Introduction to ANN and Perceptron

CS1090B Data Science II – Spring 2025

Pavlos Protopapas, Natesh Pillai, and Chris Gumb

Ahsan Cheema
Nangaparbat Basecamp

Outline

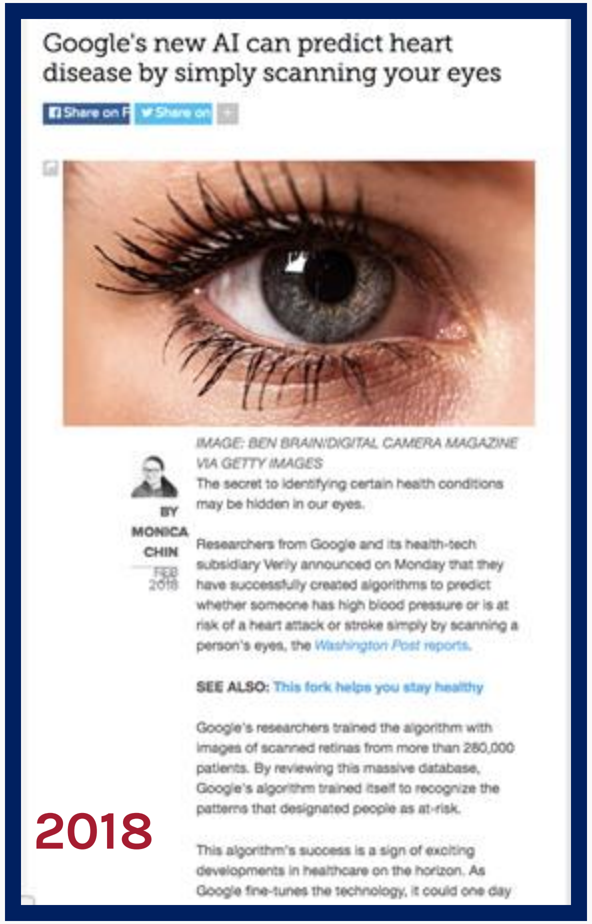
- Introduction
- Review of basic concepts
- Linear to Logistic Regression
- Logistic Regression to ANN
- Perceptron - Single neuron network
- Multi-Layer Perceptron (MLP)

Outline

- **Introduction**
- Review of basic concepts
- Linear to Logistic Regression
- Logistic Regression to ANN
- Perceptron - Single neuron network
- Multi-Layer Perceptron (MLP)

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

BY MONICA CHIN

FEB 2018

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

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.

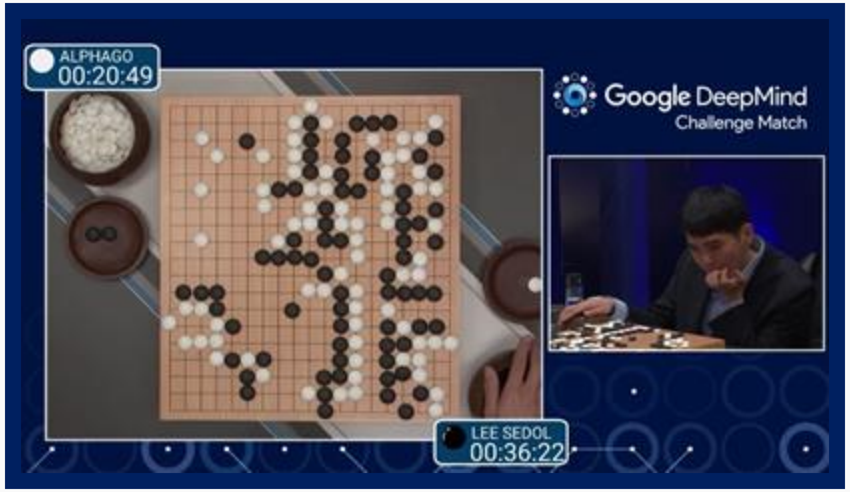
SEE ALSO: This fork helps you stay healthy

