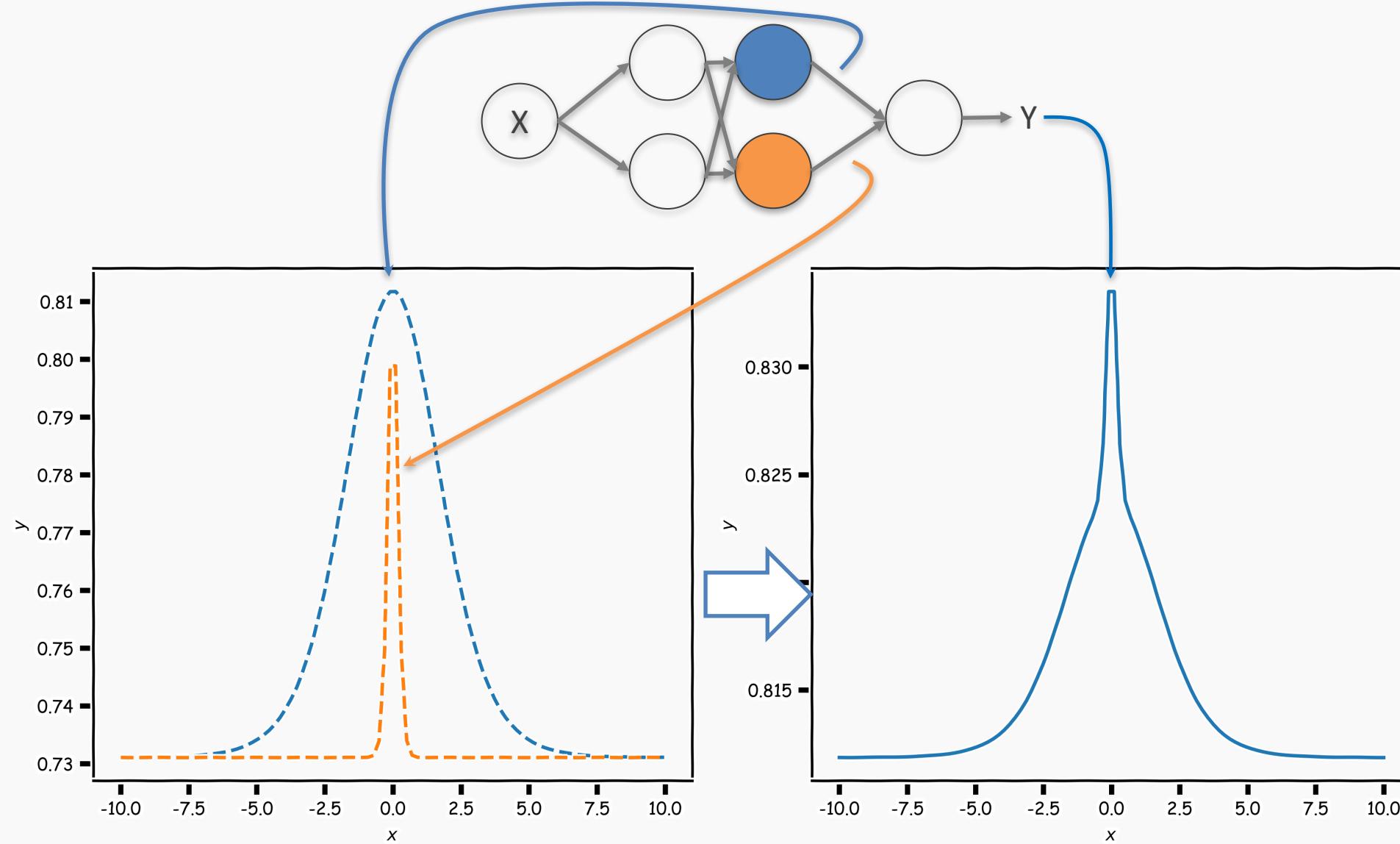
# Combining neurons allows us to model interesting functions



