

Perceptron and Multilayer Perceptron

Pavlos Protopapas

Outline

1. Introduction to Artificial Neural Networks
2. Review of basic concepts
3. Single Neuron Network ('Perceptron')
4. Multi-Layer Perceptron (MLP)



Outline

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



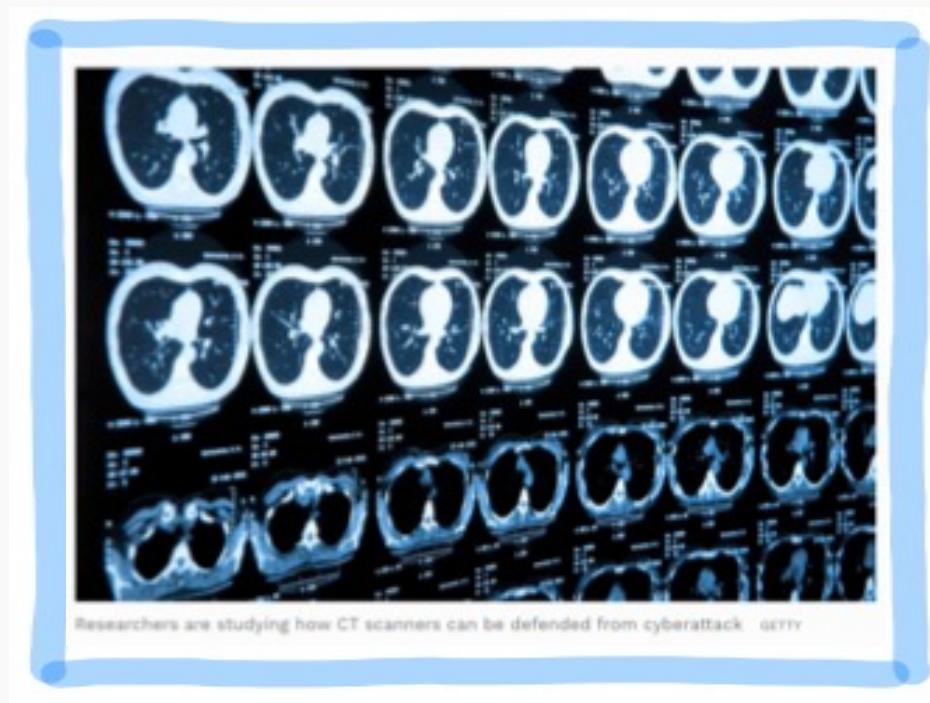
Today's news

Skin Conditions



Using Deep Learning in diagnosing skin conditions

Stopping Cyberattacks

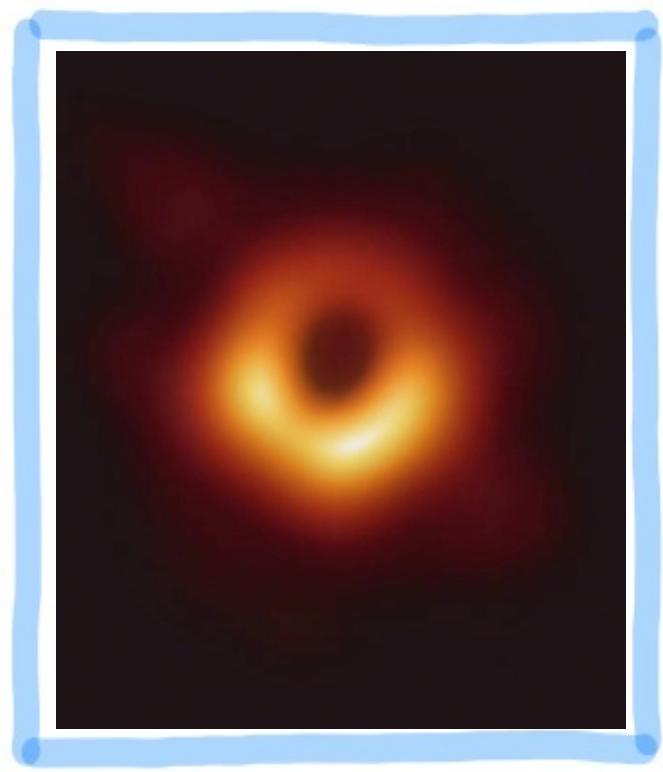


Detecting tampering with the diagnostic images, or quietly upped the radiation levels.



Today's news

Image generation



Katie Bouman's CHIRP produces the first-ever image of a black hole.

Computer Code Generation

Sharif Shameem
@sharifshameem

This is mind blowing.

With GPT-3, I built a layout generator where you just describe any layout you want, and it generates the JSX code for you.

W H A T

Describe a layout.

Just describe any layout you want, and it'll try to render below!

A div that contains 3 buttons each with a random color. Generate

Play button

The image shows a screenshot of a Twitter post by Sharif Shameem (@sharifshameem). The post includes a profile picture, the author's name, their Twitter handle, and a short caption: "This is mind blowing." Below the caption, Sharif writes: "With GPT-3, I built a layout generator where you just describe any layout you want, and it generates the JSX code for you." Underneath this, there is a section titled "W H A T" containing a text input field with the placeholder "Just describe any layout you want, and it'll try to render below!" and a "Generate" button. Below the input field is a large blue play button. The entire screenshot is framed by a blue border.



The Potential of Data Science

Gender Bias



Some DS models for evaluate job applications show bias in favor of male candidate

Racial Bias



Risk models used in US courts have shown to be biased against non-white defendants



Historical Trends

Disease prediction

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

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



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

Game strategy



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



2018

Natural Language Processing

"Siri, what is Deep Learning?"
tap to edit



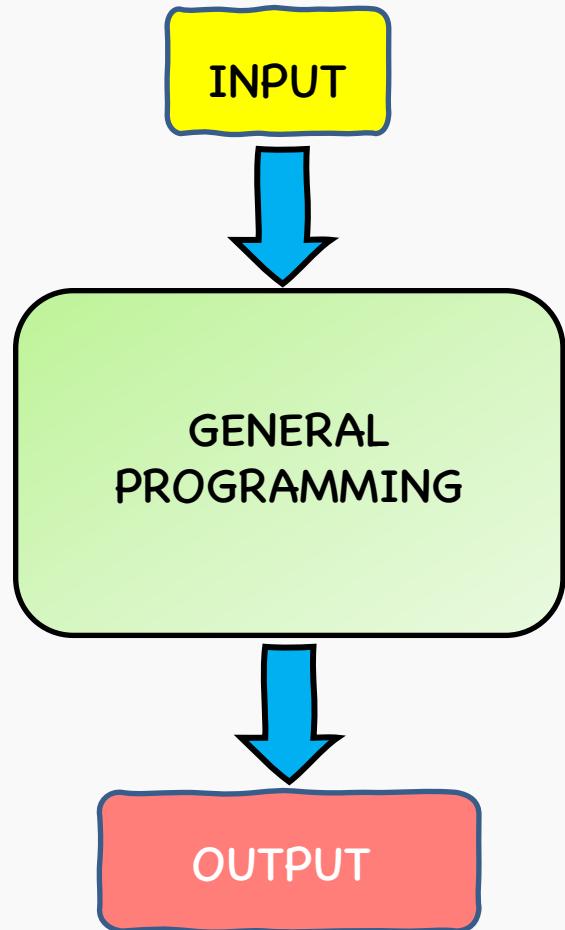
Outline

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

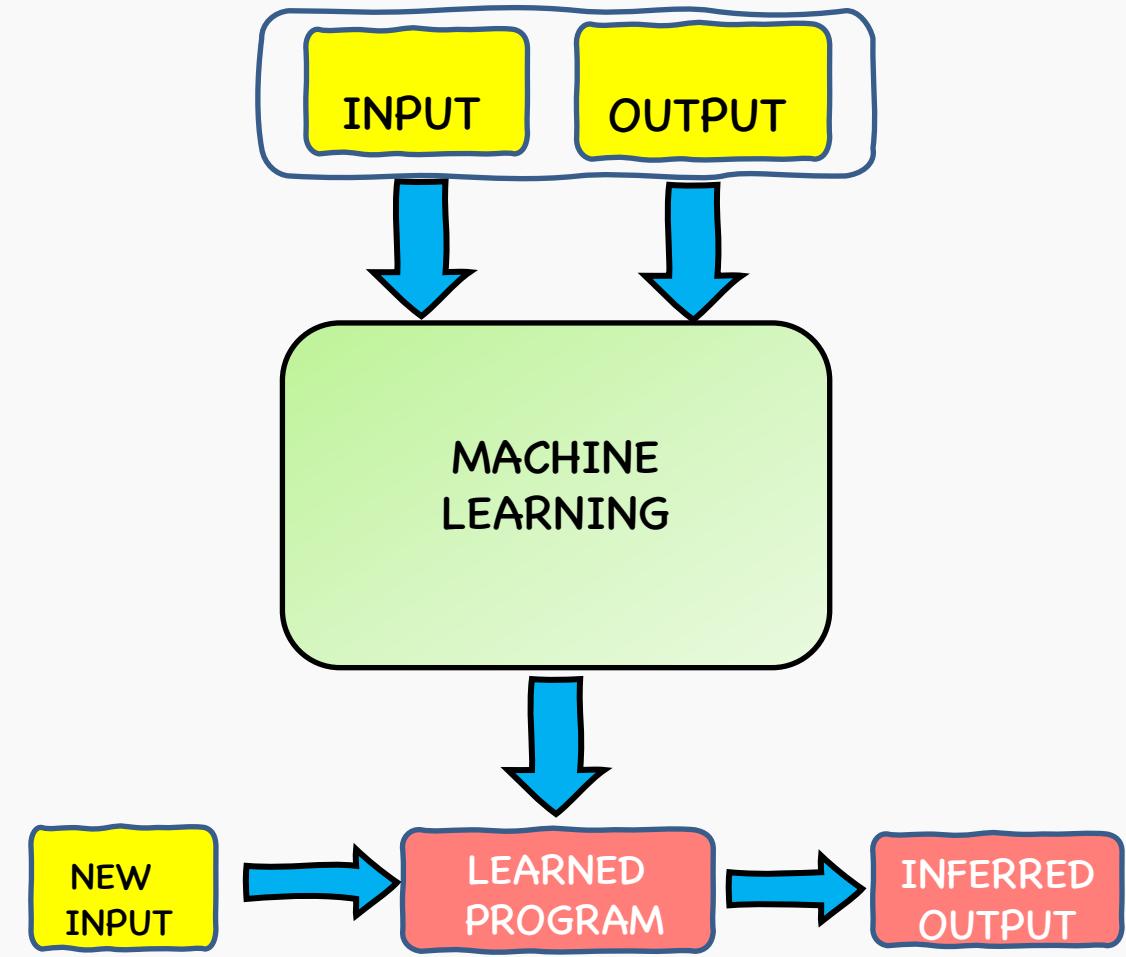


What is Machine Learning?

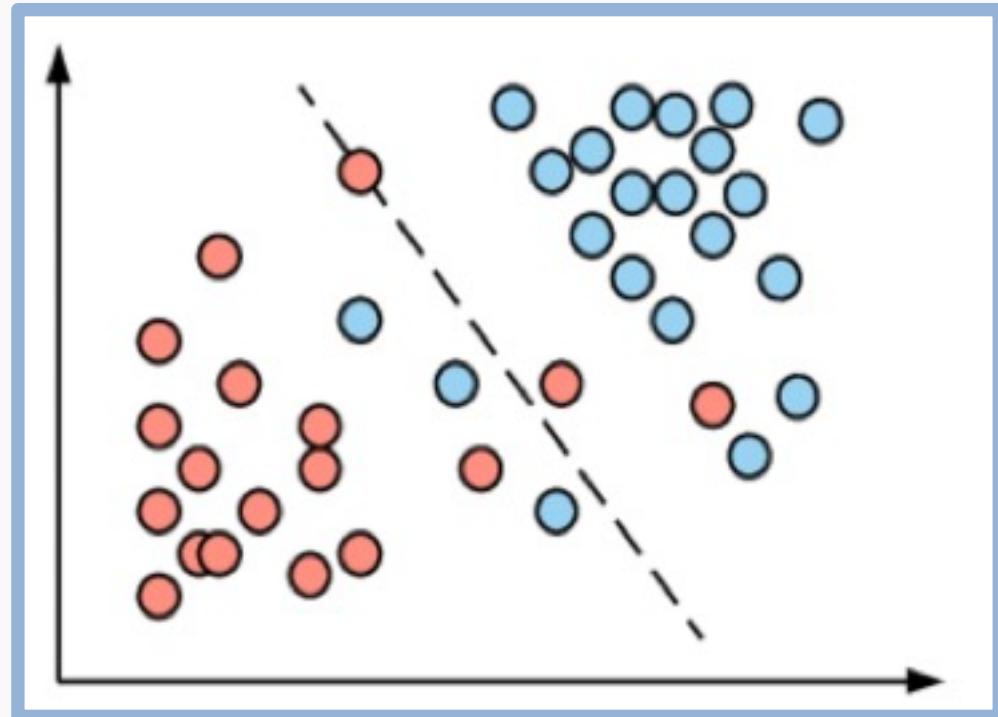
TRADITIONAL PROGRAMMING



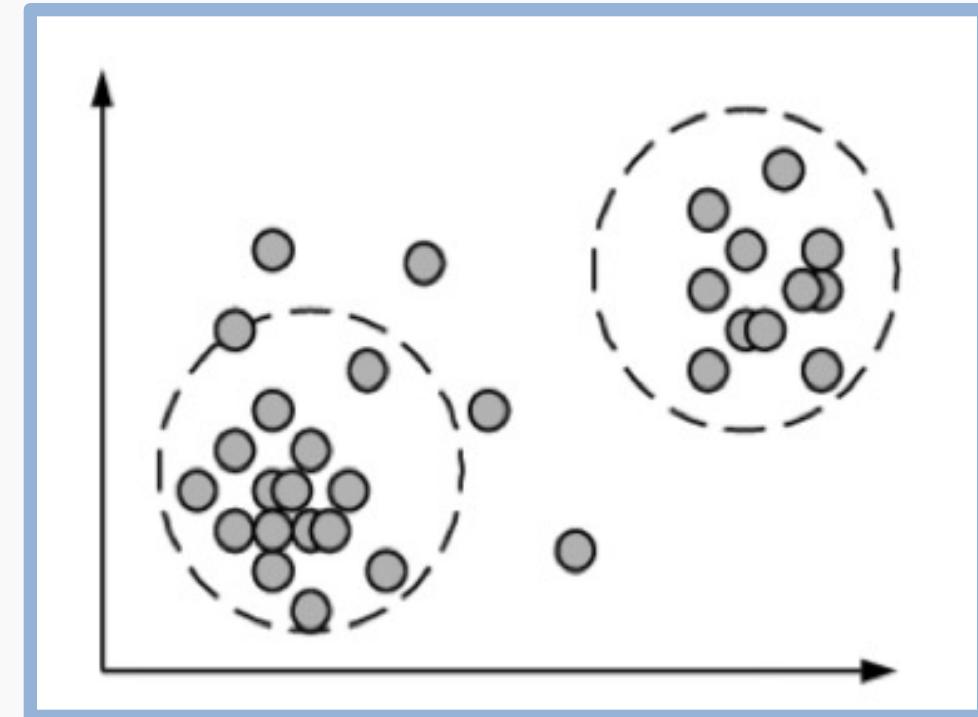
MACHINE LEARNING



Supervised v/s Unsupervised Machine Learning



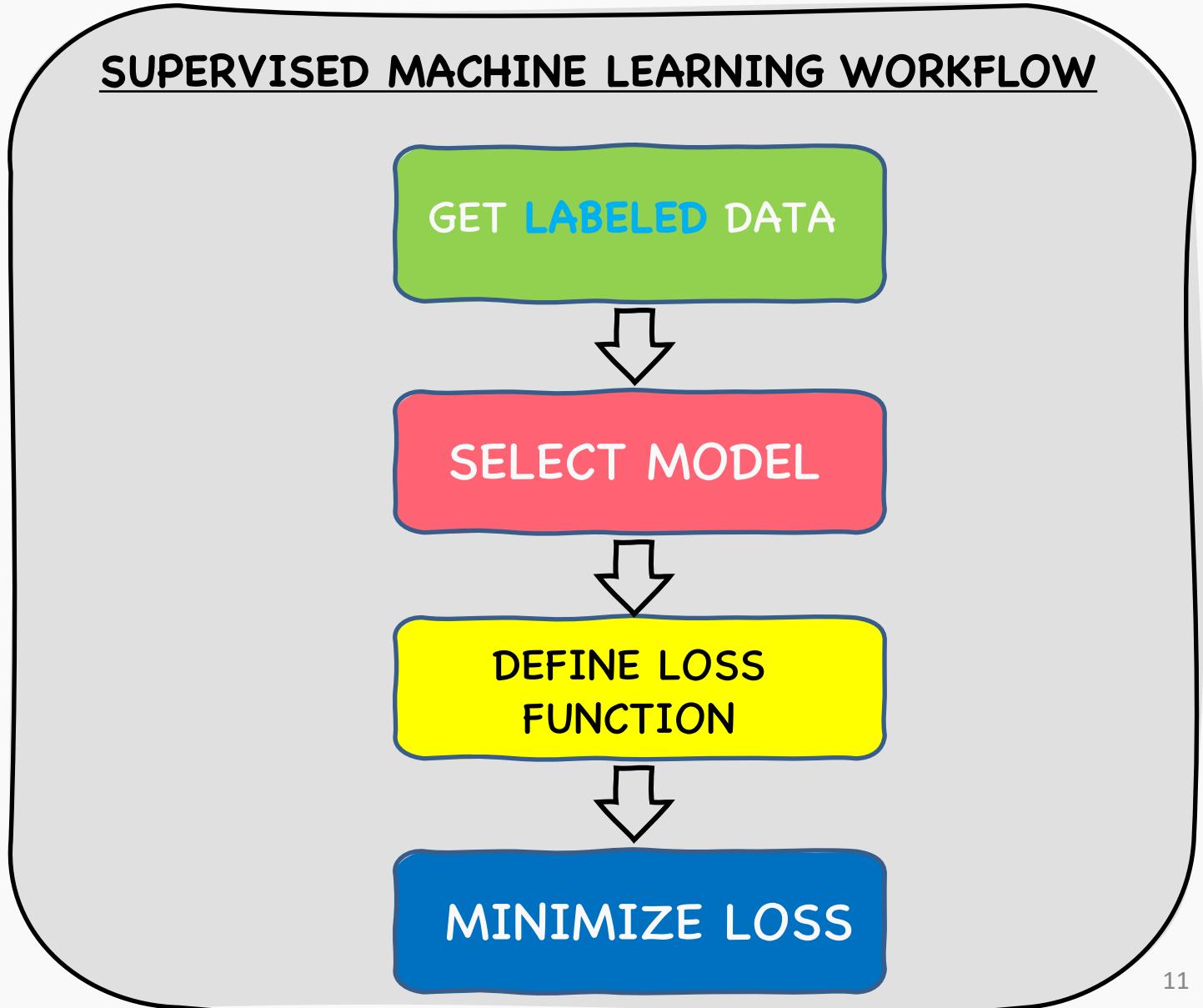
Supervised Learning: Learns with “labeled” data