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

2018

Game strategy



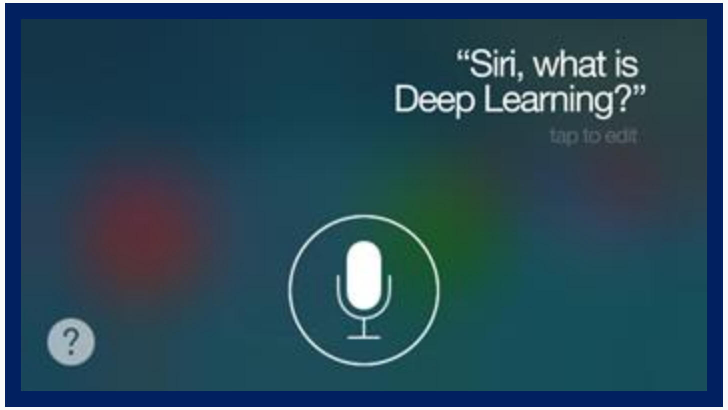

DeepMind

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

2017

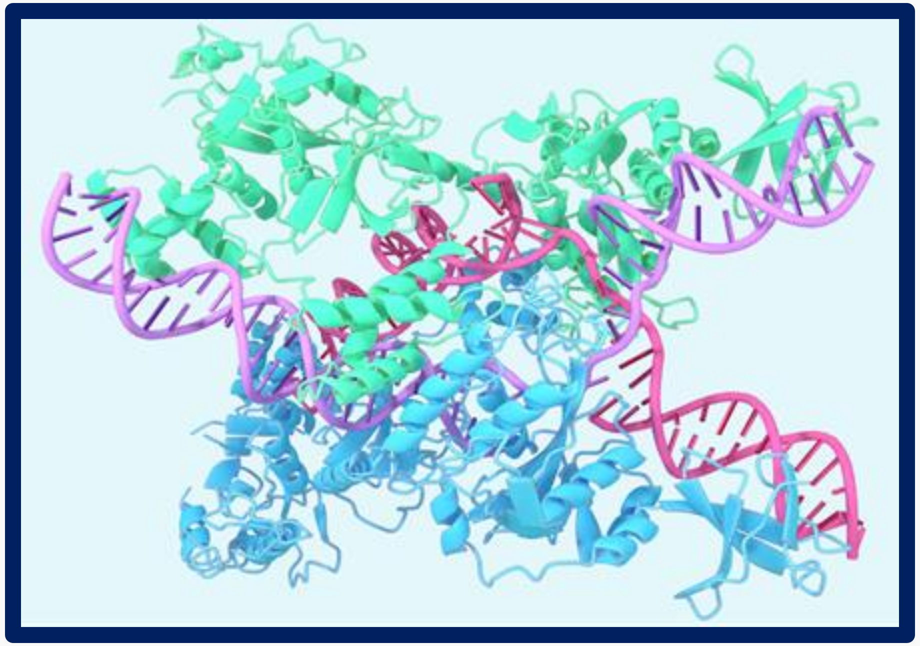
Natural Language Processing



2011

Recent developments

Protein folding



AlphaFold, a DeepMind AI, revolutionized biochemistry by solving the long-standing protein folding problem.

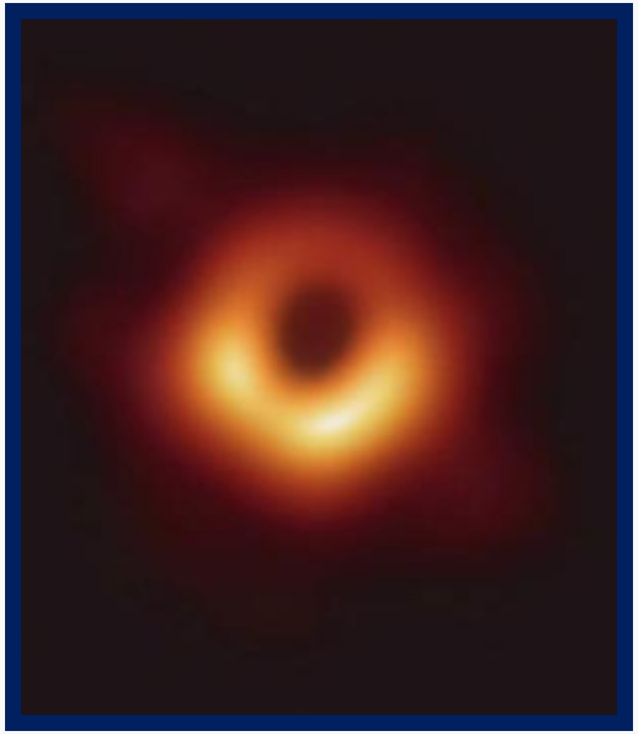
2020

Autonomous cars

AI Detecting objects to assist with autonomous driving.