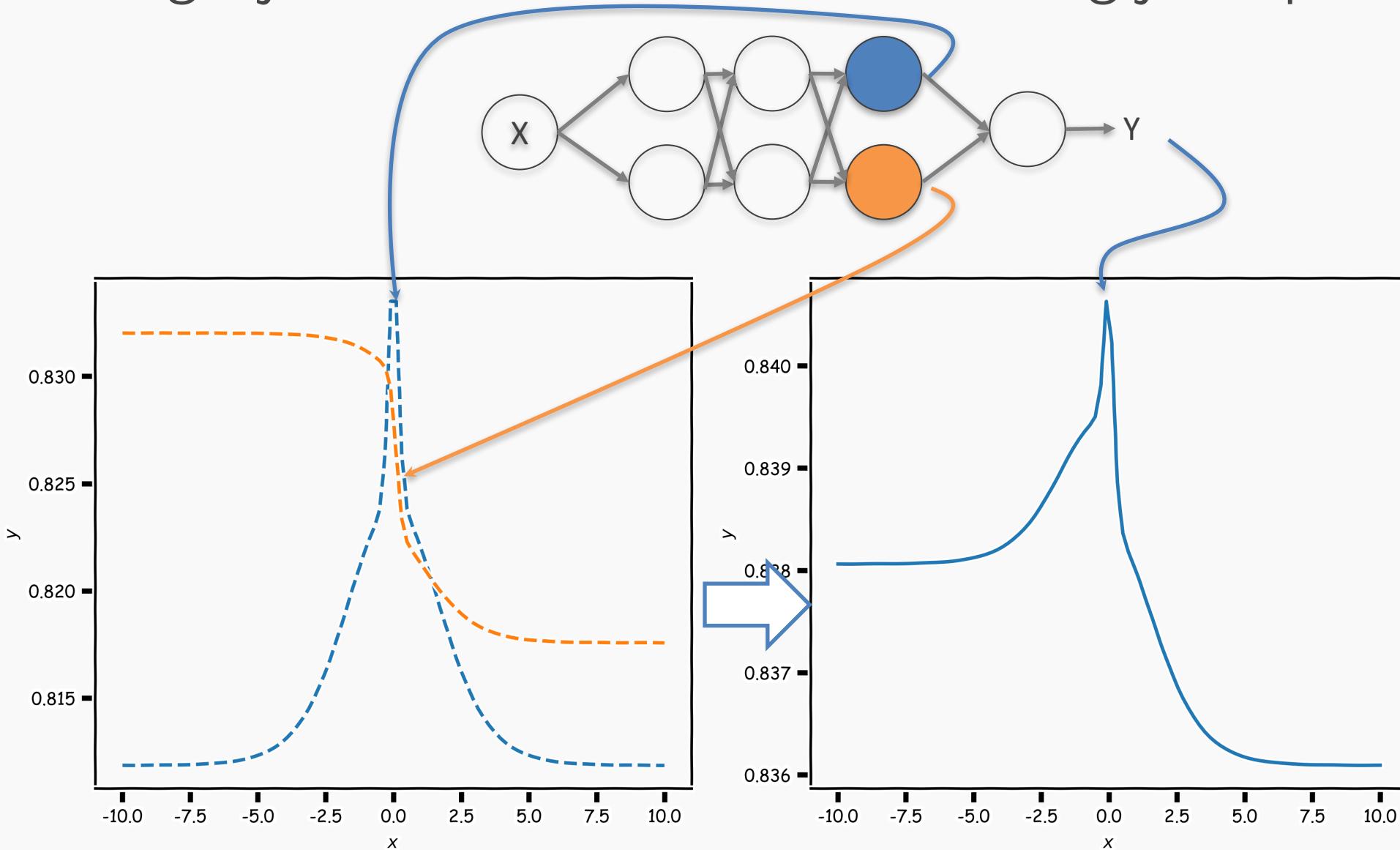
# Different weights change the shape and position



# Neural networks can model any reasonable function



Adding layers allows us to model increasingly complex functions



# Summary

---

## So far:

- A single neuron can be a logistic regression unit. We will soon see other choices. *doesn't have to be sigmoid*
- A neural network is a combination of logistic regression (or other types) units.
- A neural network can approximate non-linear functions.

## Next Lecture:

- What kind of activations, how many neurons, how many layers, output unit and loss function?

## Following two lectures on NN:

- How do we estimate the weights and biases?
- How to regularize Neural Networks.

*complexity comes with responsibility (danger of overfitting)*

# Exercises

---

Quiz (survey)

And two super cool ED exercises