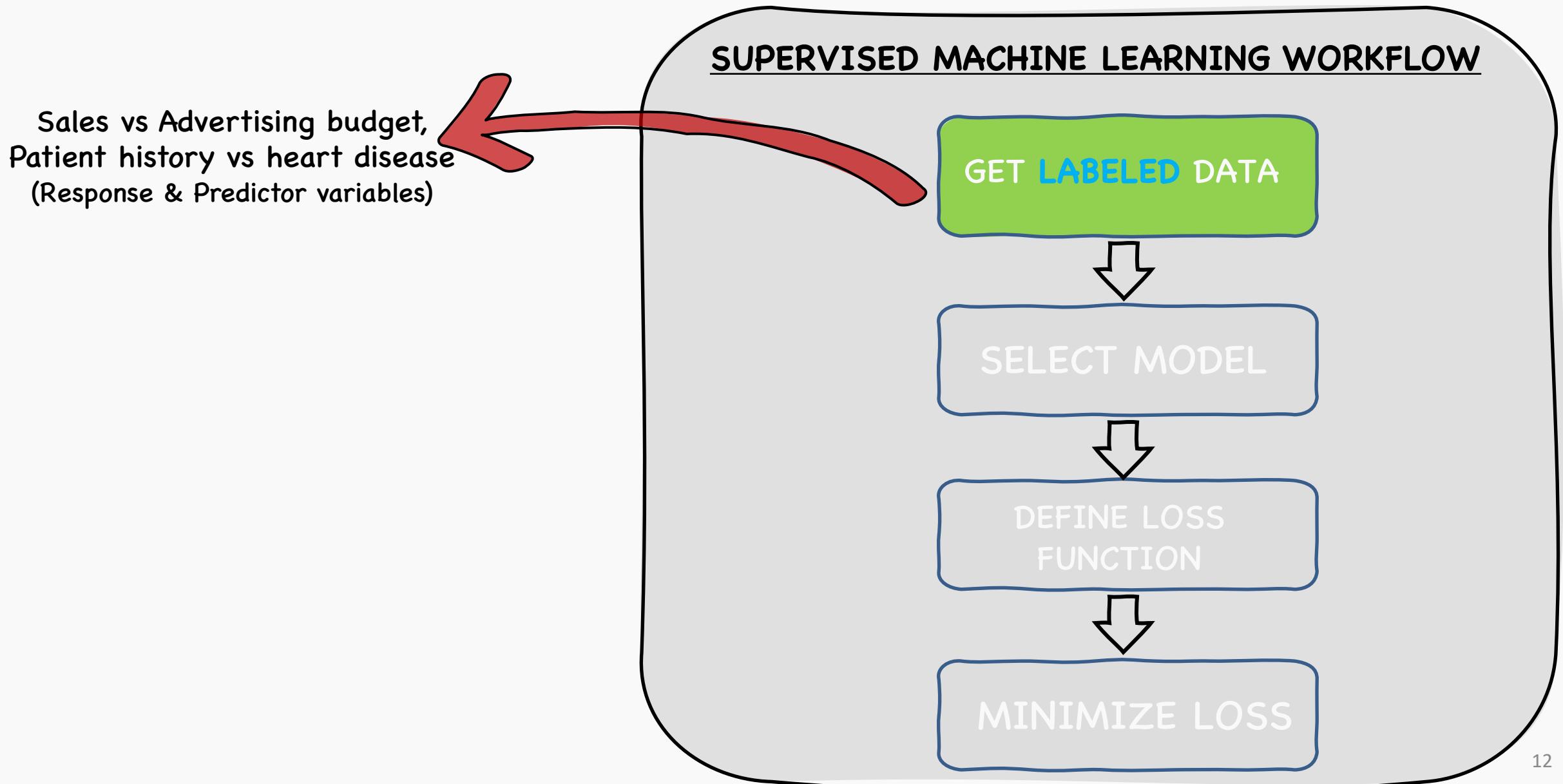
Unsupervised Learning: Learns by clustering or association



Building blocks of supervised machine learning



Building blocks of supervised machine learning



Response vs. Predictor Variables

$X = X_1, \dots, X_p$
 $X_j = x_{1j}, \dots, x_{ij}, \dots, x_{nj}$
predictors
features
covariates

response variable Y
is continuous

$Y = y_1, \dots, y_n$
outcome
response variable
dependent variable

n observations

p predictors

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



Heart Data

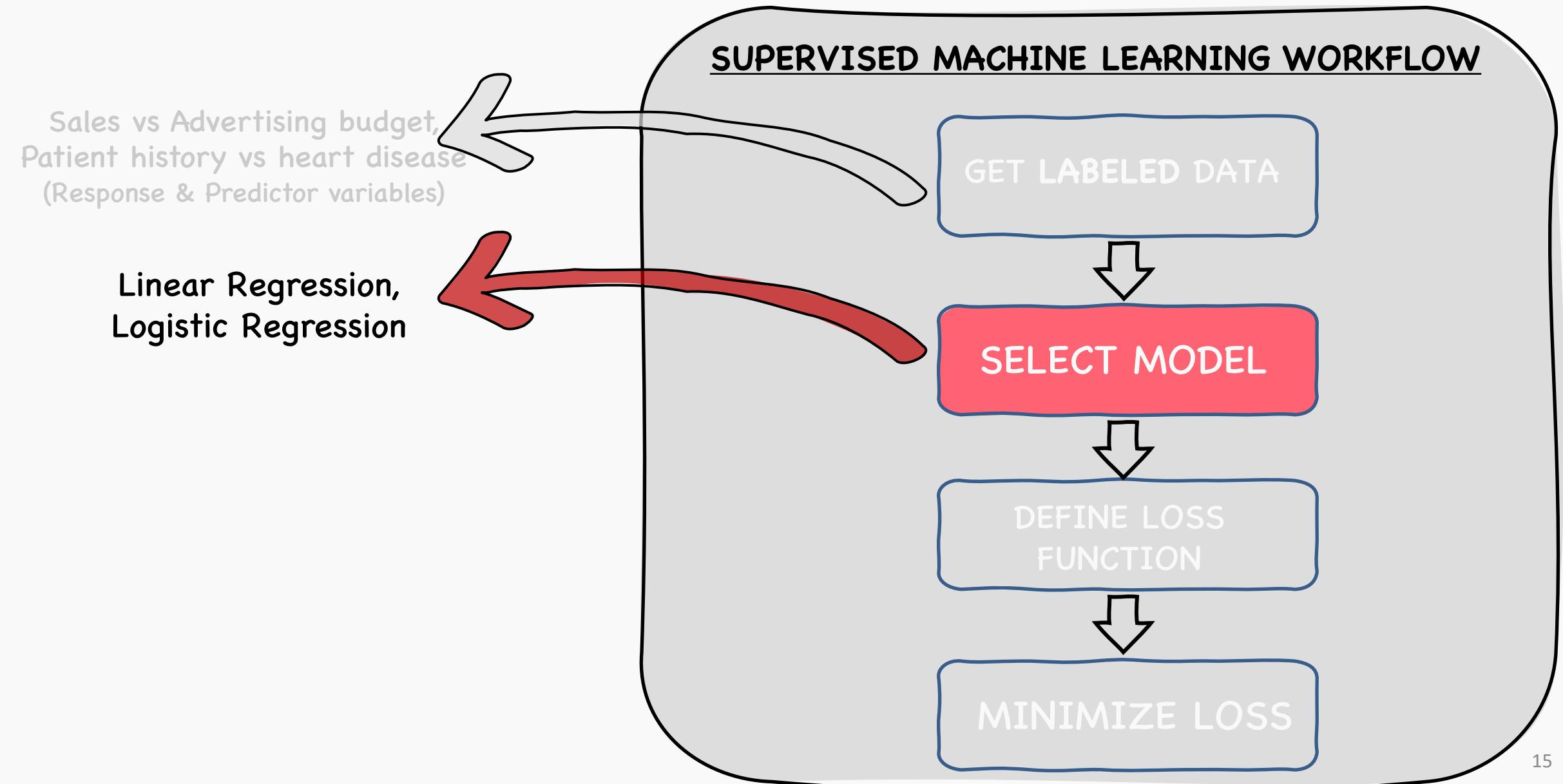
These data contain a binary outcome AHD for 303 patients who presented with chest pain.

response variable Y
is Yes/No

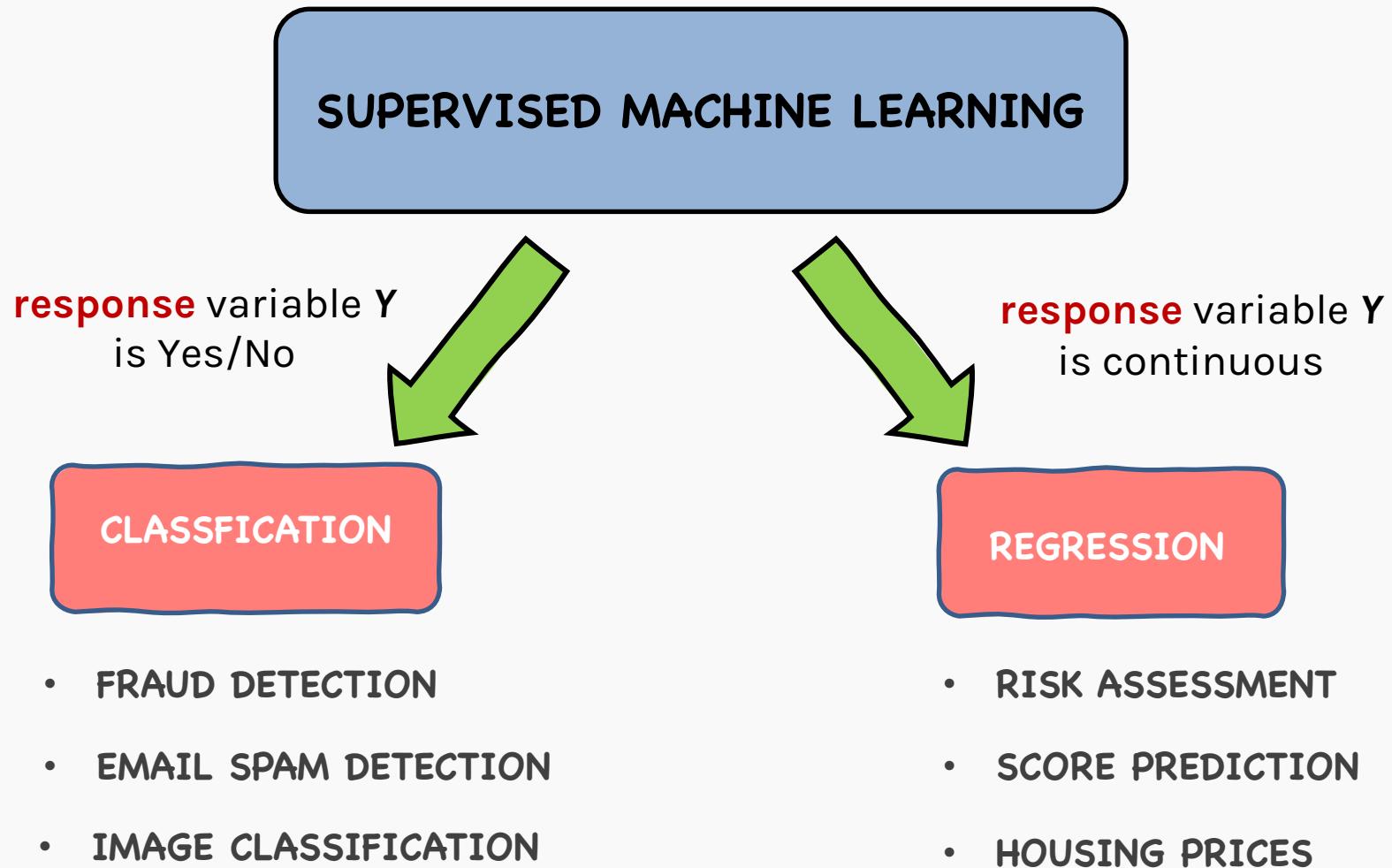
Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak	Slope	Ca	Thal	AHD
63	1	typical	145	233	1	2	150	0	2.3	3	0.0	fixed	No
67	1	asymptomatic	160	286	0	2	108	1	1.5	2	3.0	normal	Yes
67	1	asymptomatic	120	229	0	2	129	1	2.6	2	2.0	reversible	Yes
37	1	nonanginal	130	250	0	0	187	0	3.5	3	0.0	normal	No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



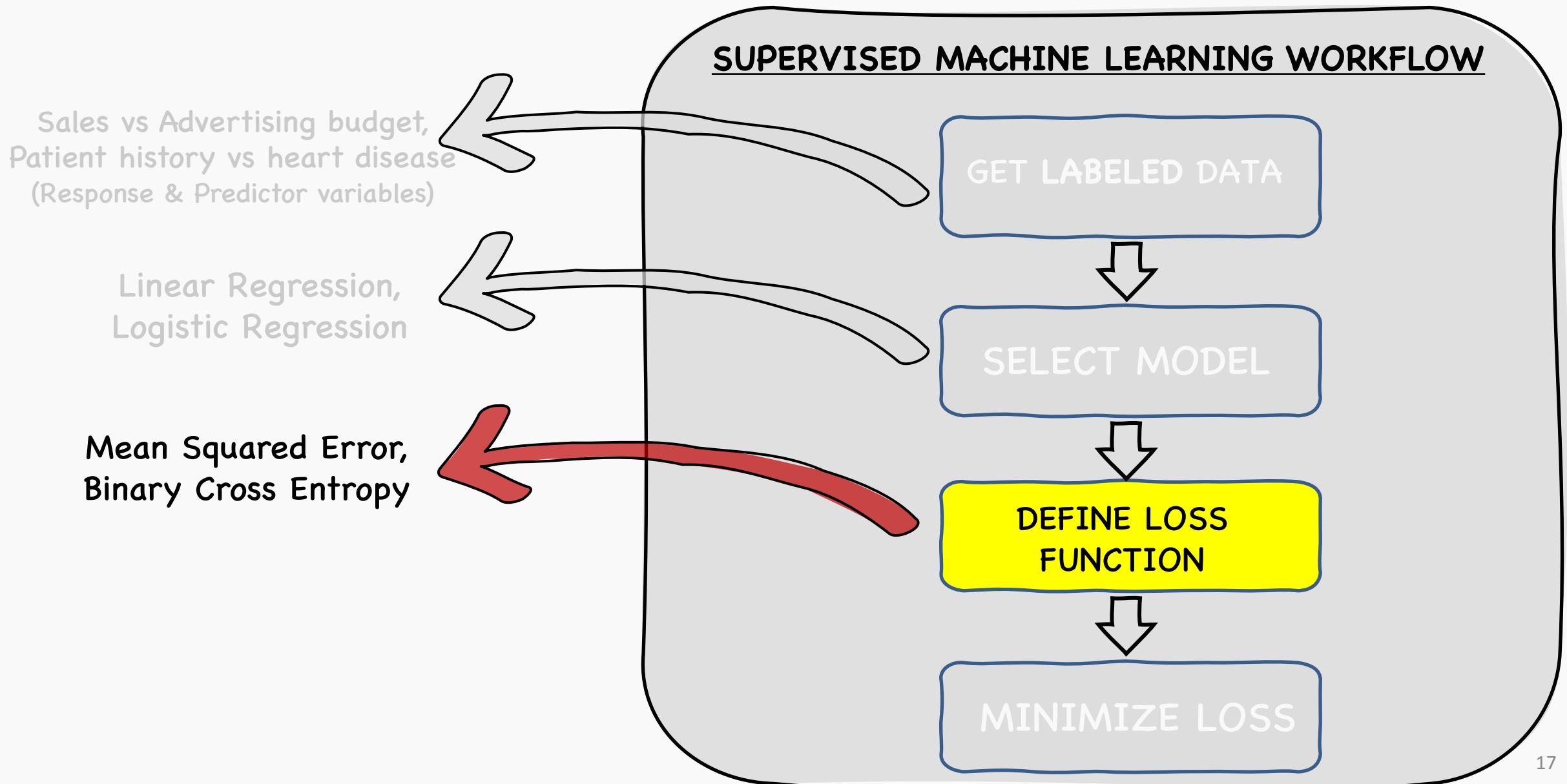
Building blocks of supervised machine learning



Supervised Machine Learning examples



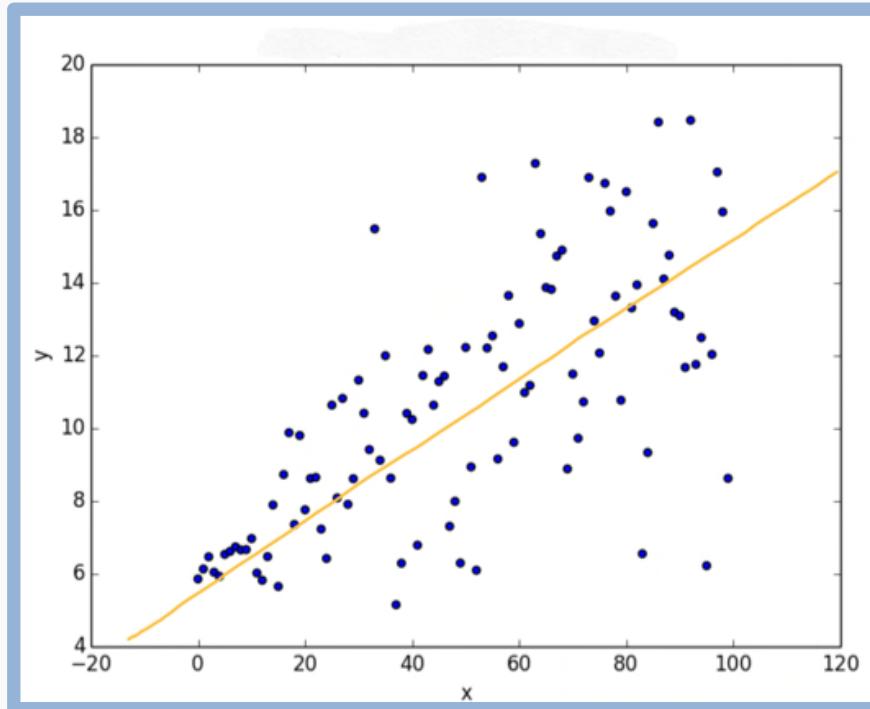
Building blocks of supervised machine learning



Estimate of the regression coefficients (cont)

MSE as the **loss function**,

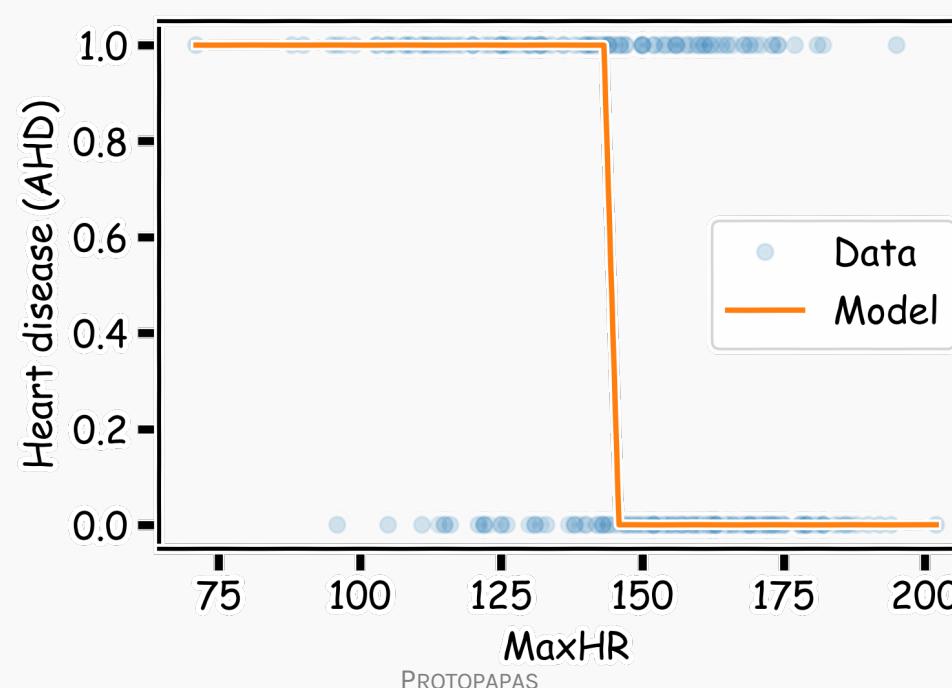
$$L(\beta_0, \beta_1) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \frac{1}{n} \sum_{i=1}^n [y_i - (\beta_1 X + \beta_0)]^2.$$