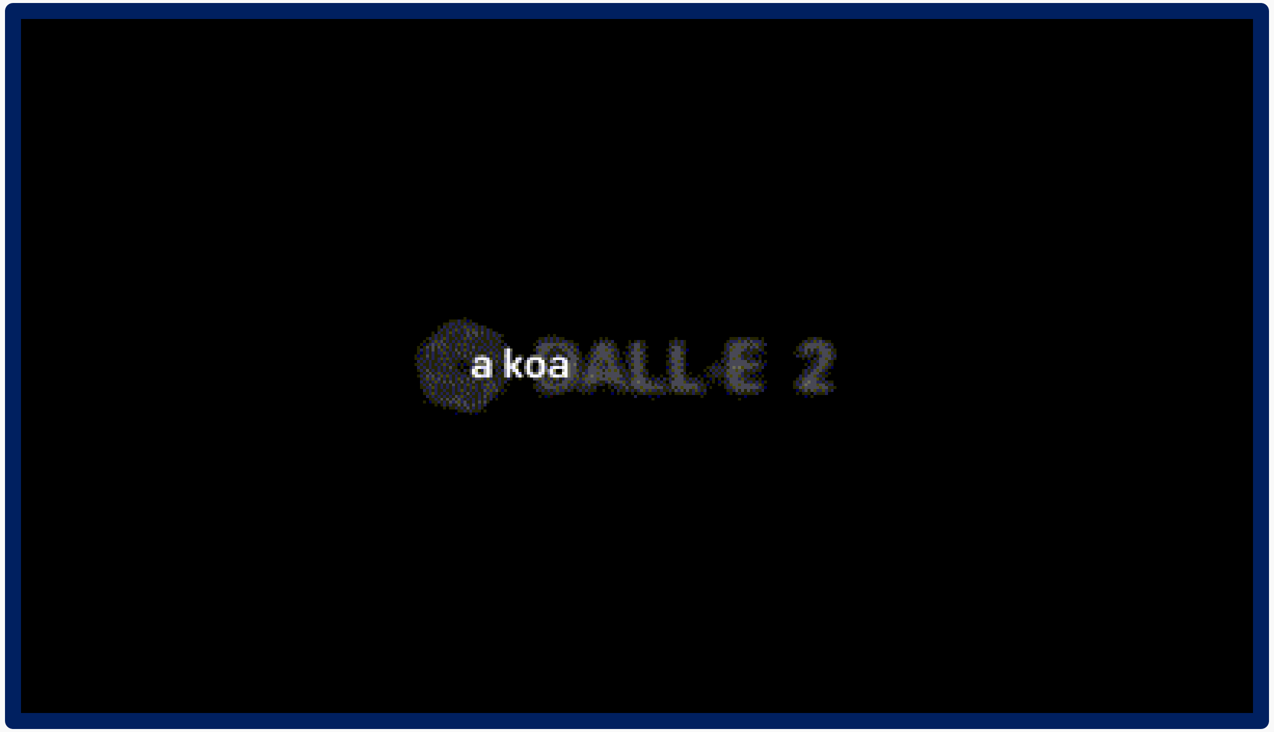
Recent developments

Image Reconstruction from Sparse Frequency Measurements



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

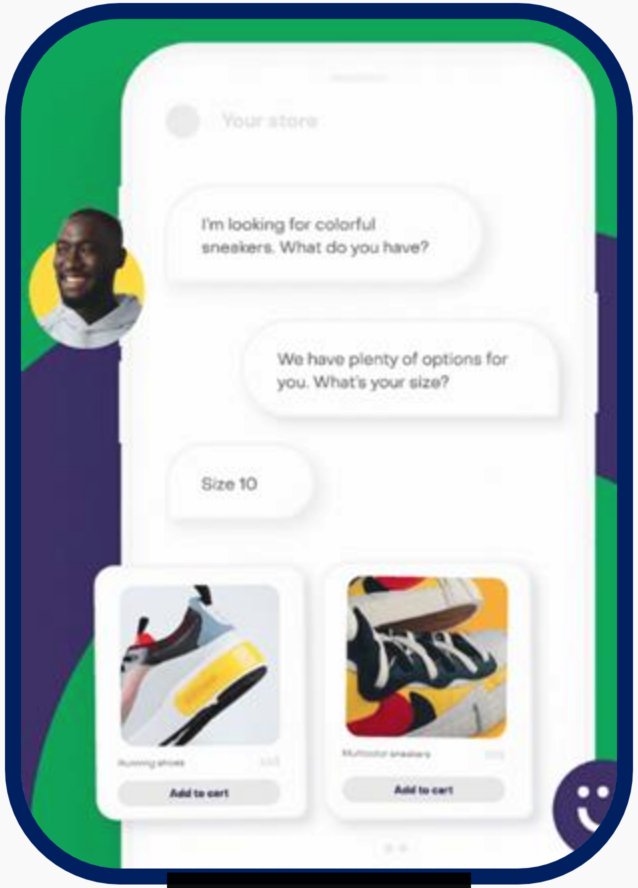
Text to Image Generation



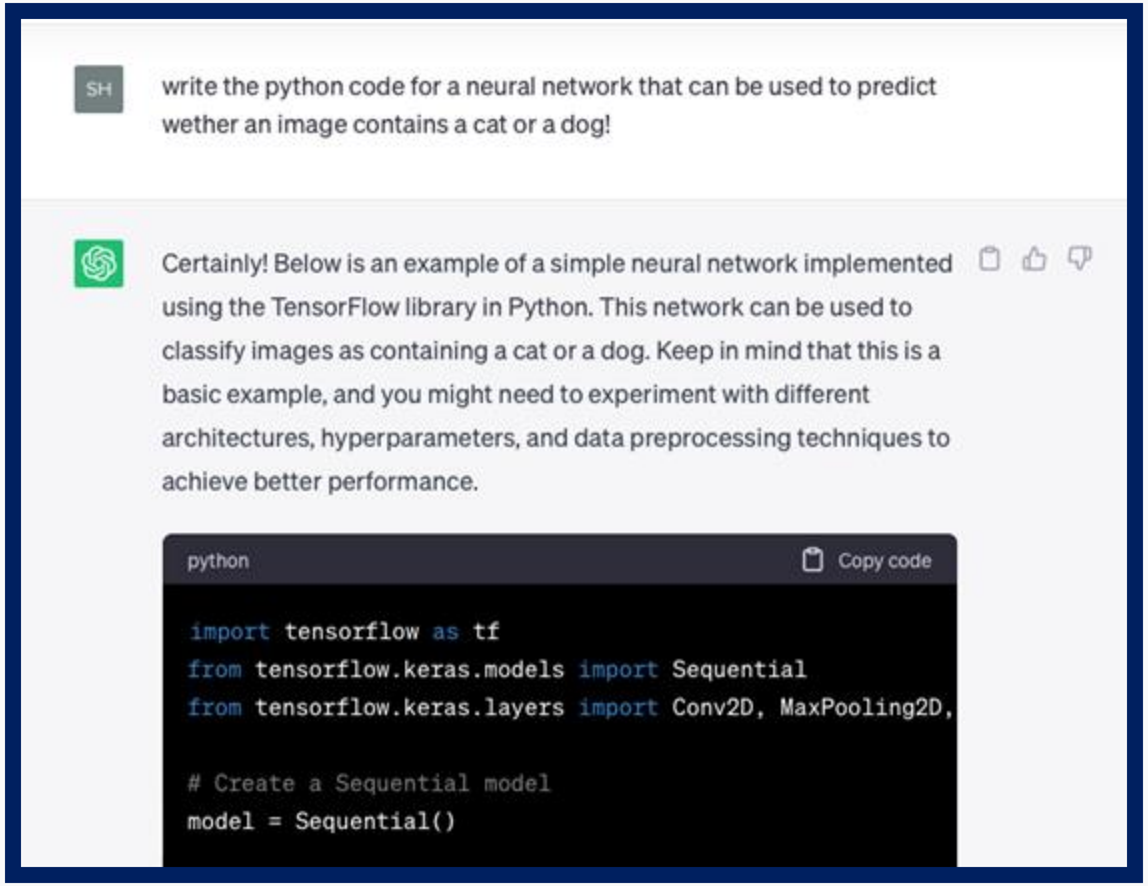
DALL·E

Recent developments

Personalized Customer Assistance