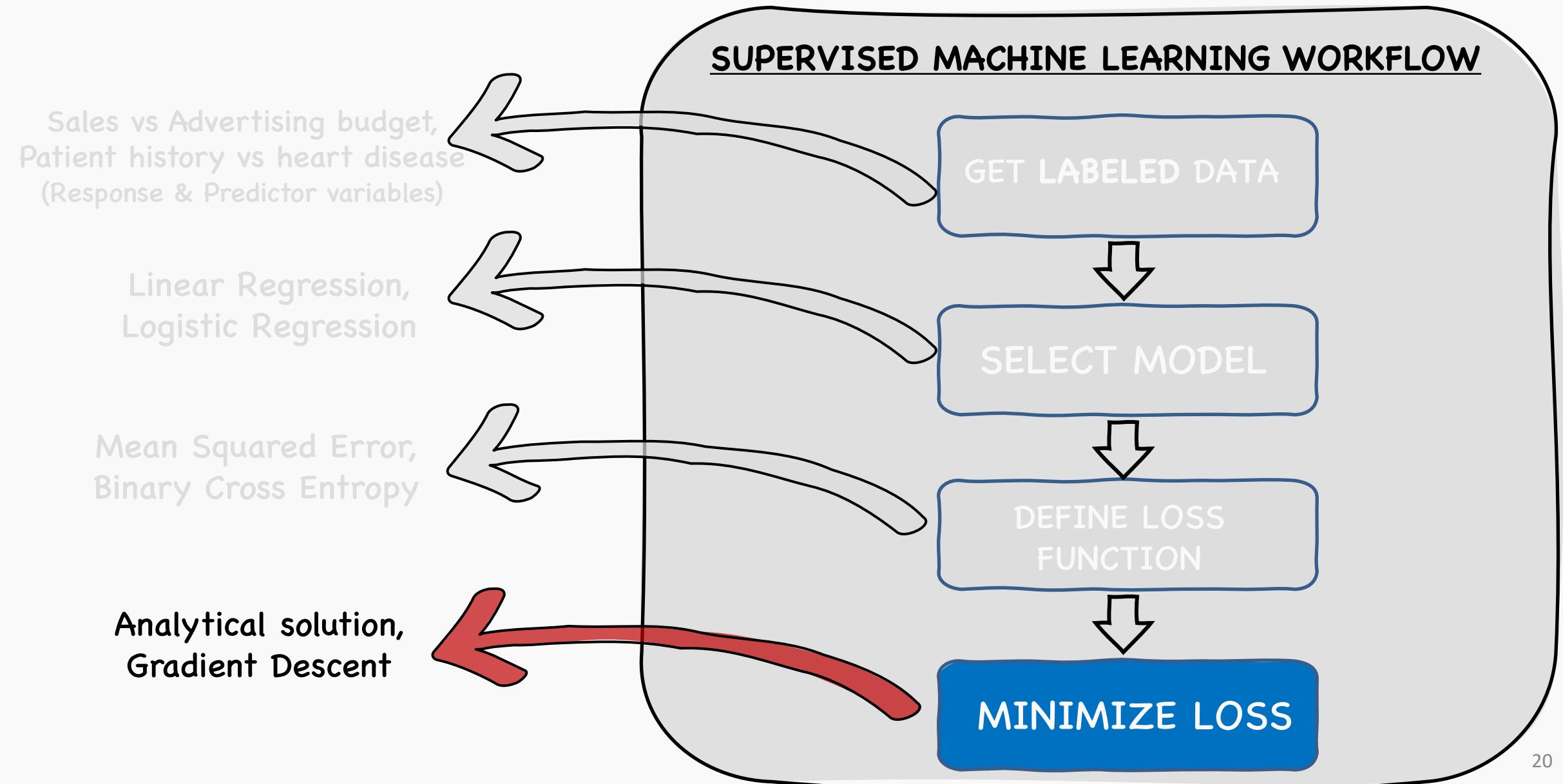
Loss function for Logistic Regression

Cross Entropy as a loss function

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

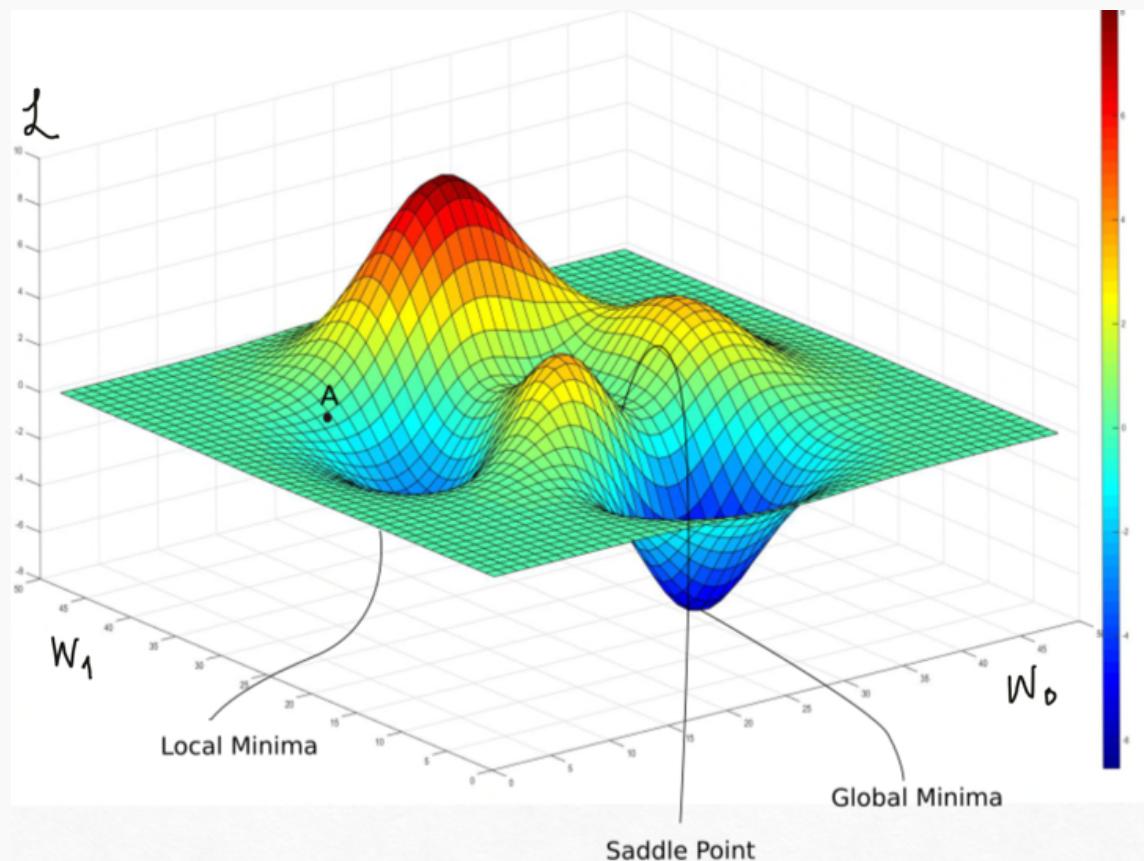


Building blocks of supervised machine learning



Optimization

How does one minimize a loss function?



The global minima or maxima of $L(\beta_0, \beta_1)$ must occur at a point where the gradient (slope)

$$\nabla L = \left[\frac{\partial L}{\partial \beta_0}, \frac{\partial L}{\partial \beta_1} \right] = 0$$

- **Brute Force:** Try every combination
- **Exact:** Solve the above equation
- **Greedy Algorithm:** Gradient Descent

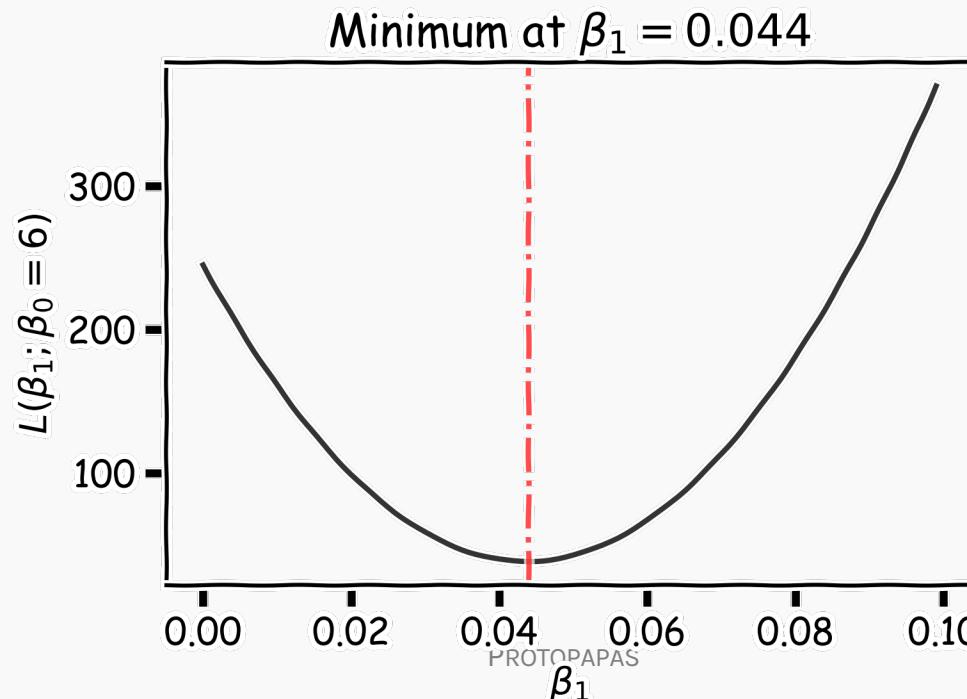


Optimization: Estimate of the regression coefficients

Brute force

A way to estimate $\operatorname{argmin}_{\beta_0, \beta_1} L$ is to calculate the loss function for every possible β_0 and β_1 . Then select the β_0 and β_1 where the loss function is minimum.

E.g. the loss function for different β_1 when β_0 is fixed to be 6:



Very **computationally expensive** with many coefficients



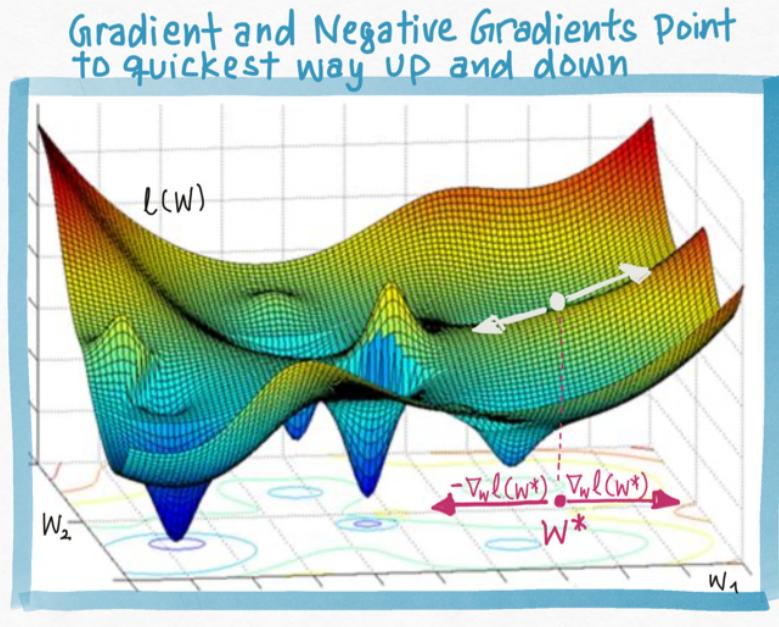
Gradient Descent

When we can't analytically solve for the stationary points of the gradient, we can still exploit the information in the gradient.

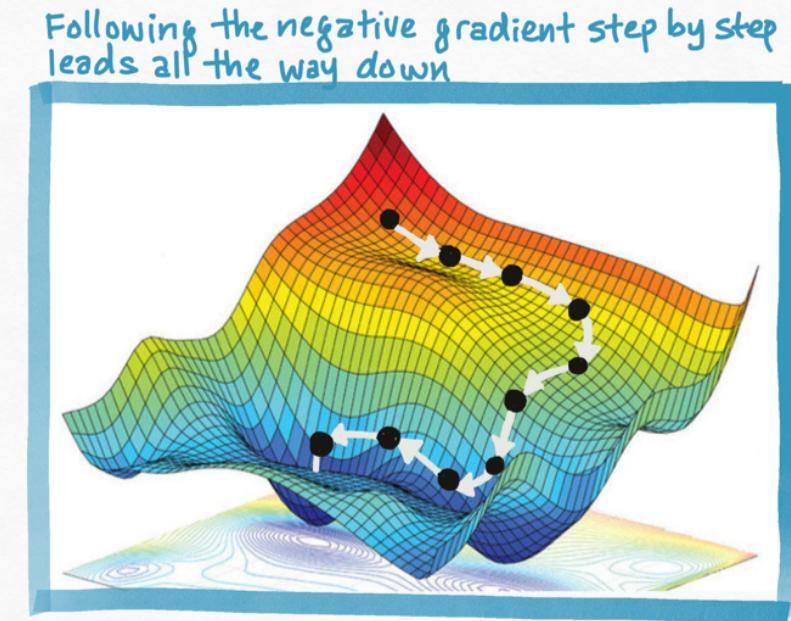
The gradient ∇L at any point is the **direction of the steepest increase**. The negative gradient is the **direction of steepest decrease**.

By following the -ve gradient, we can eventually find the lowest point.

This method is called **Gradient Descent**



PROTOPAPAS

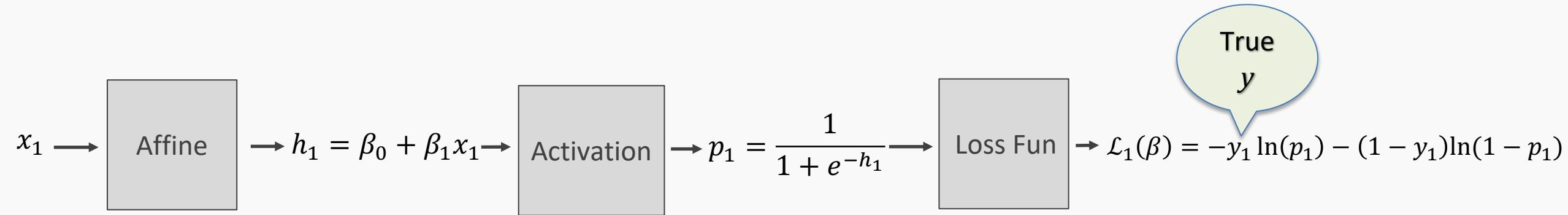


Outline

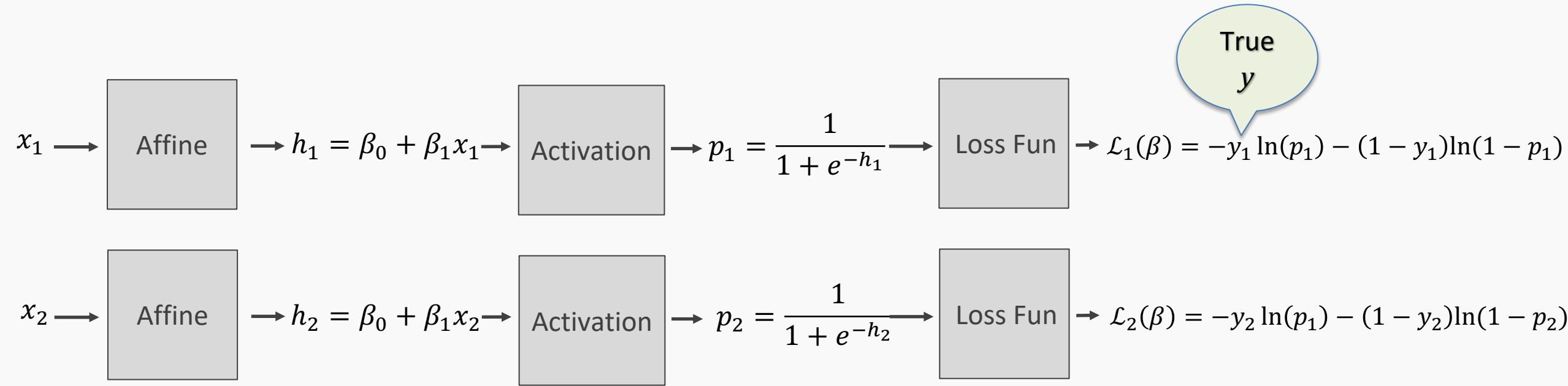
1. Introduction to Artificial Neural Networks
2. Review of basic concepts
- 3. Single Neuron Network ('Perceptron')**
4. Multi-Layer Perceptron (MLP)