Computer Code Generation

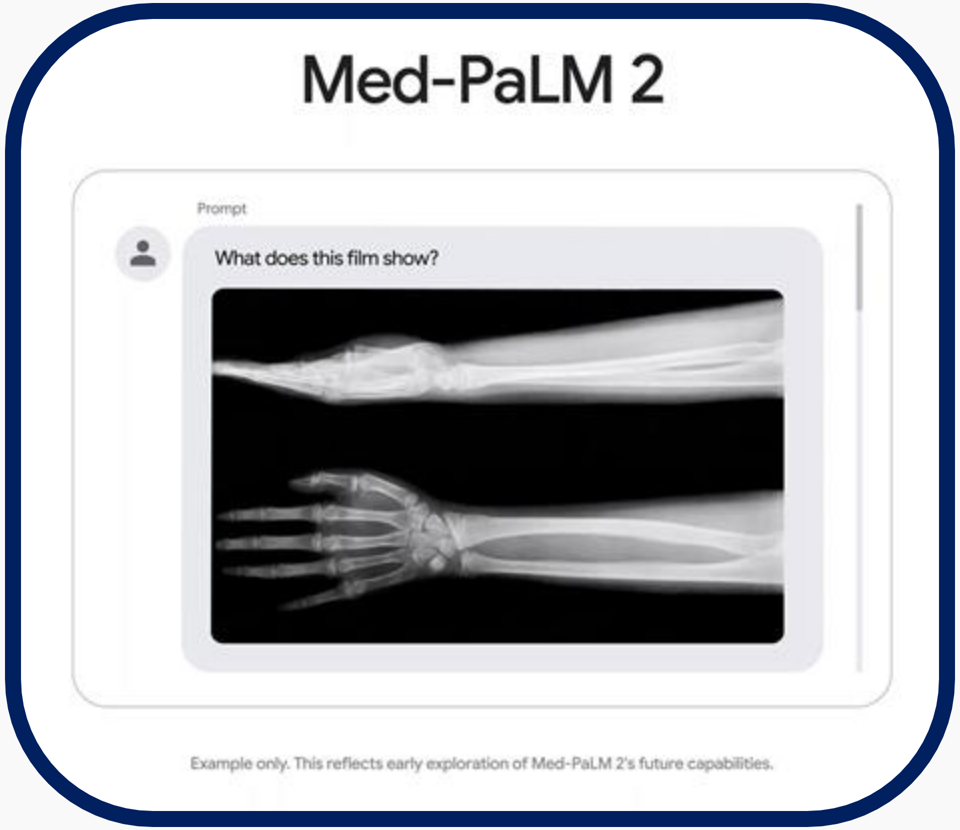


AI Conversational Assistant

ChatGPT

Recent developments

Disease Prediction



Google, 2023

Complex Object Detection

YOLOv5, 2024

The potential challenges in Data Science

Gender Bias

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

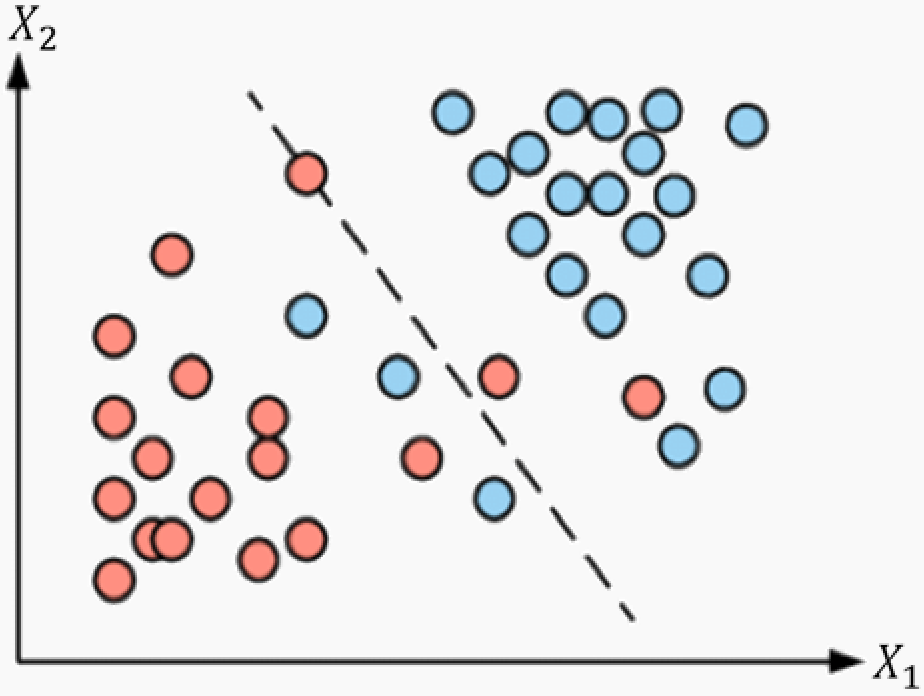
Racial Bias

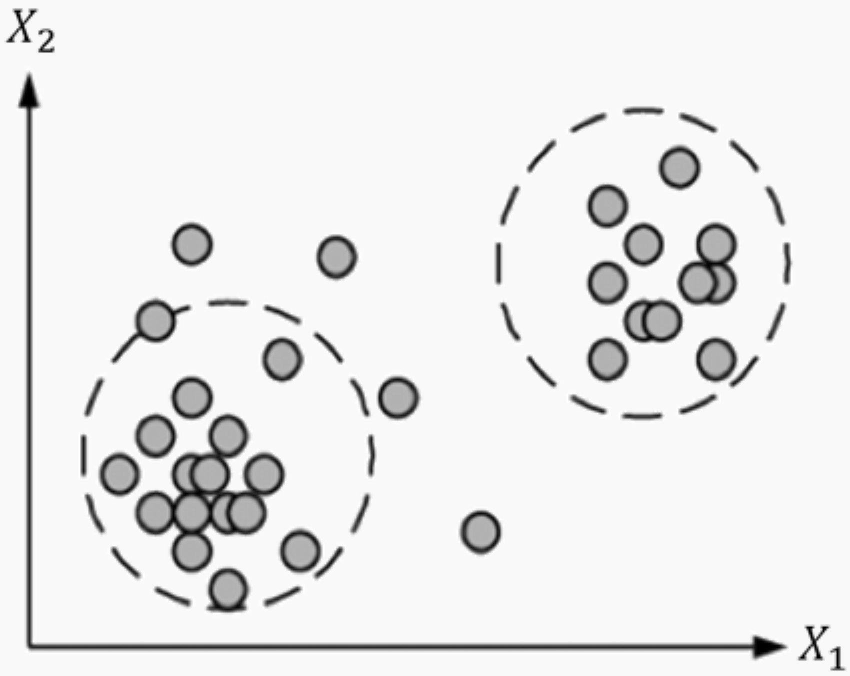
Risk models used in US courts have shown to be biased against non-white defendants

Outline

- Introduction
- **Review of basic concepts**
- Linear to Logistic Regression
- Logistic Regression to ANN
- Perceptron - Single neuron network
- Multi-Layer Perceptron (MLP)