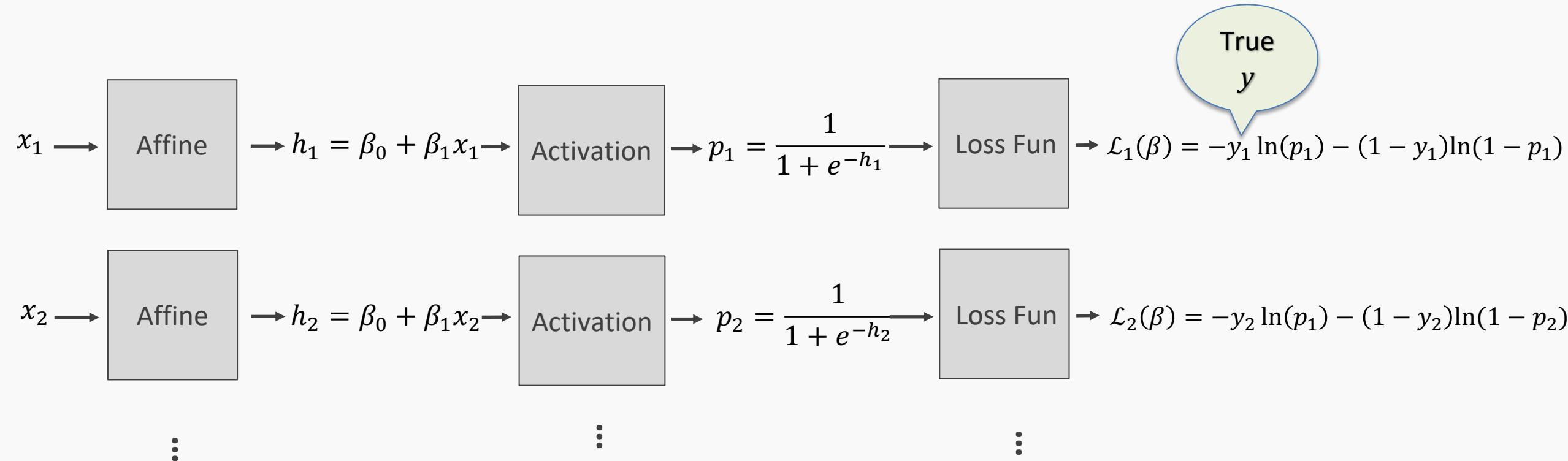
Logistic Regression Revisited



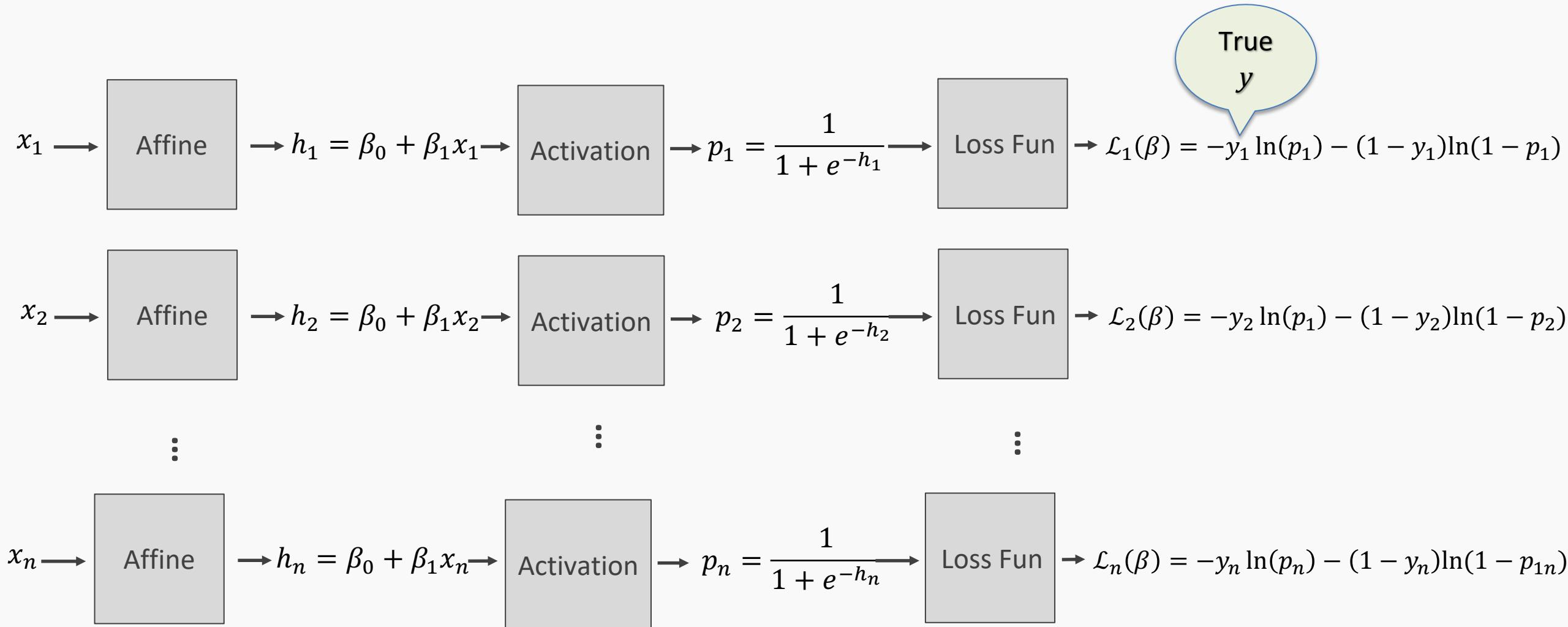
Logistic Regression Revisited



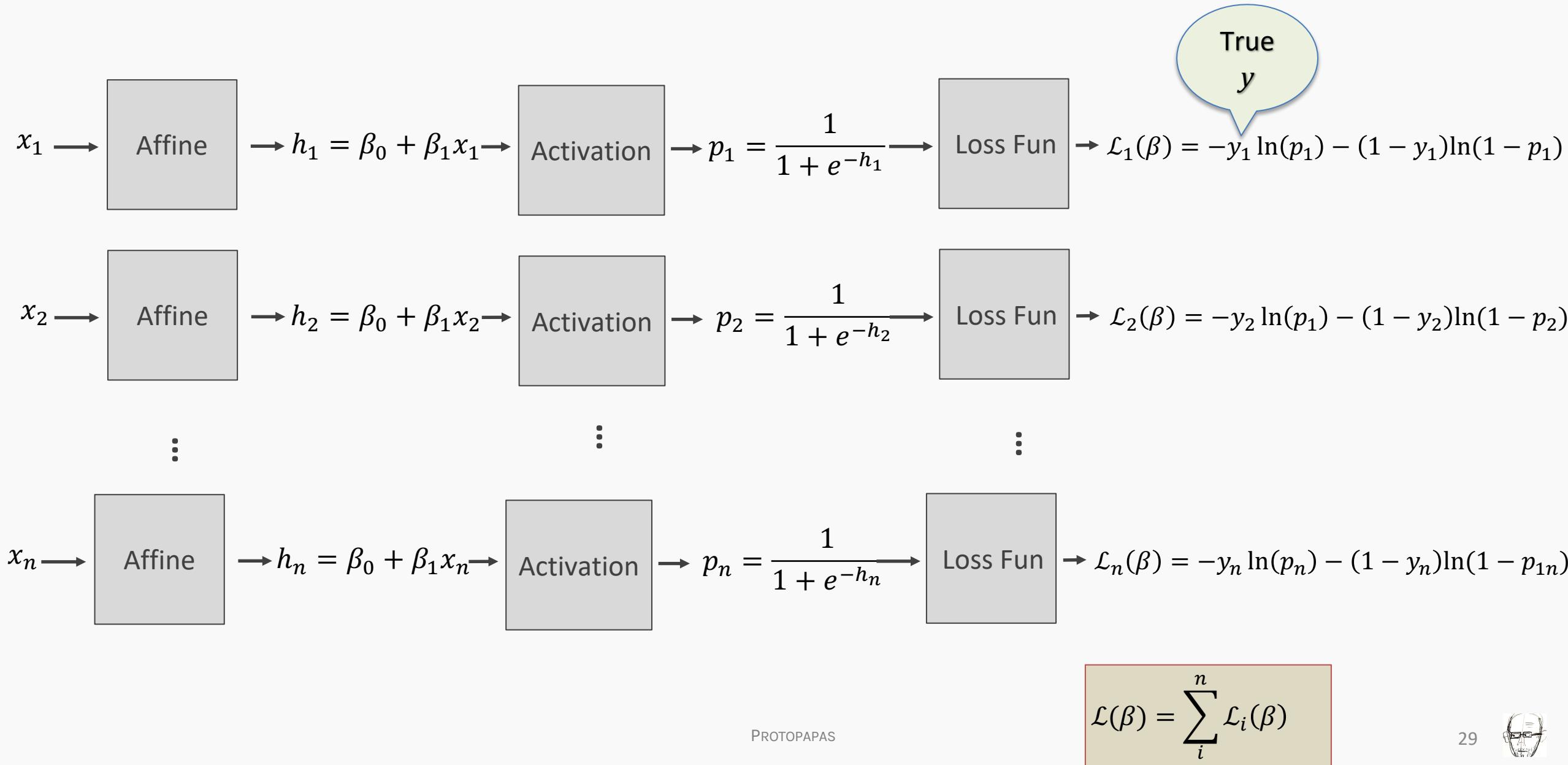
Logistic Regression Revisited



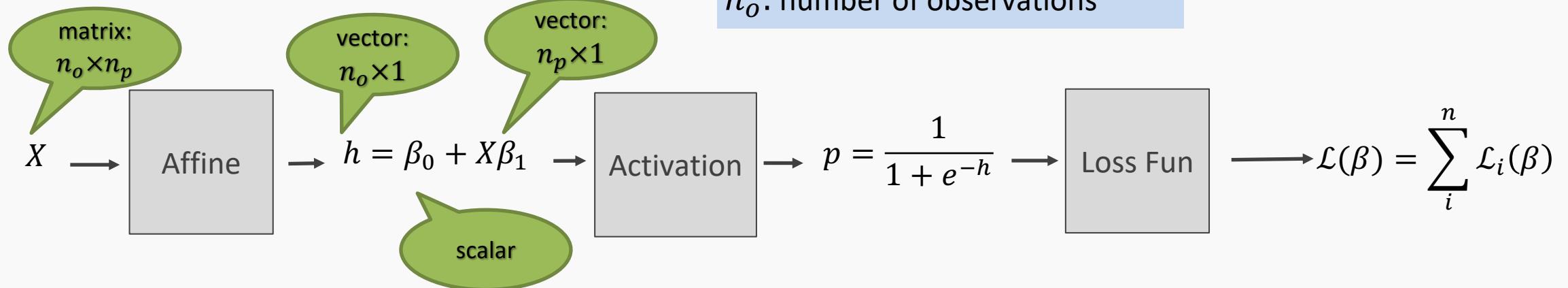
Logistic Regression Revisited



Logistic Regression Revisited

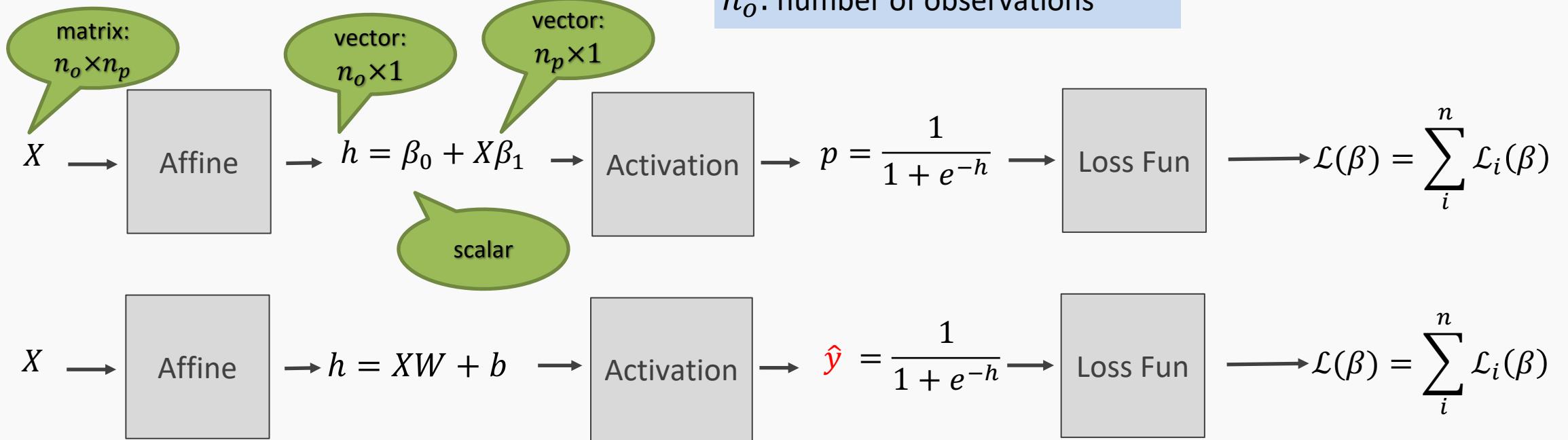


Build our first ANN



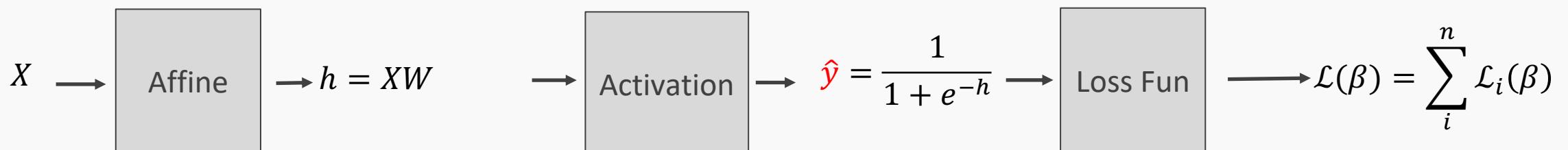
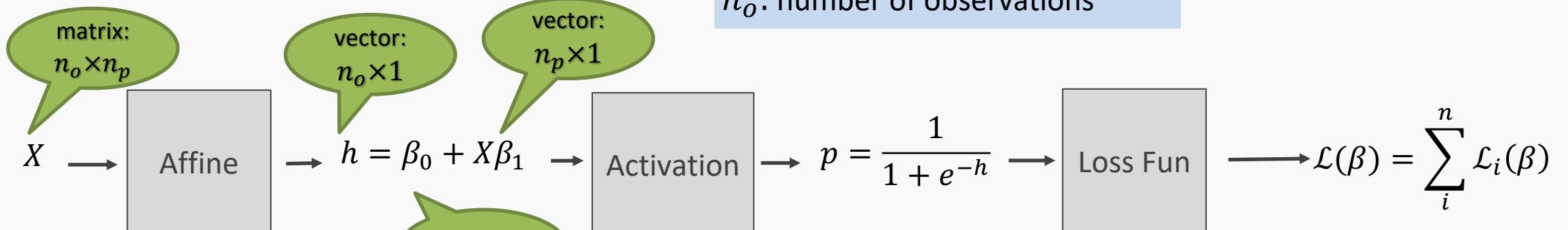
Build our first ANN

n_p : number of predictors
 n_o : number of observations



Build our first ANN

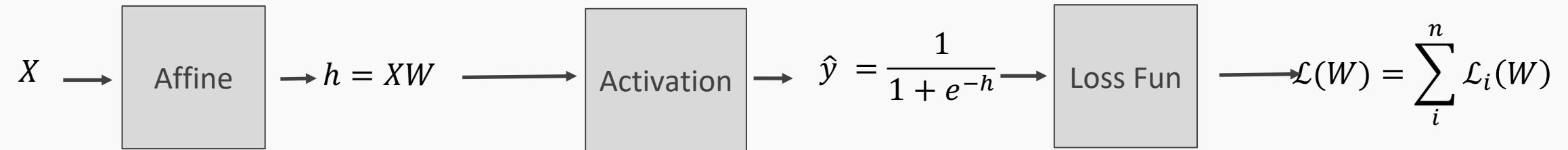
n_p : number of predictors
 n_o : number of observations



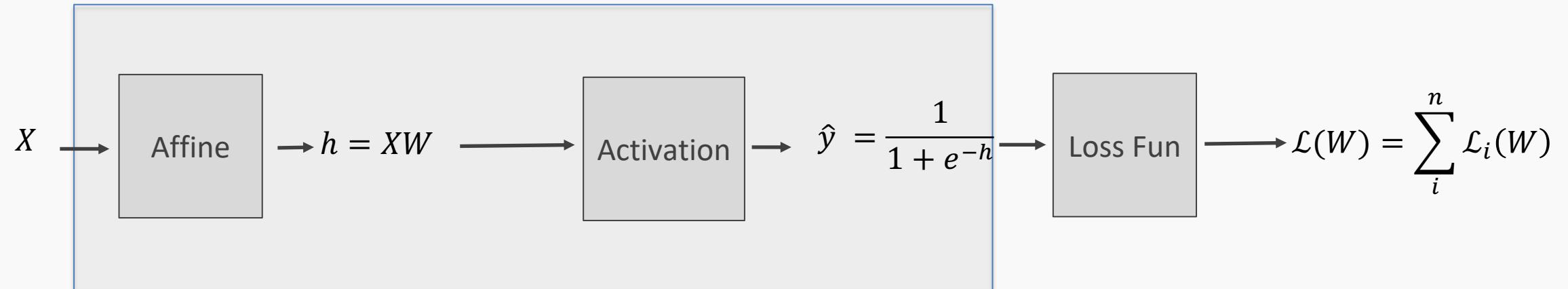
$$X = \begin{bmatrix} 1 & X_{11} & \dots & X_{1p} \\ 1 & \vdots & \dots & \vdots \\ 1 & X_{o1} & \dots & X_{op} \end{bmatrix} \quad W = \begin{bmatrix} b \\ W_1 \\ \vdots \\ W_p \end{bmatrix}$$



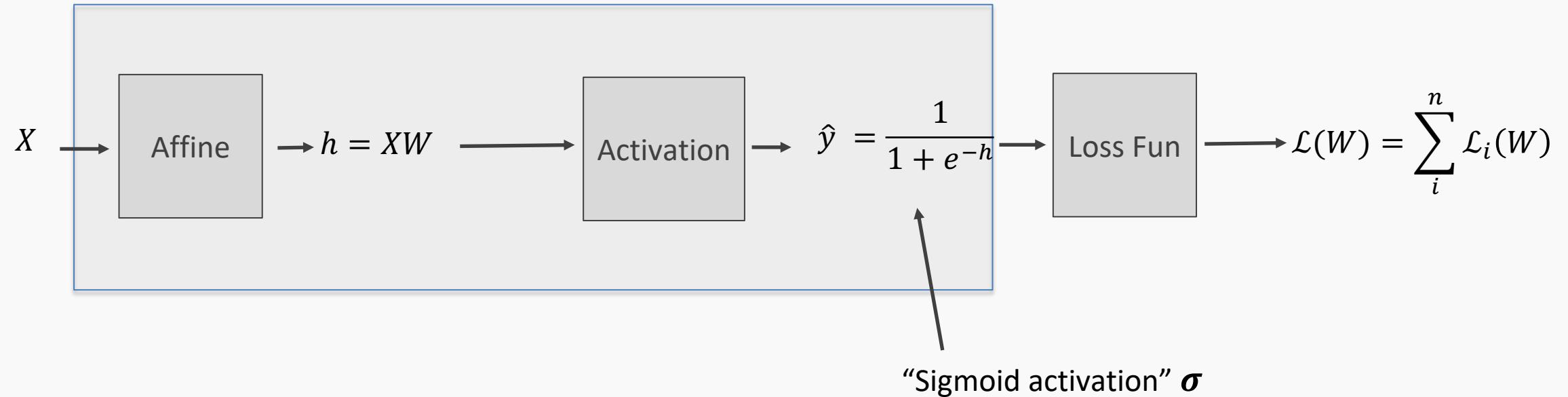
Build our first ANN



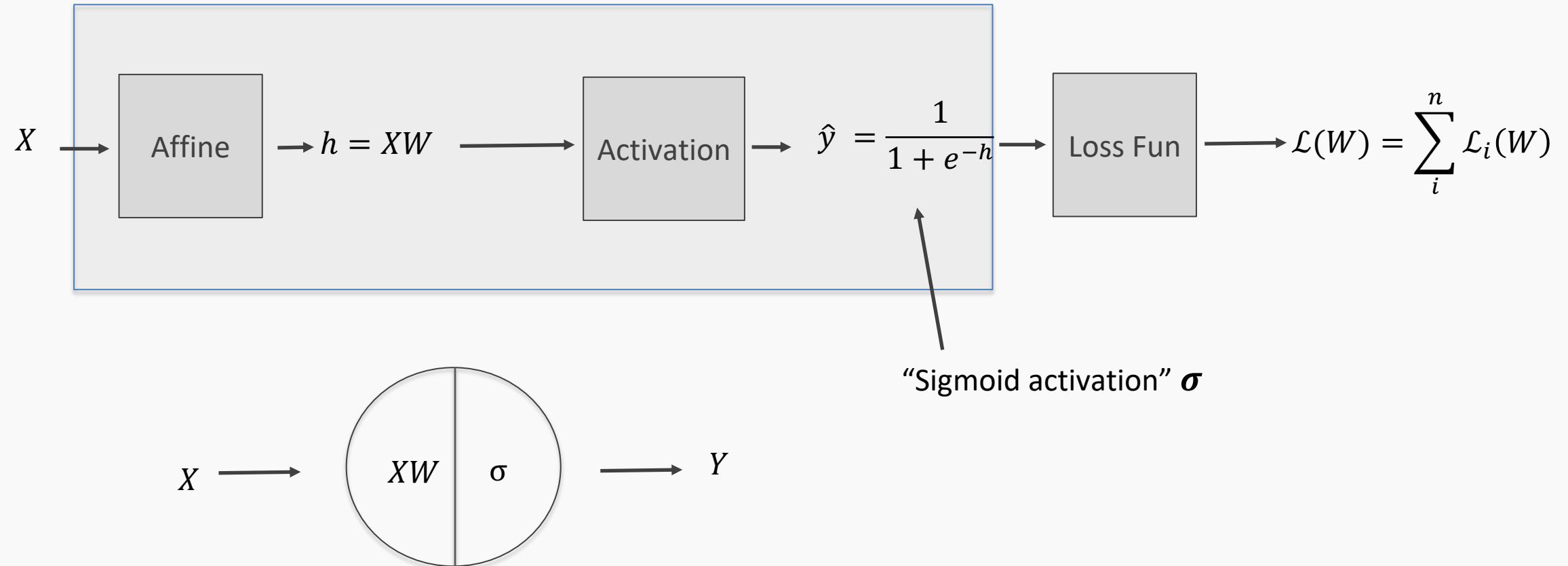
Build our first ANN



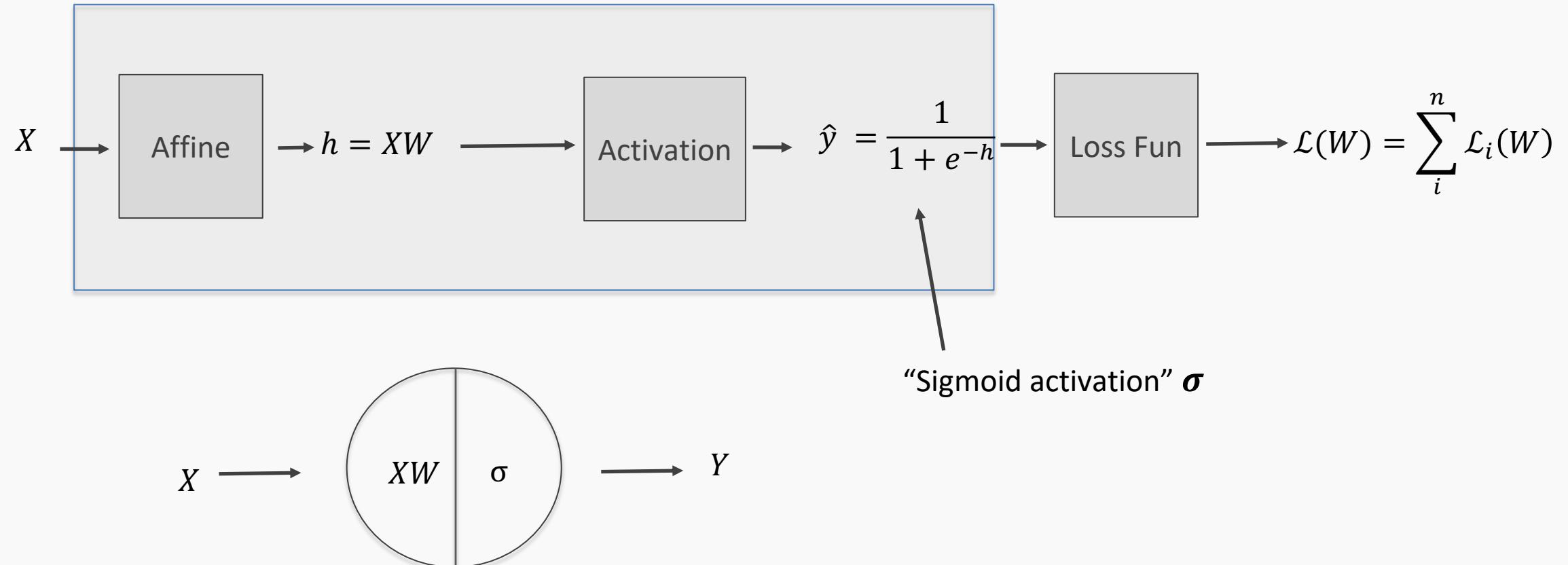
Build our first ANN



Build our first ANN



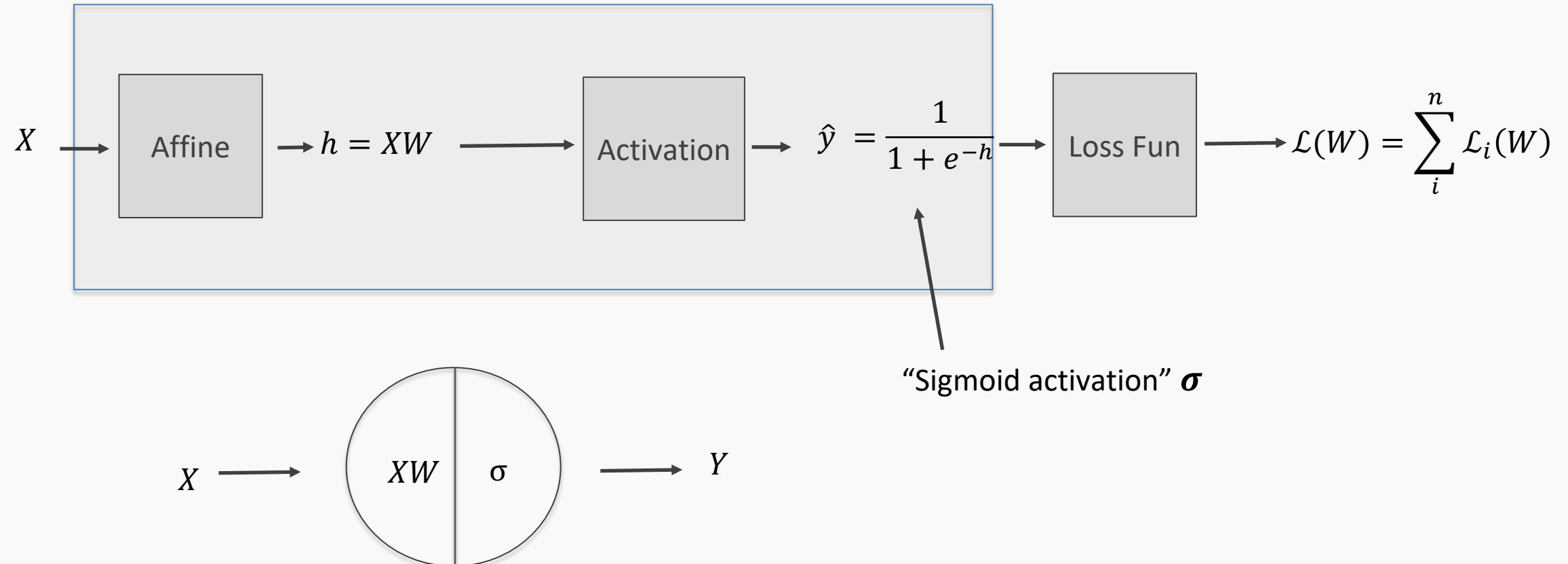
Build our first ANN



Single Neuron Neural “Network”



Build our first ANN

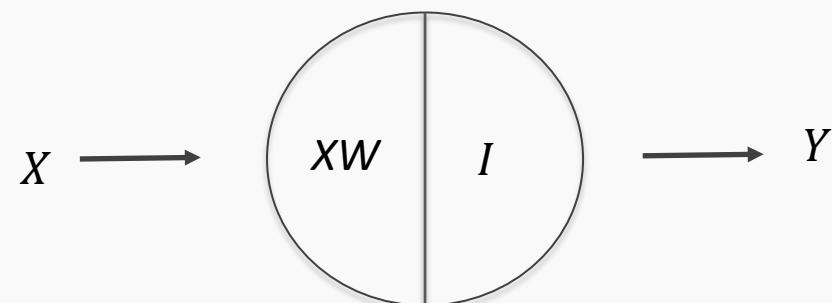
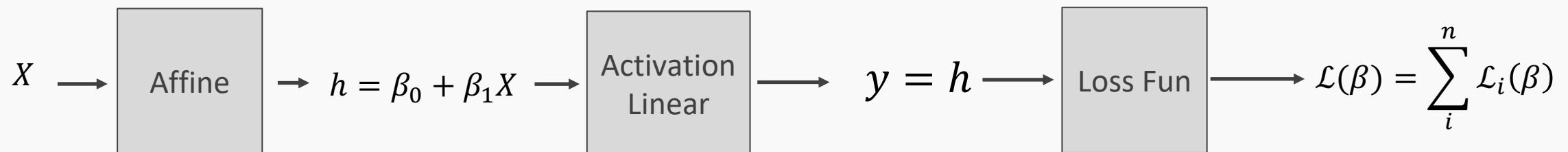


Single Neuron Neural “Network”



Up to this point we just re-branded logistic regression to look like a neuron.

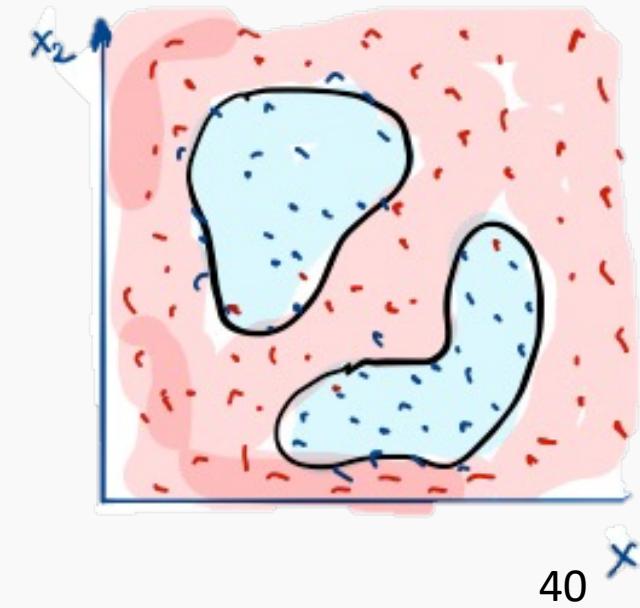
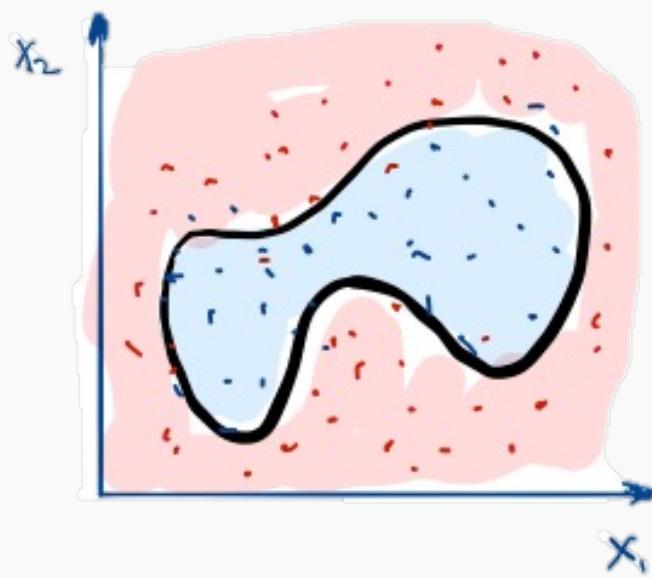
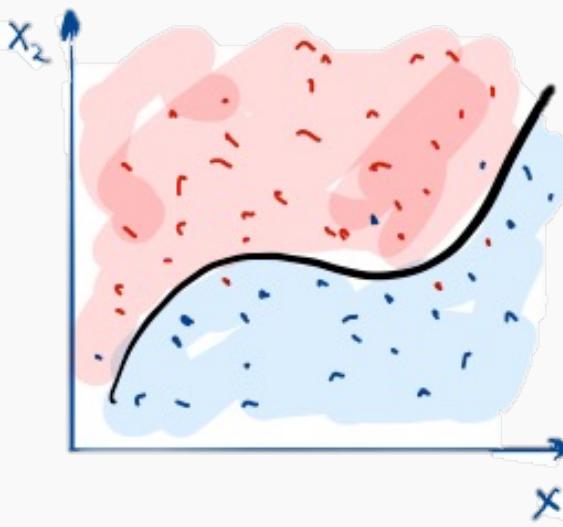
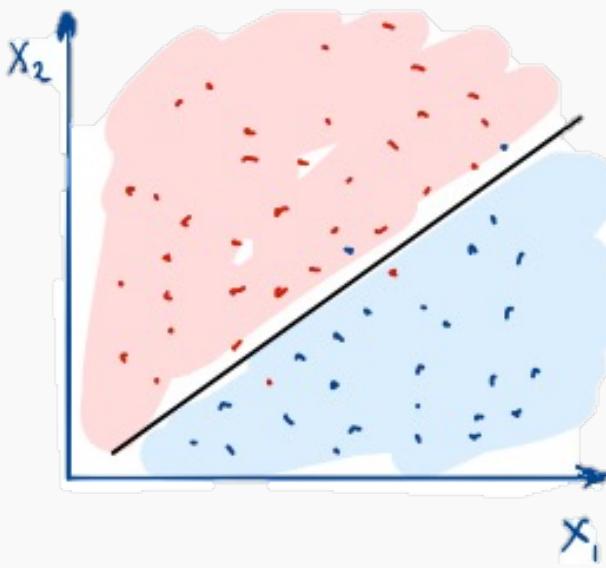
How about linear regression?



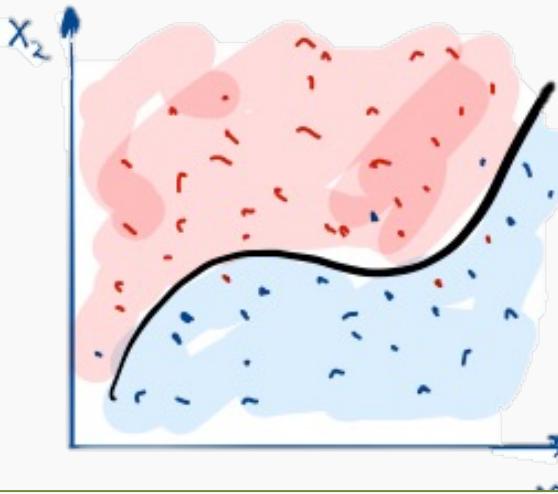
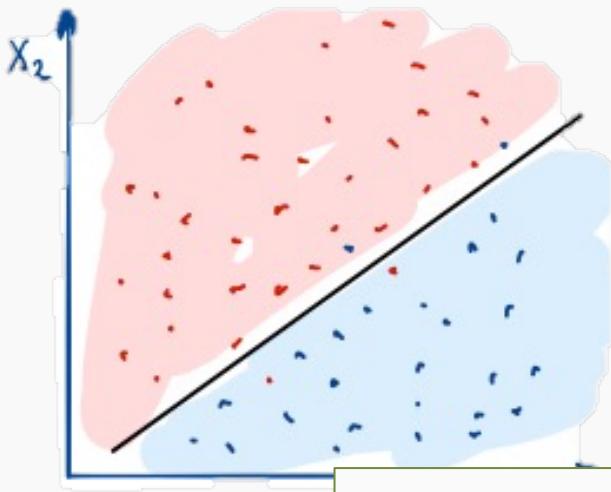
Where I is the identity function



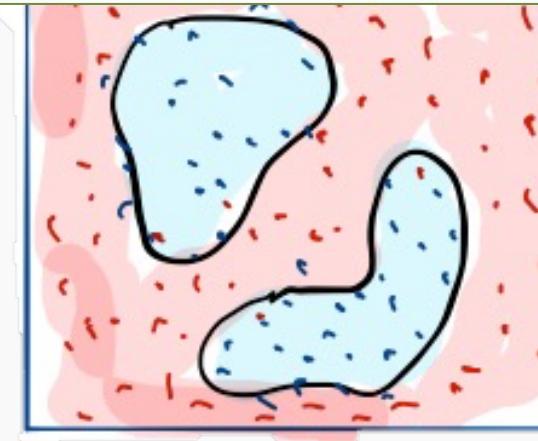
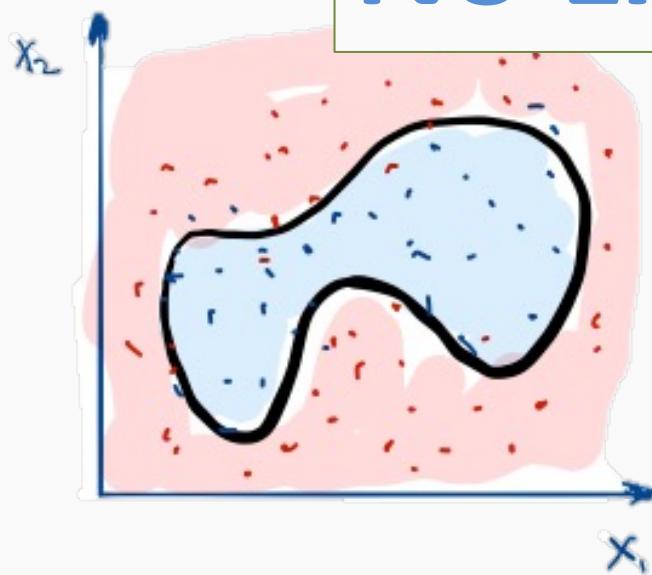
So what's the big deal about Neural Networks?



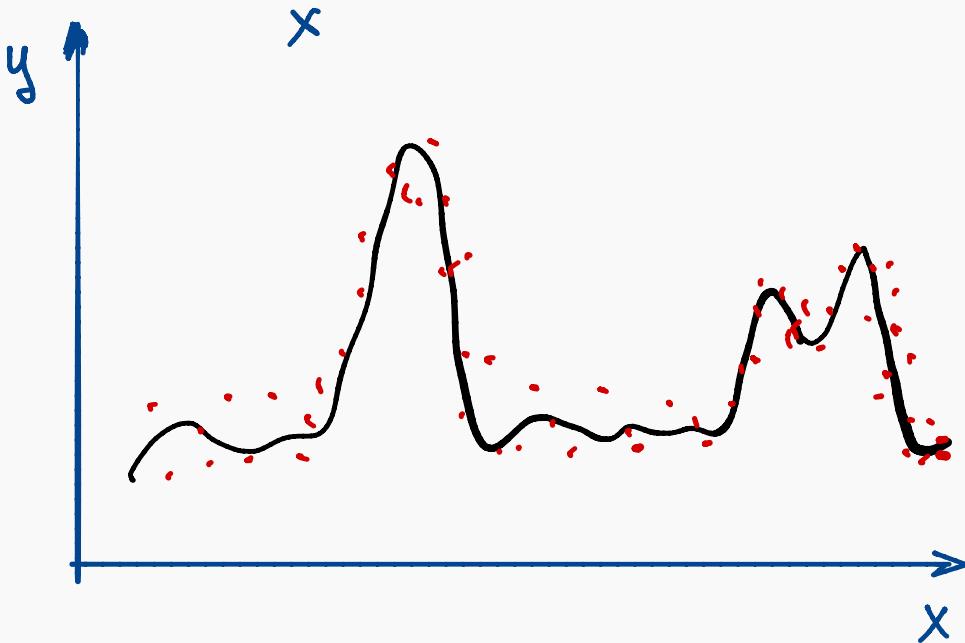
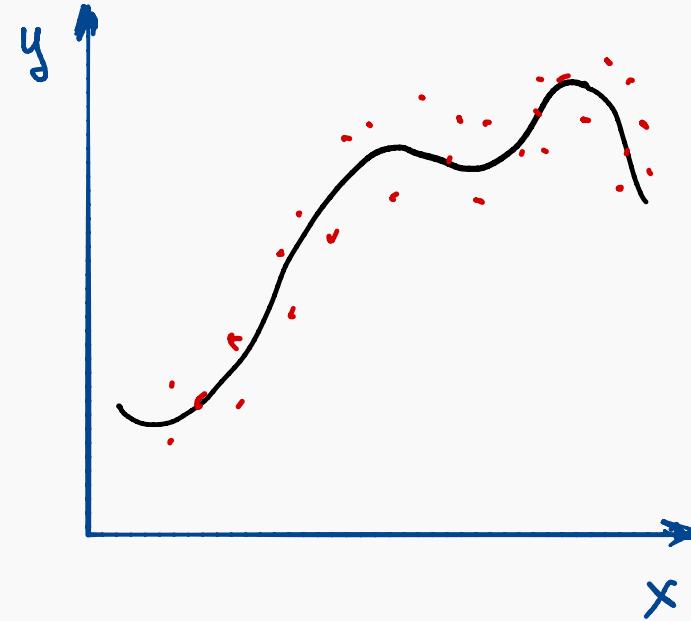
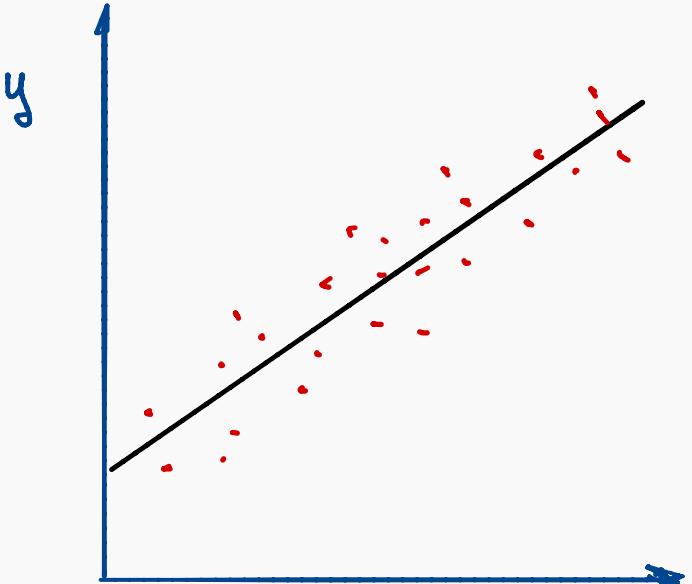
So what's the big deal about Neural Networks?



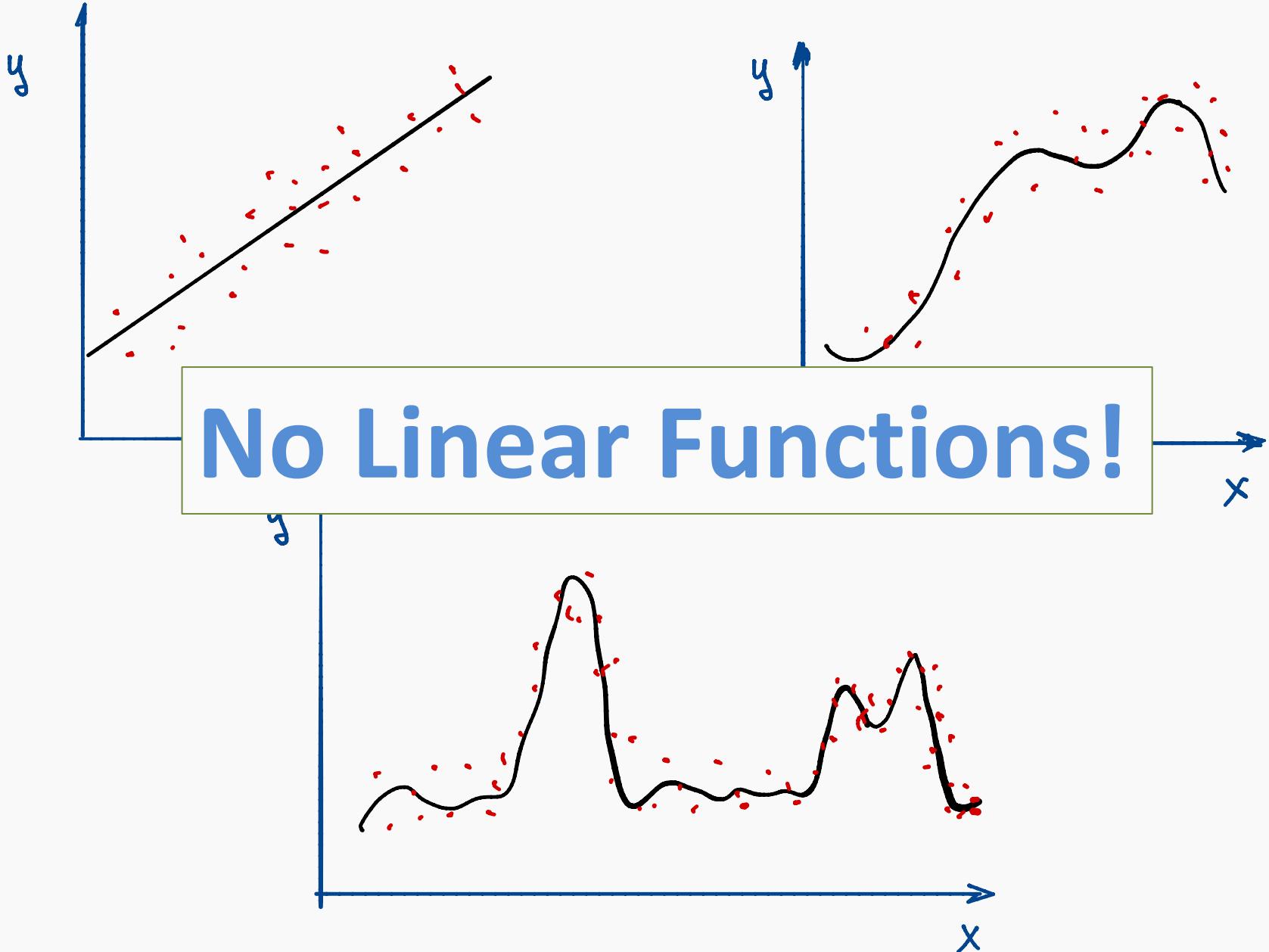
No Linear Functions!