Supervised v/s Unsupervised Machine Learning

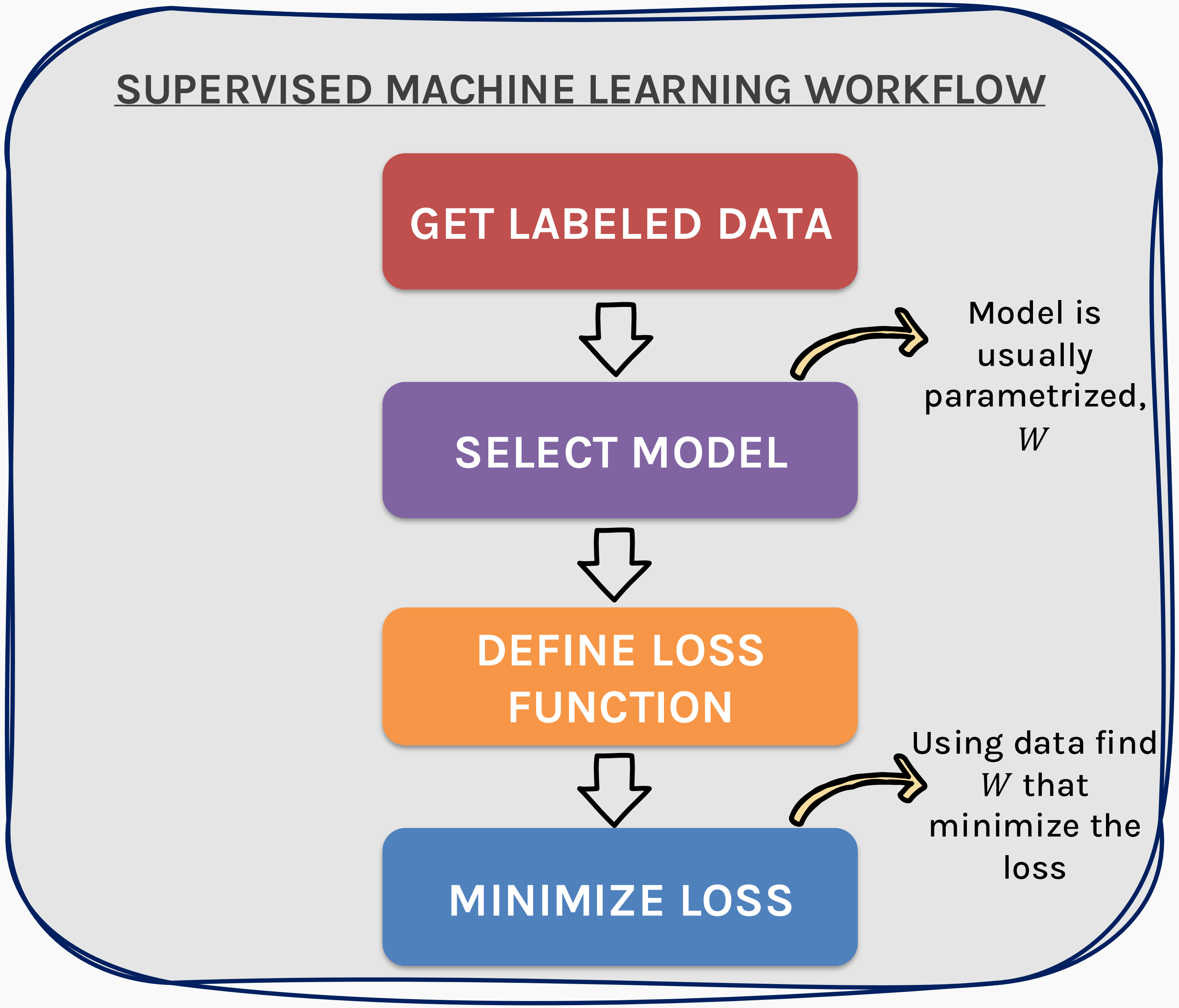




Supervised Learning: Learns with “labeled” data

Unsupervised Learning: Learns by clustering or association

Building blocks of **supervised** machine learning



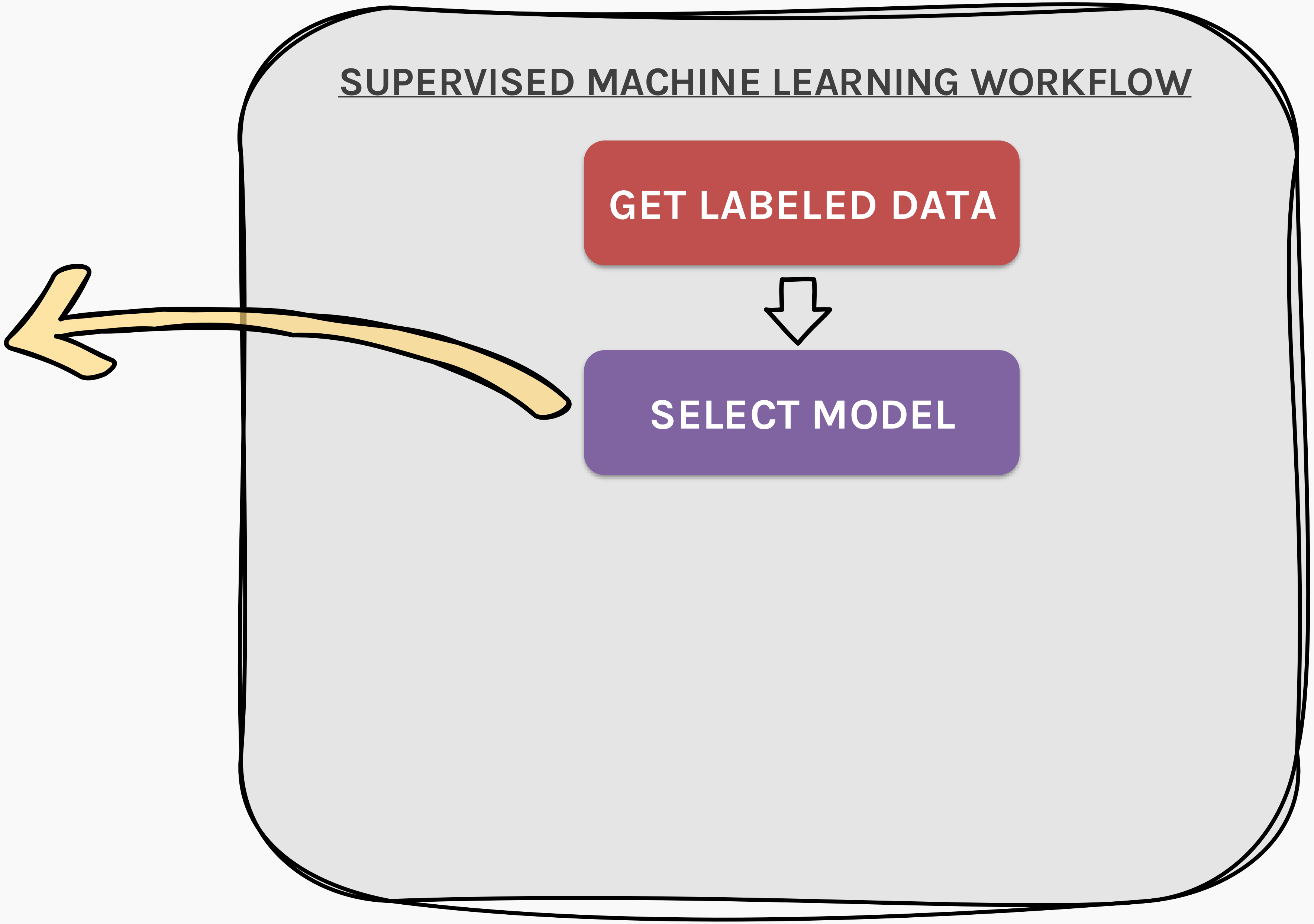
Building blocks of **supervised** machine learning