For regression?



For regression?



Outline

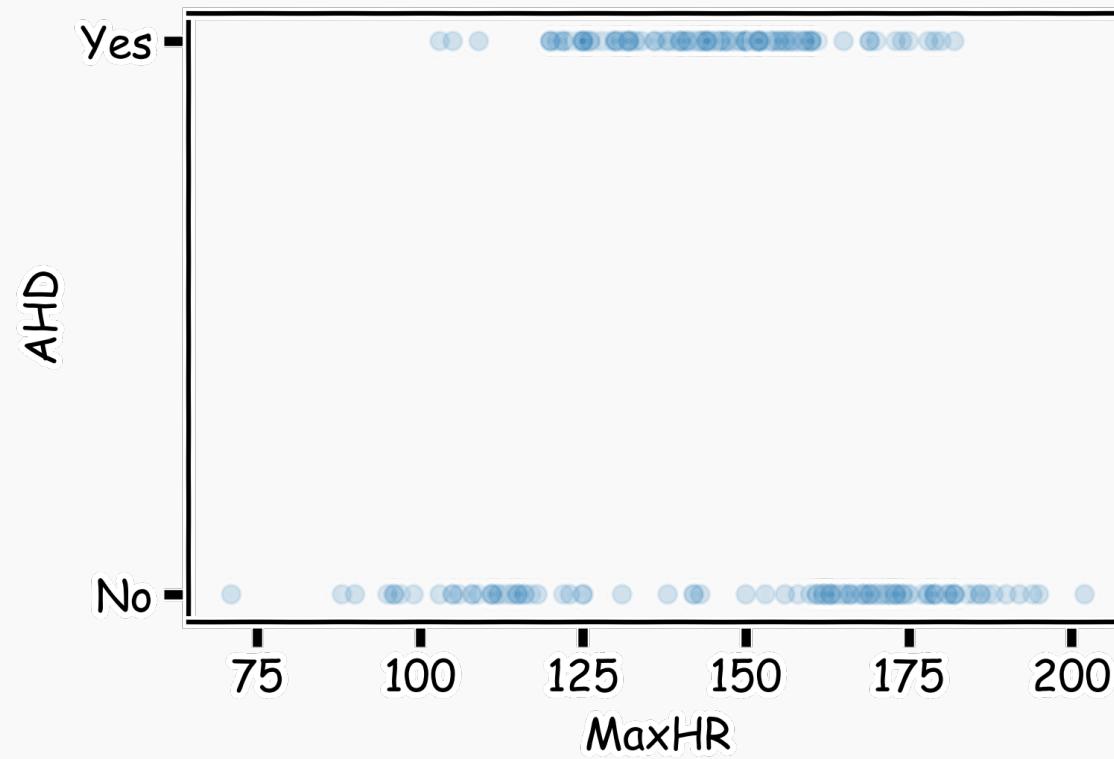
1. Introduction to Artificial Neural Networks
2. Review of Classification and Logistic Regression
3. Single Neuron Network ('Perceptron')
4. **Multi-Layer Perceptron (MLP)**

Example Using Heart Data

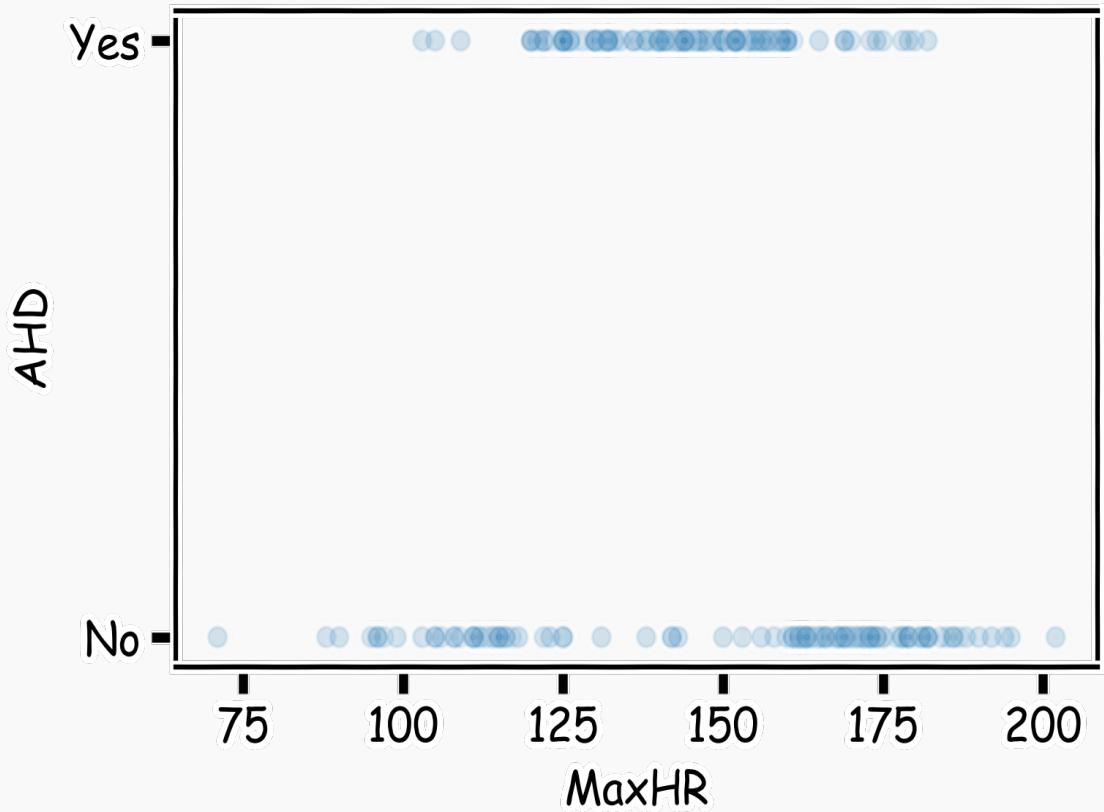
Slightly modified data to illustrate concepts.

Example Using Heart Data

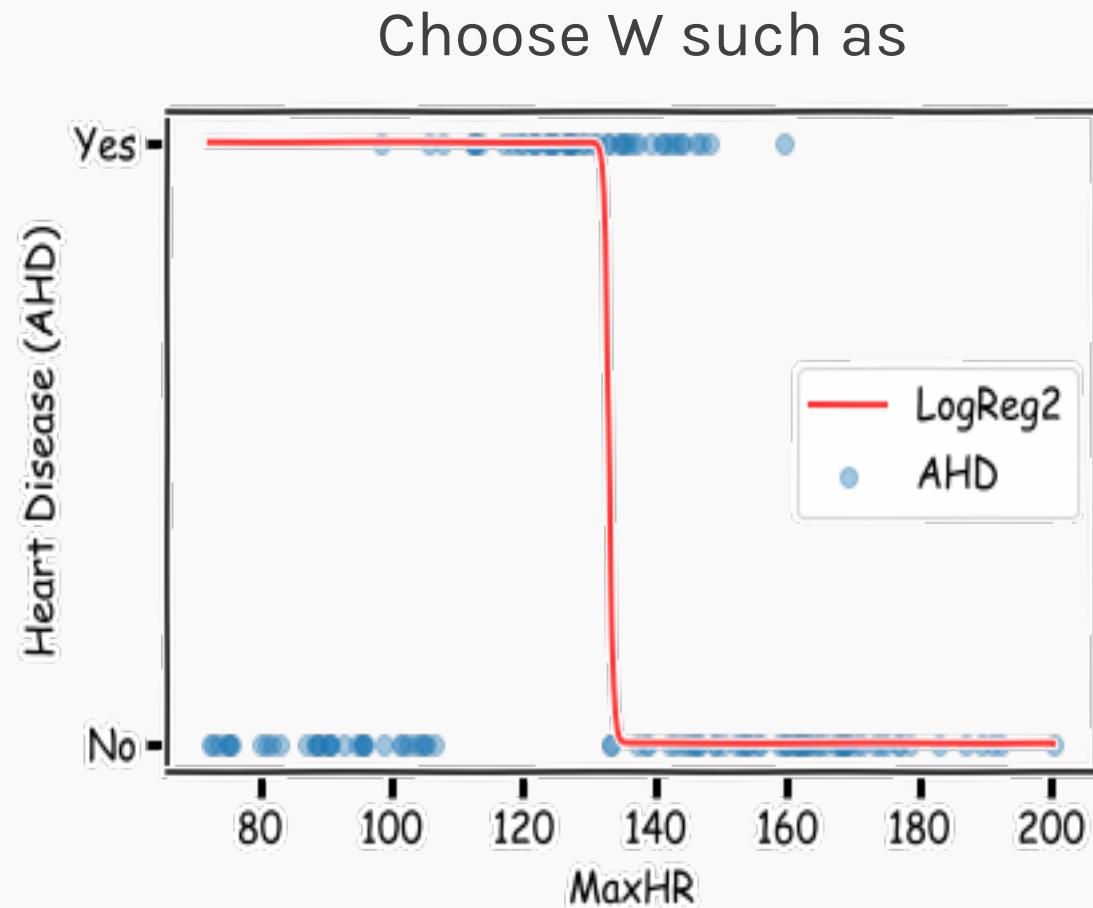
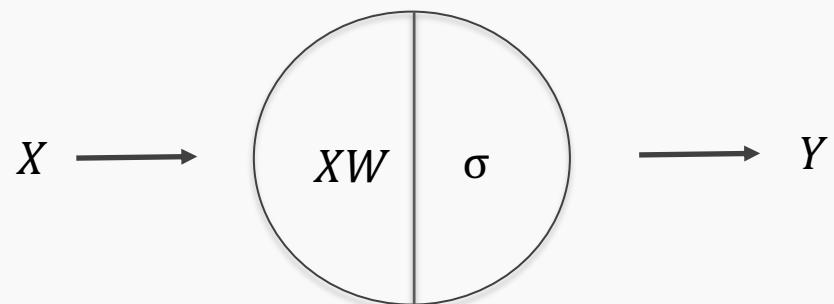
Slightly modified data to illustrate concepts.



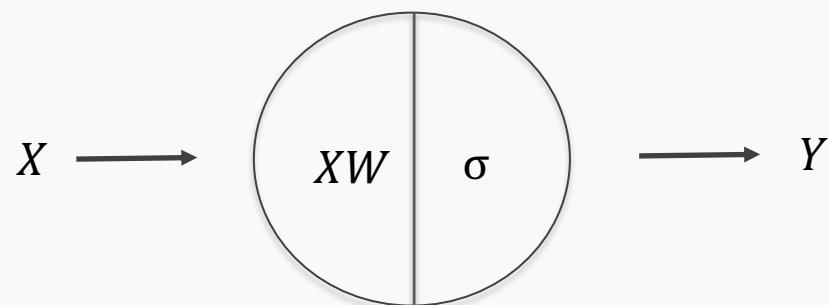
Example Using Heart Data



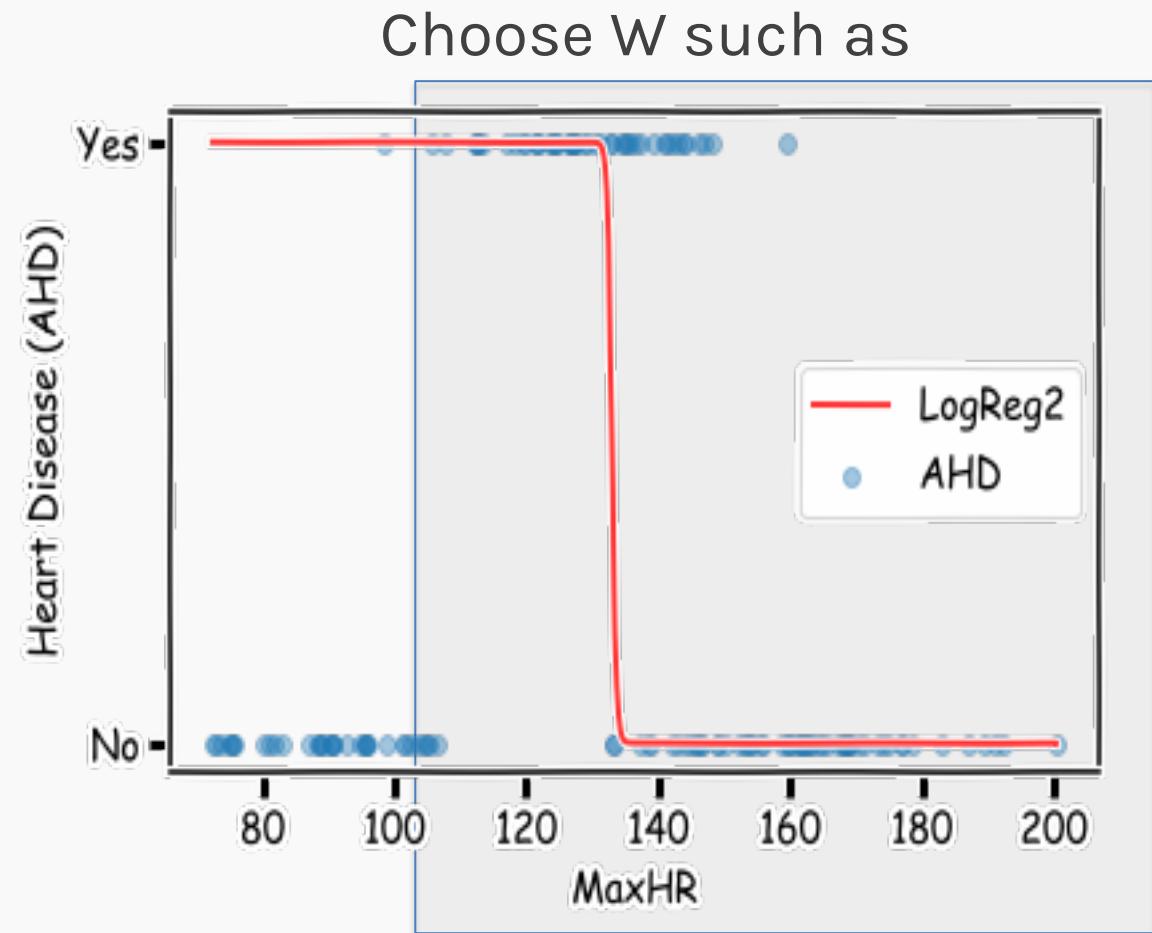
Example Using Heart Data



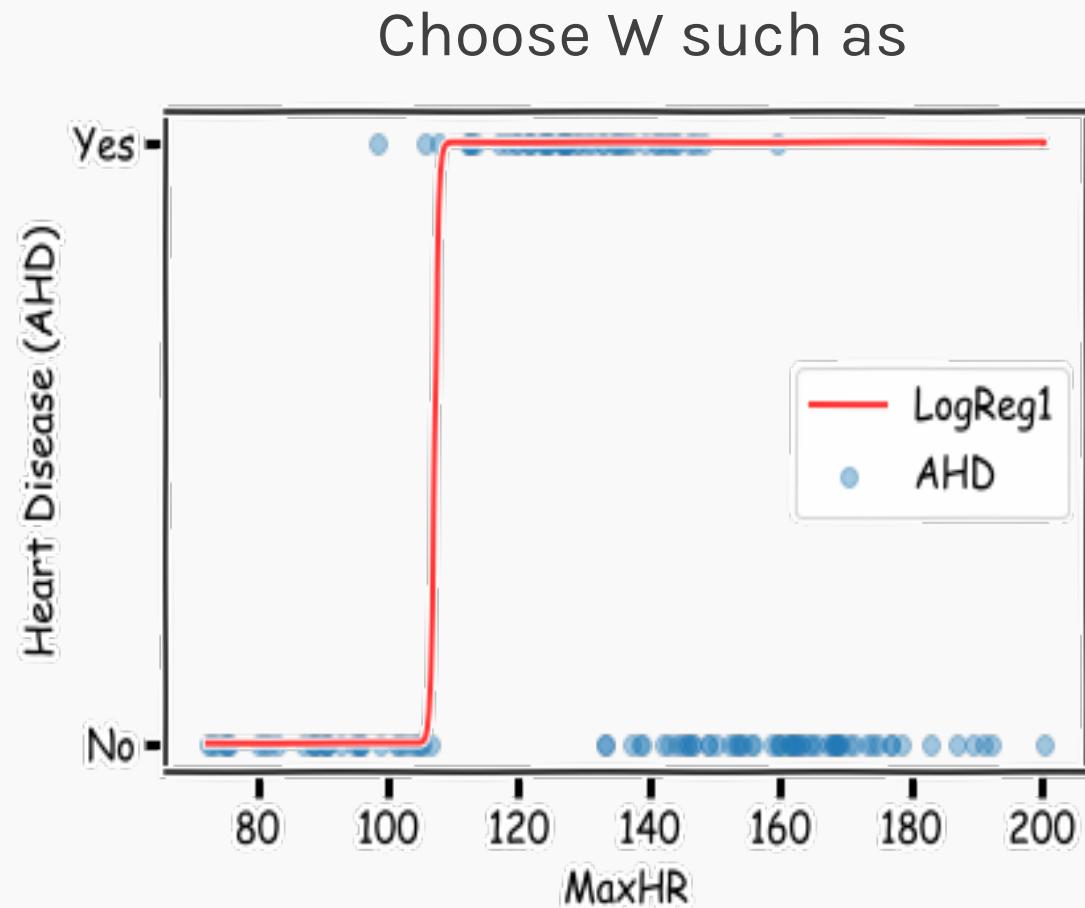
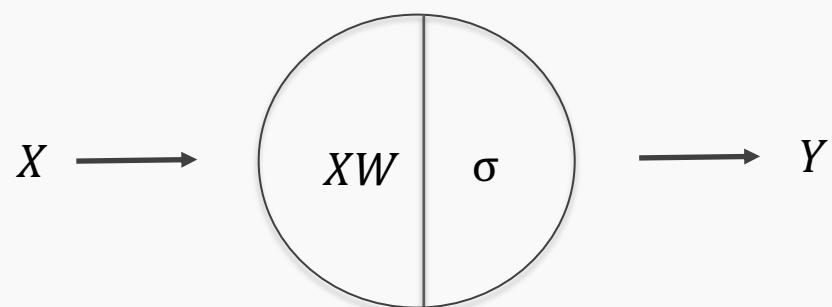
Example Using Heart Data



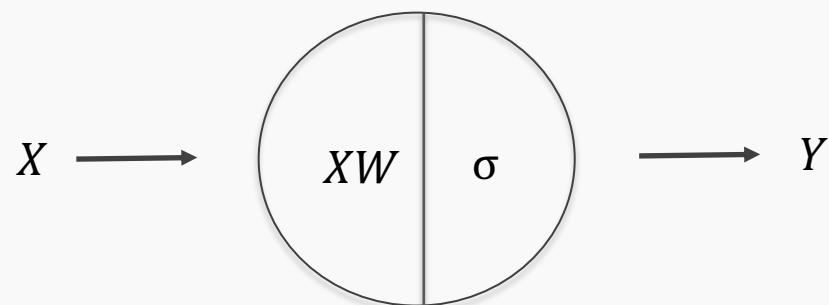
Right part of data is fitted well



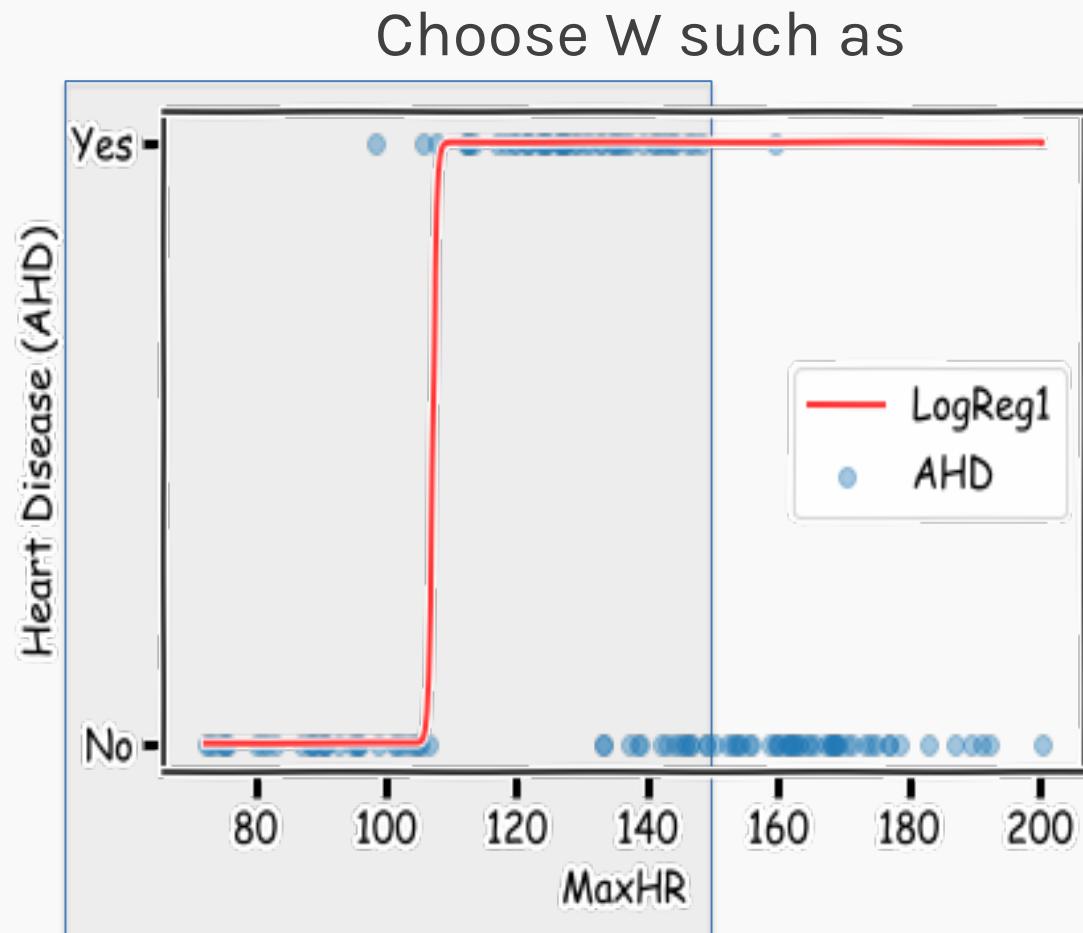
Example Using Heart Data



Example Using Heart Data

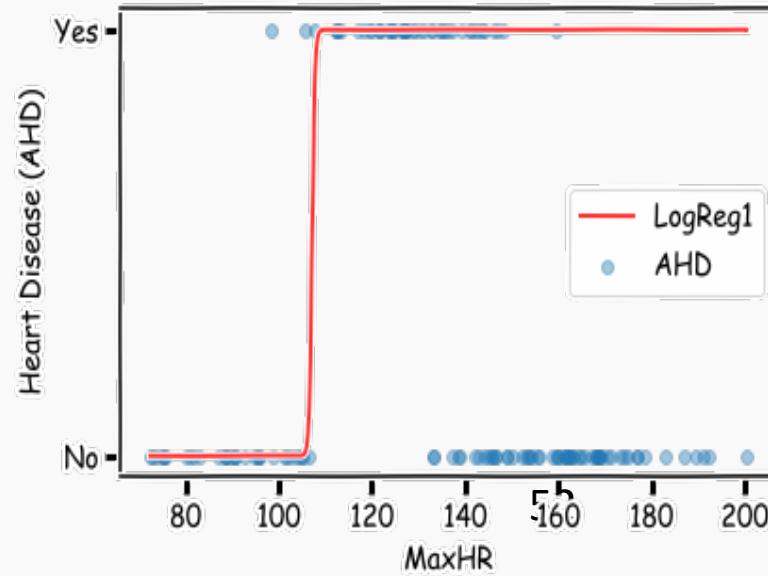
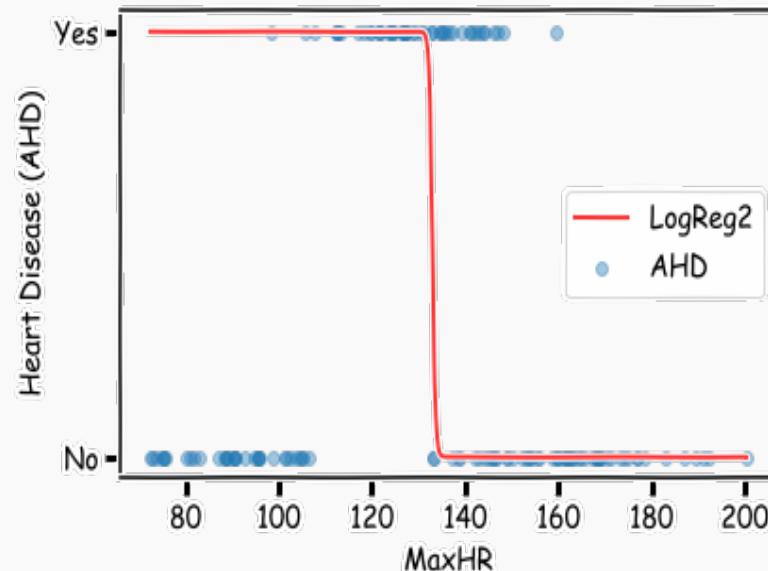
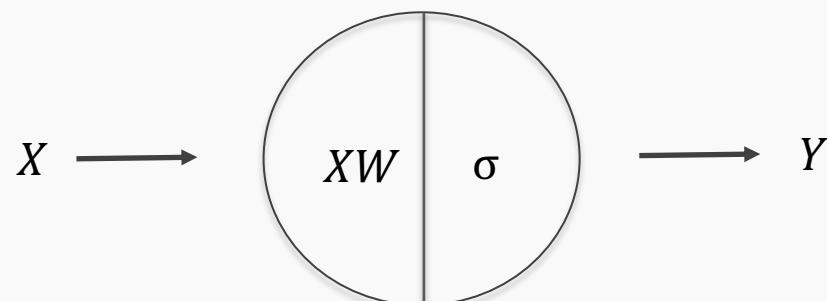
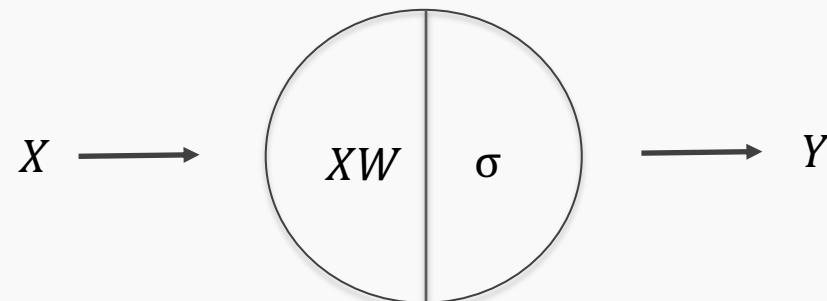