SUPERVISED MACHINE LEARNING WORKFLOW

GET LABELED DATA

Building blocks of **supervised** machine learning

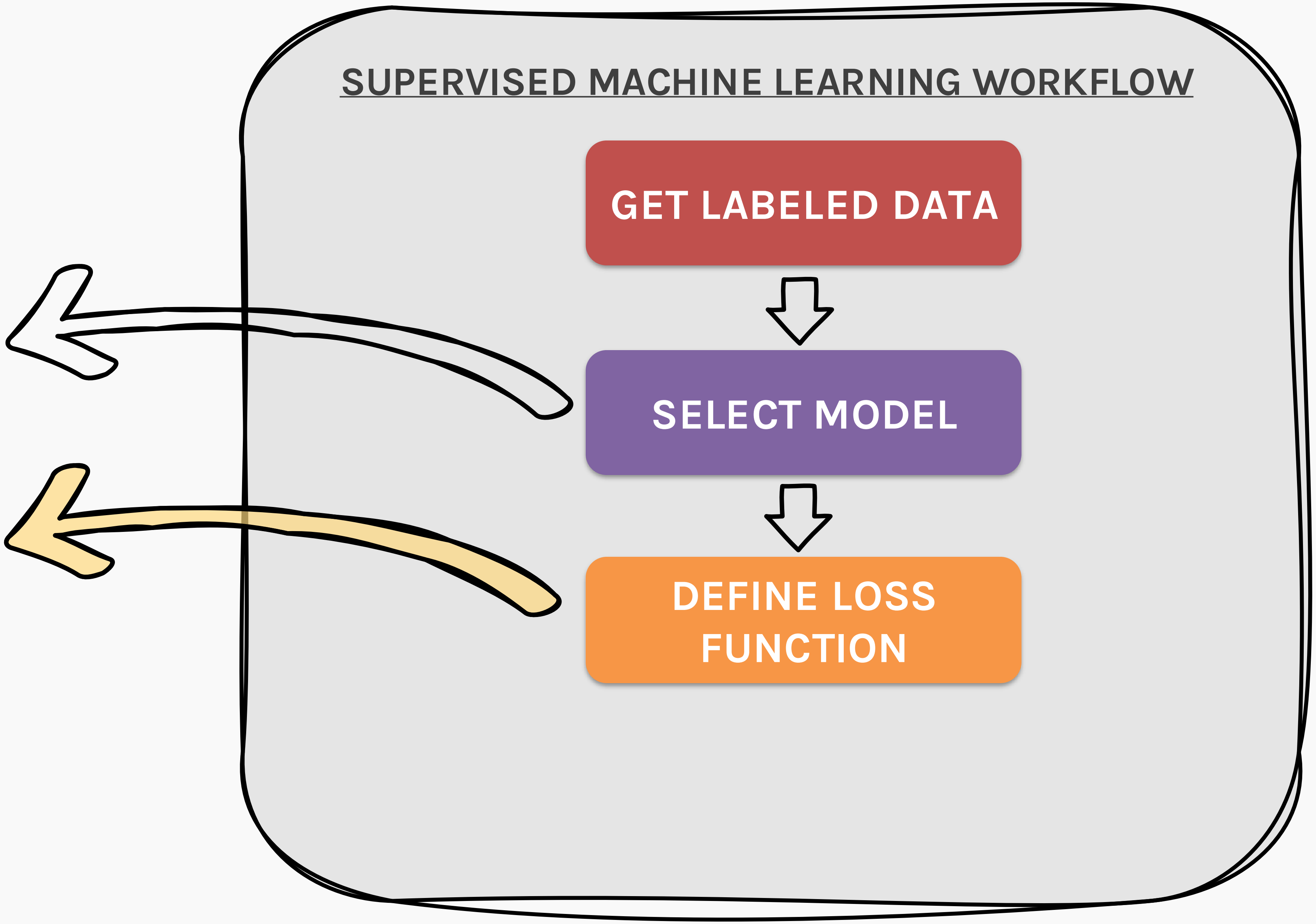
Regression, Classification, Object Detection