Left part of data is fitted well

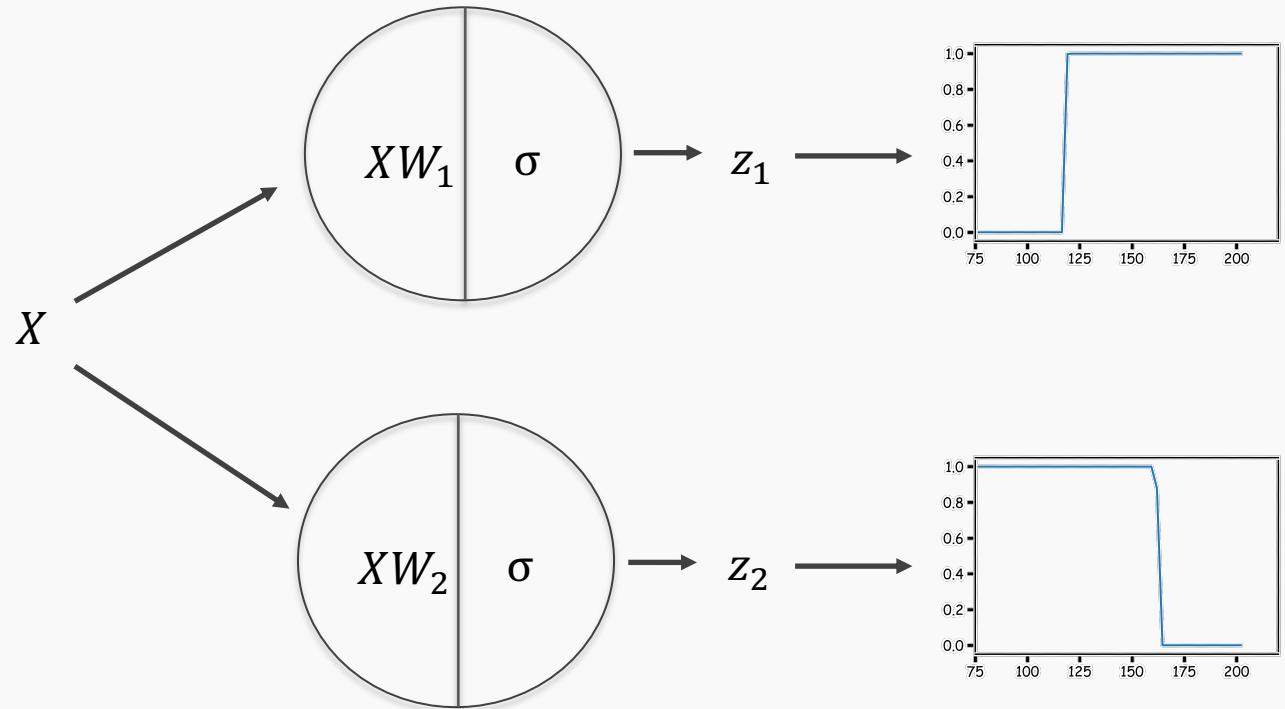


Example Using Heart Data

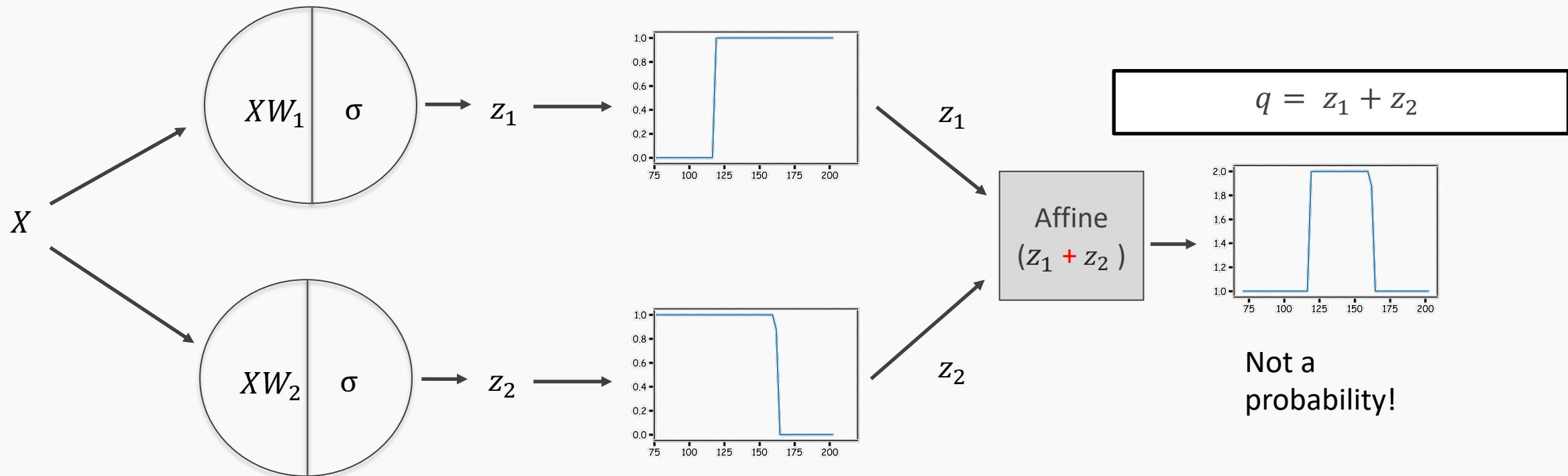
Two regions, two nodes



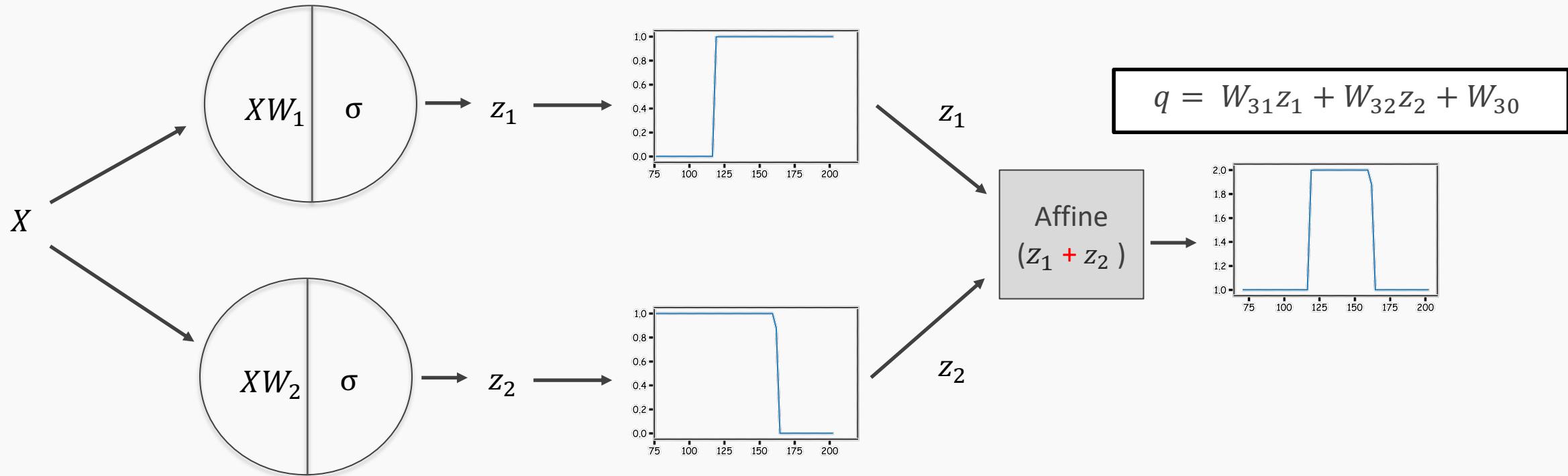
Pavlos game #232



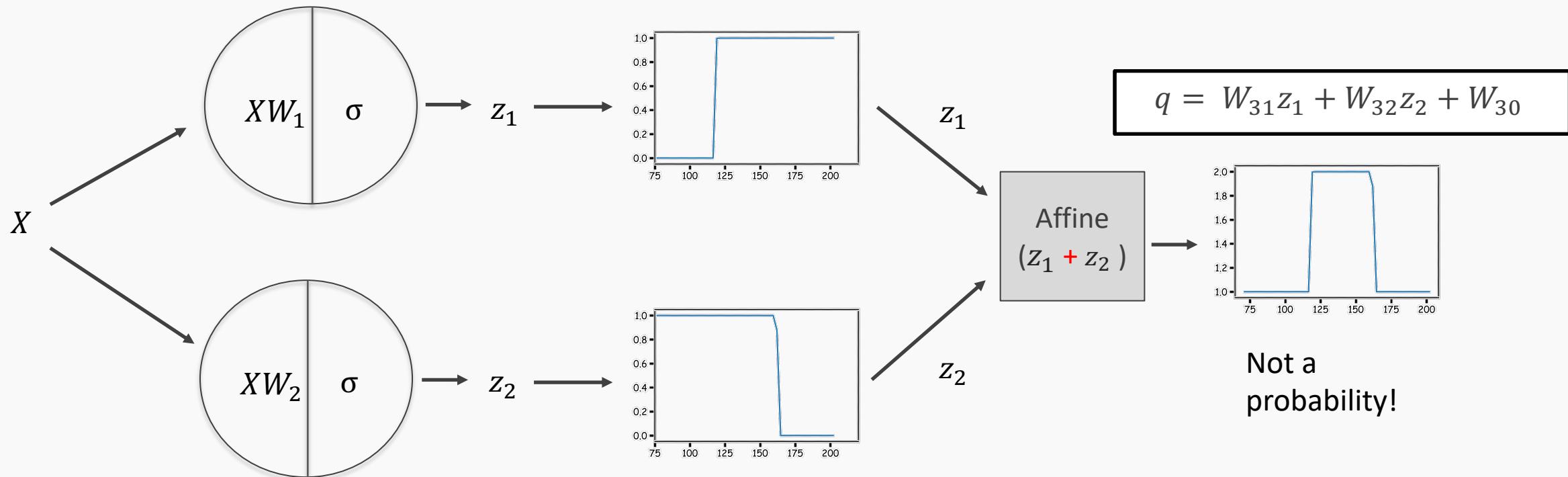
Pavlos game #232



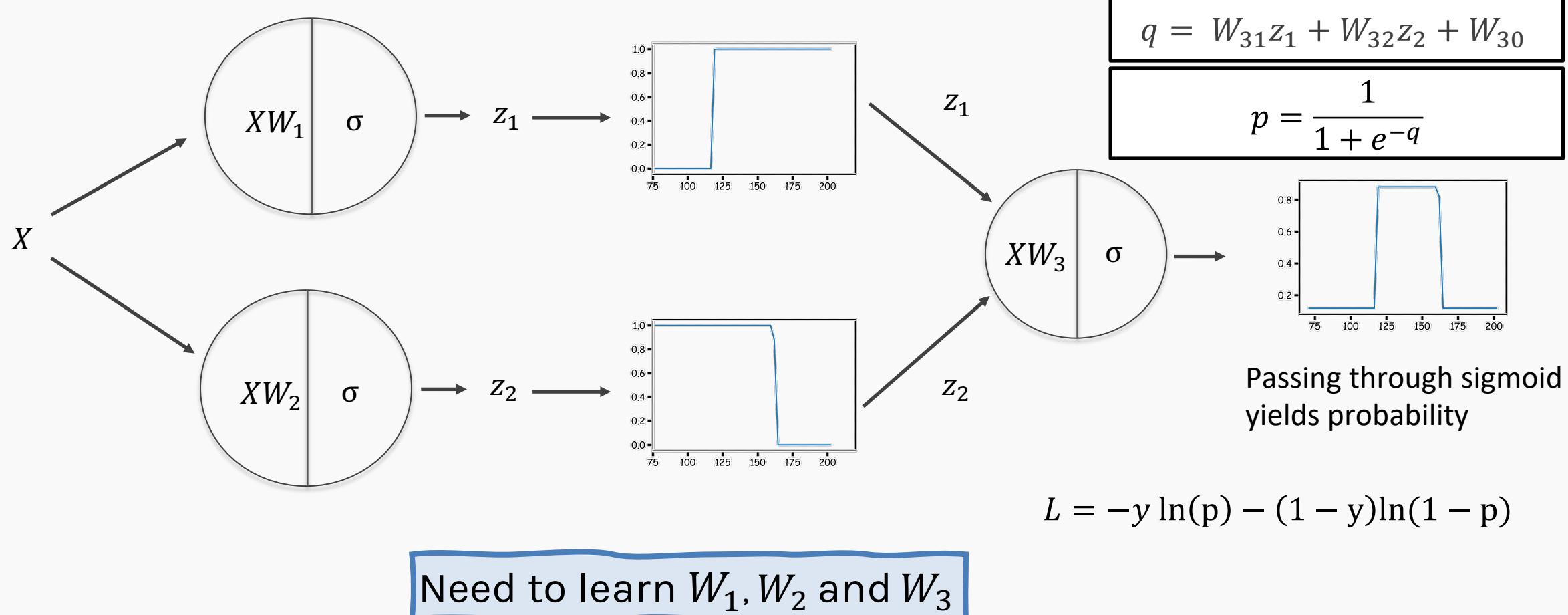
Pavlos game #232



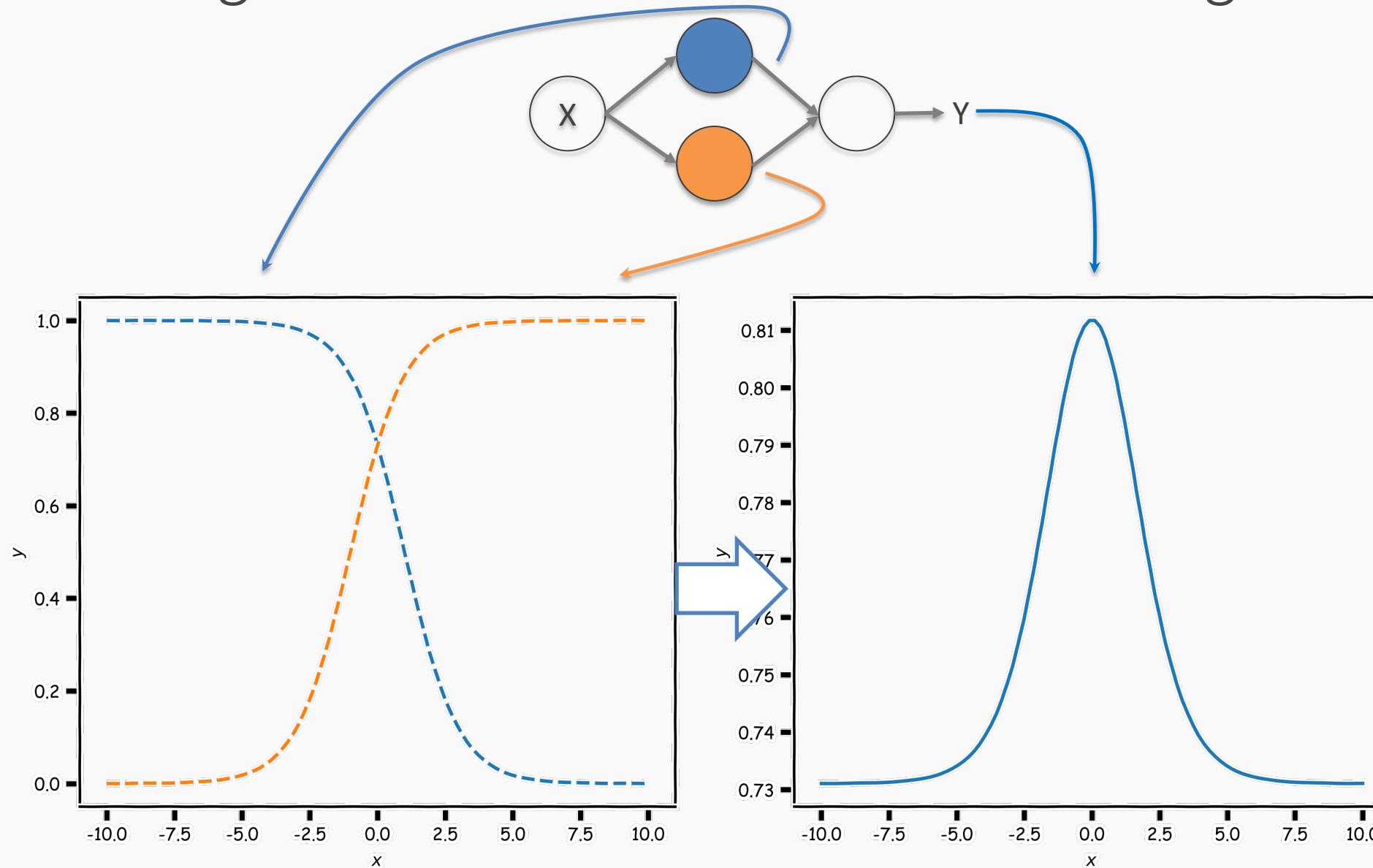
Pavlos game #232



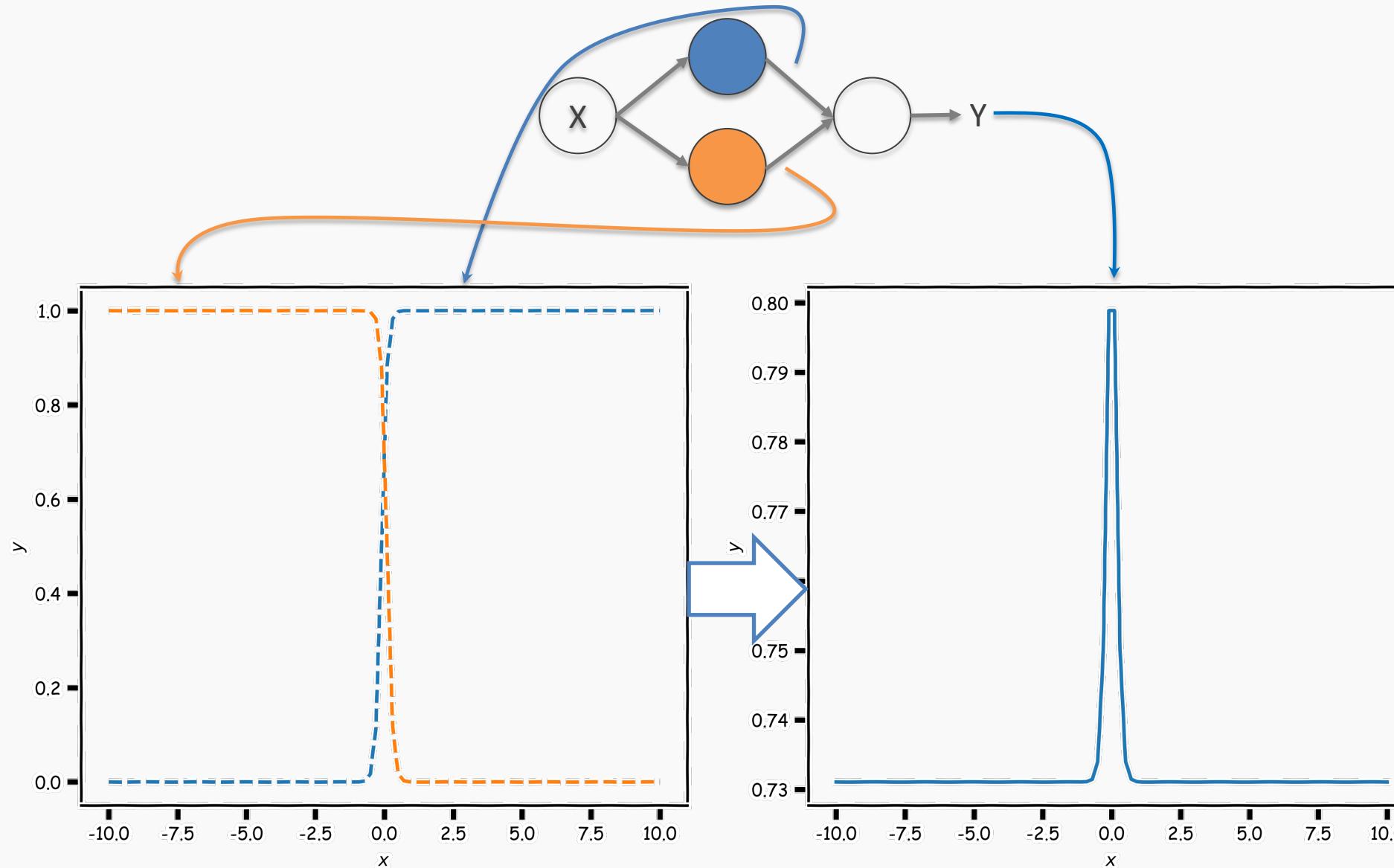
Combining Neurons ...



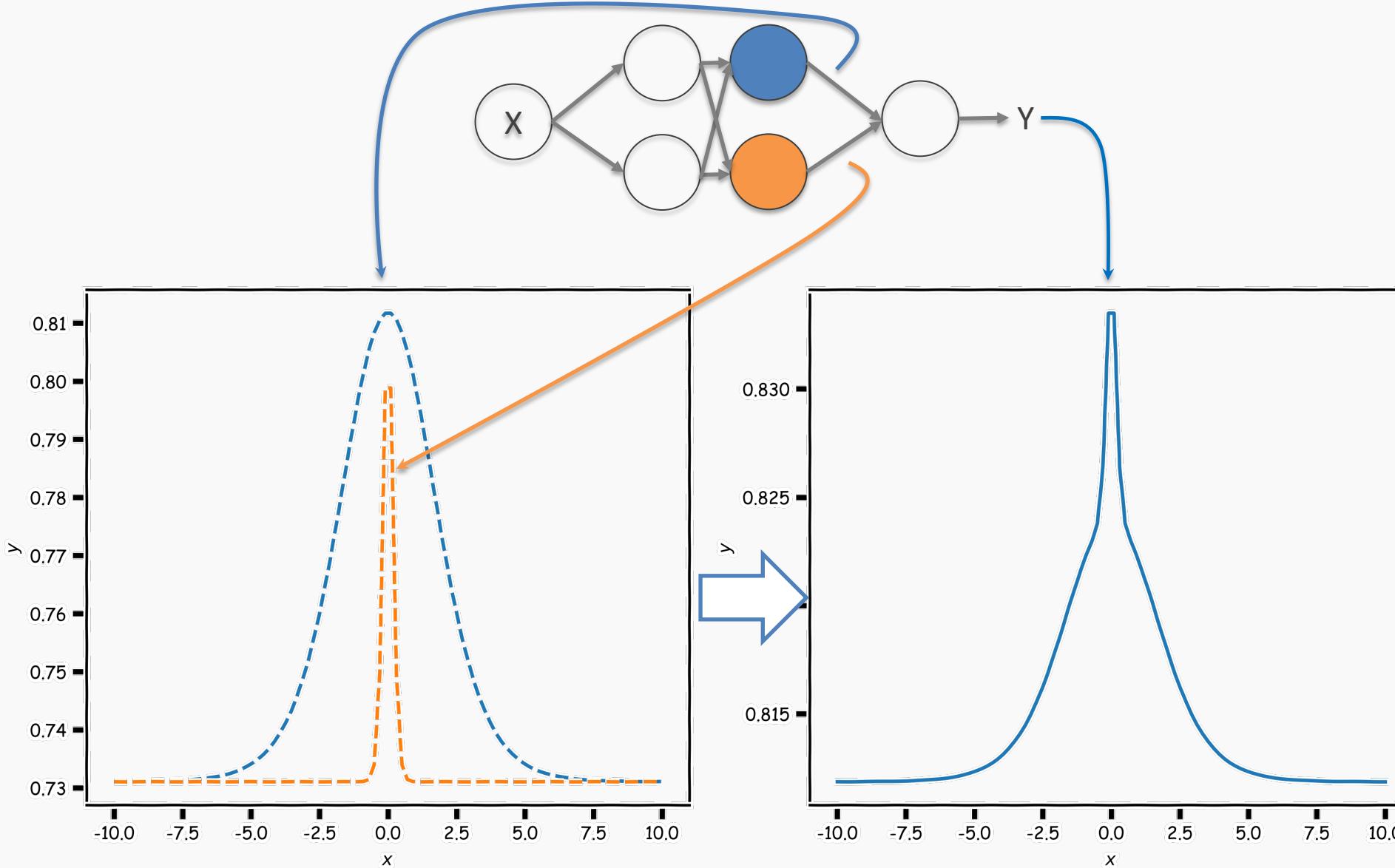
Combining neurons allows us to model interesting functions



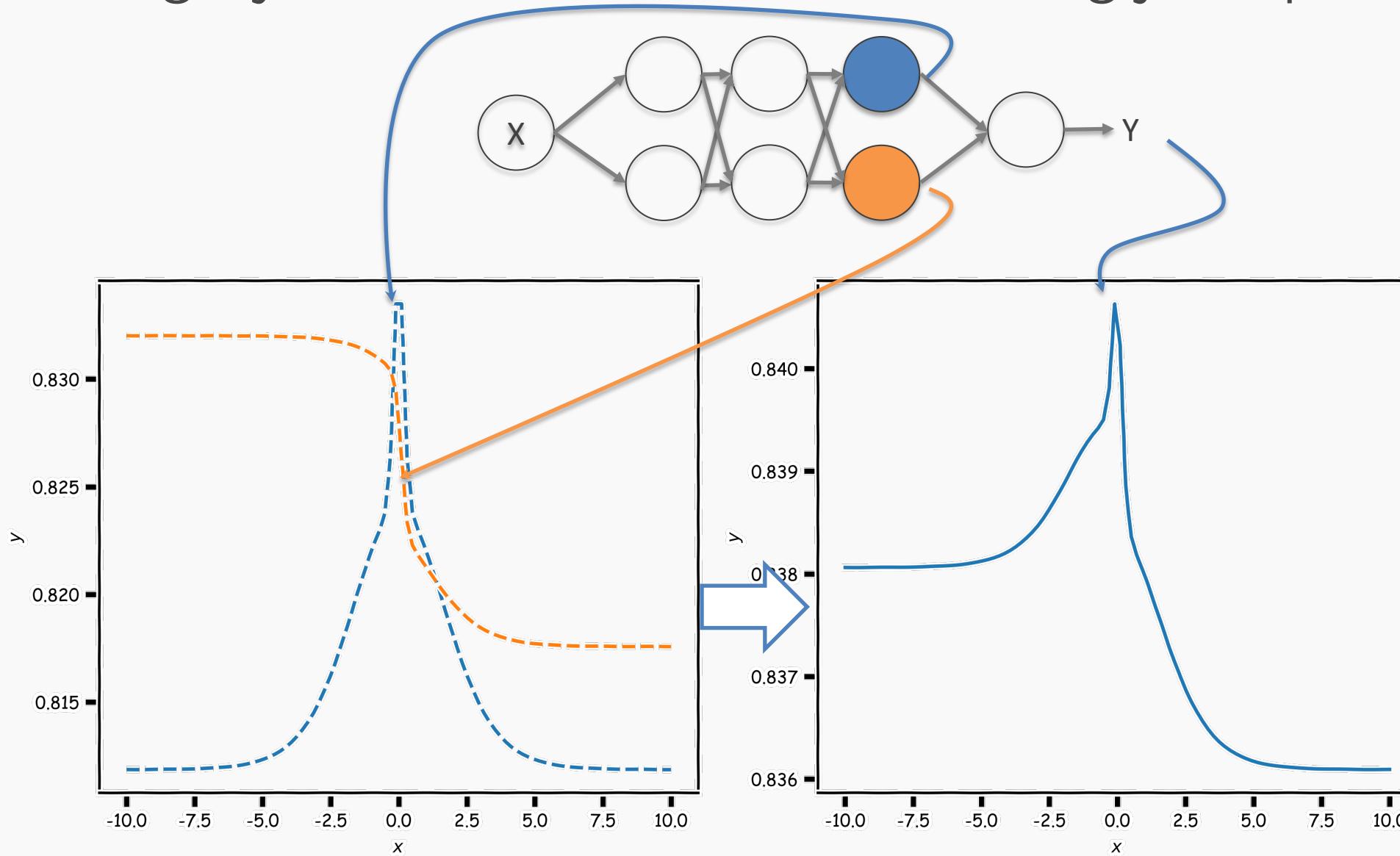
Different weights change the shape and position



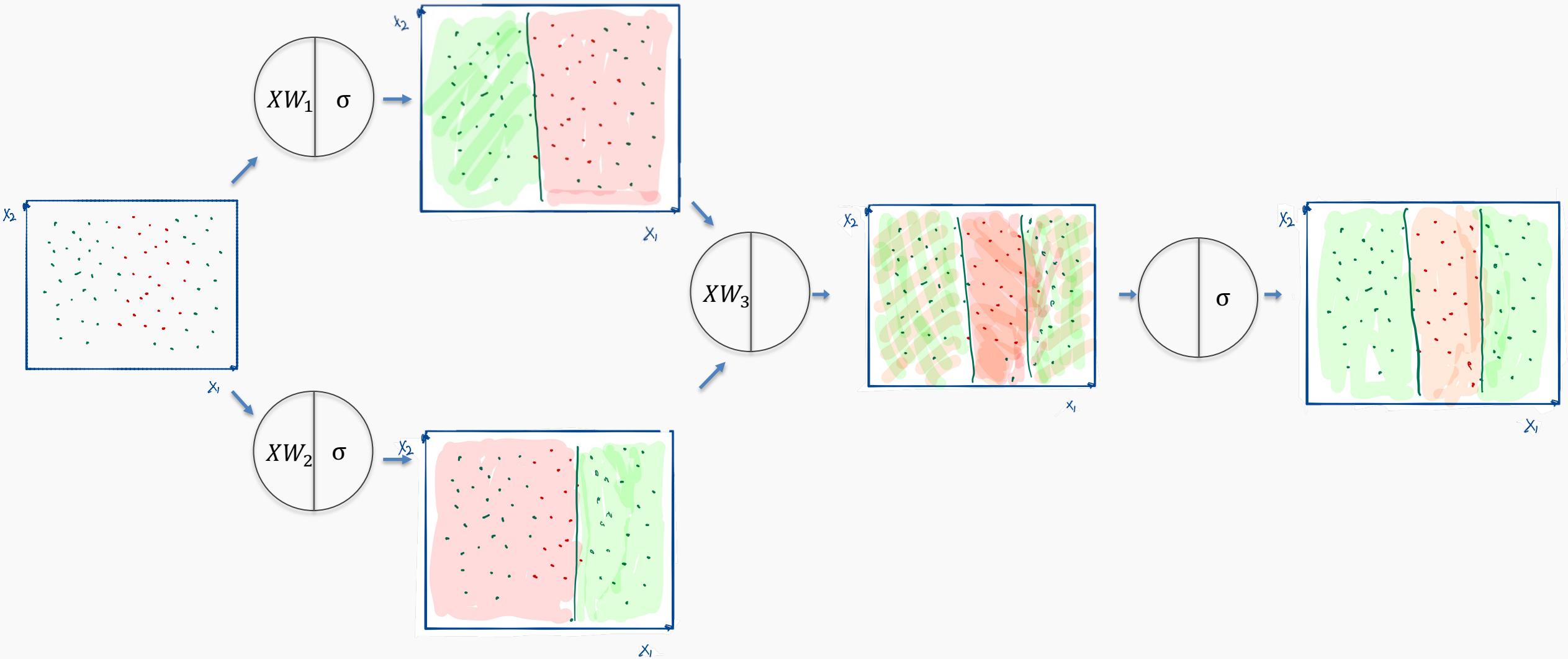
Neural networks can model any reasonable function



Adding layers allows us to model increasingly complex functions



For 2-D input the same idea applies.



Summary

So far:

- A single neuron can be a **logistic regression** or linear unit. We will soon see other choices.
- A neural network is a **combination** of logistic regression (or other types) units.
- A neural network can **approximate** non-linear functions either for regression or classification.

Summary

So far:

- A single neuron can be a **logistic regression** or linear unit. We will soon see other choices.
- A neural network is a **combination** of logistic regression (or other types) units.
- A neural network can **approximate** non-linear functions either for regression or classification.

Next:

- What kind of **activations**, how many **neurons**, how many **layers**, how to construct the **output** unit and what **loss** functions are appropriate?



Summary

So far:

- A single neuron can be a **logistic regression** or linear unit. We will soon see other choices.
- A neural network is a **combination** of logistic regression (or other types) units.
- A neural network can **approximate** non-linear functions either for regression or classification.

Next:

- What kind of **activations**, how many **neurons**, how many **layers**, how to construct the **output** unit and what **loss** functions are appropriate?

Following two lectures on NN:

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