Building blocks of **supervised** machine learning

Regression, Classification, Object Detection

Mean Squared Error, Binary Cross Entropy, Categorical Cross Entropy, Hinge Loss

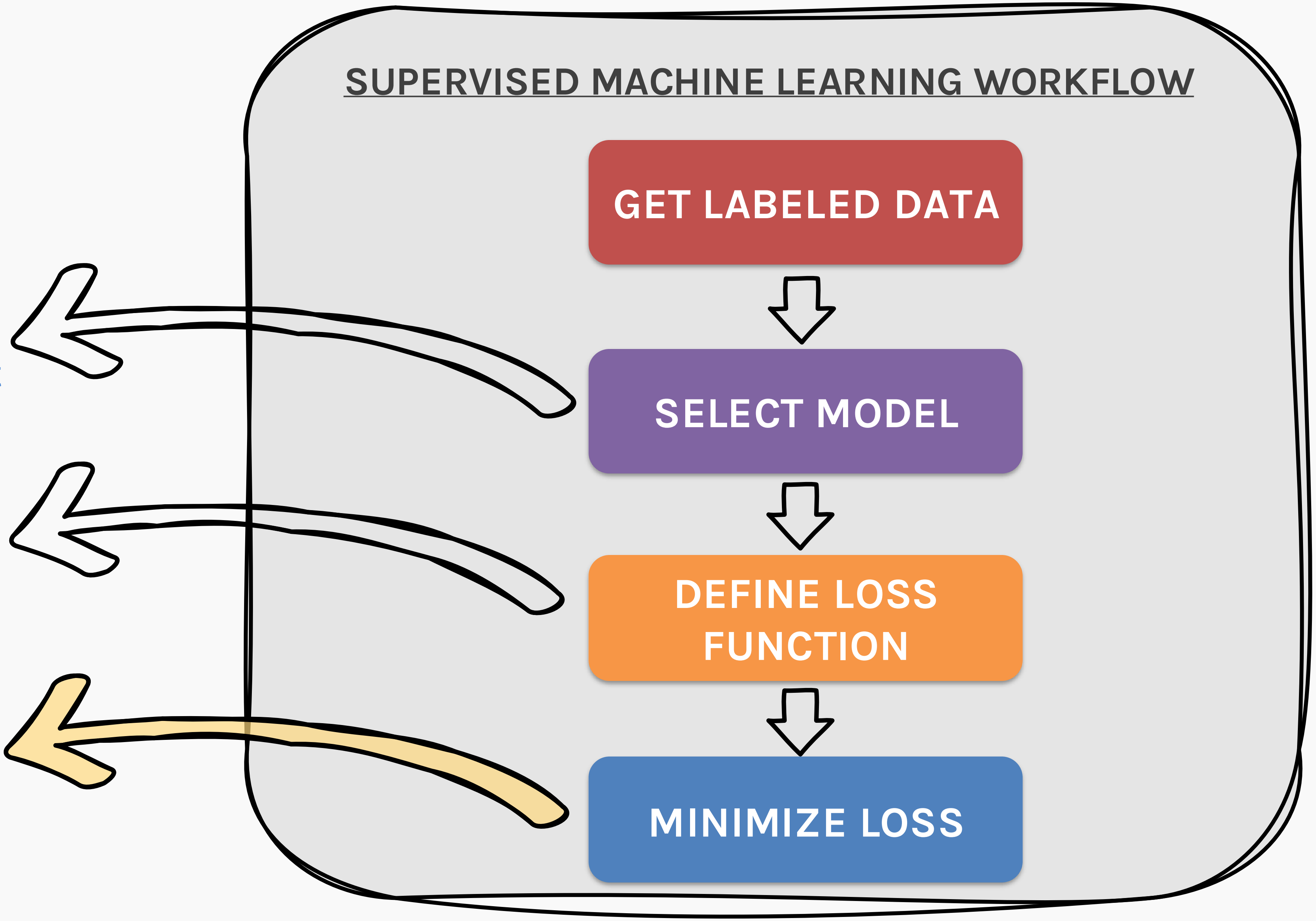


Building blocks of **supervised** machine learning

Regression, Classification, Object Detection

Mean Squared Error, Binary Cross Entropy, Categorical Cross Entropy, Hinge Loss

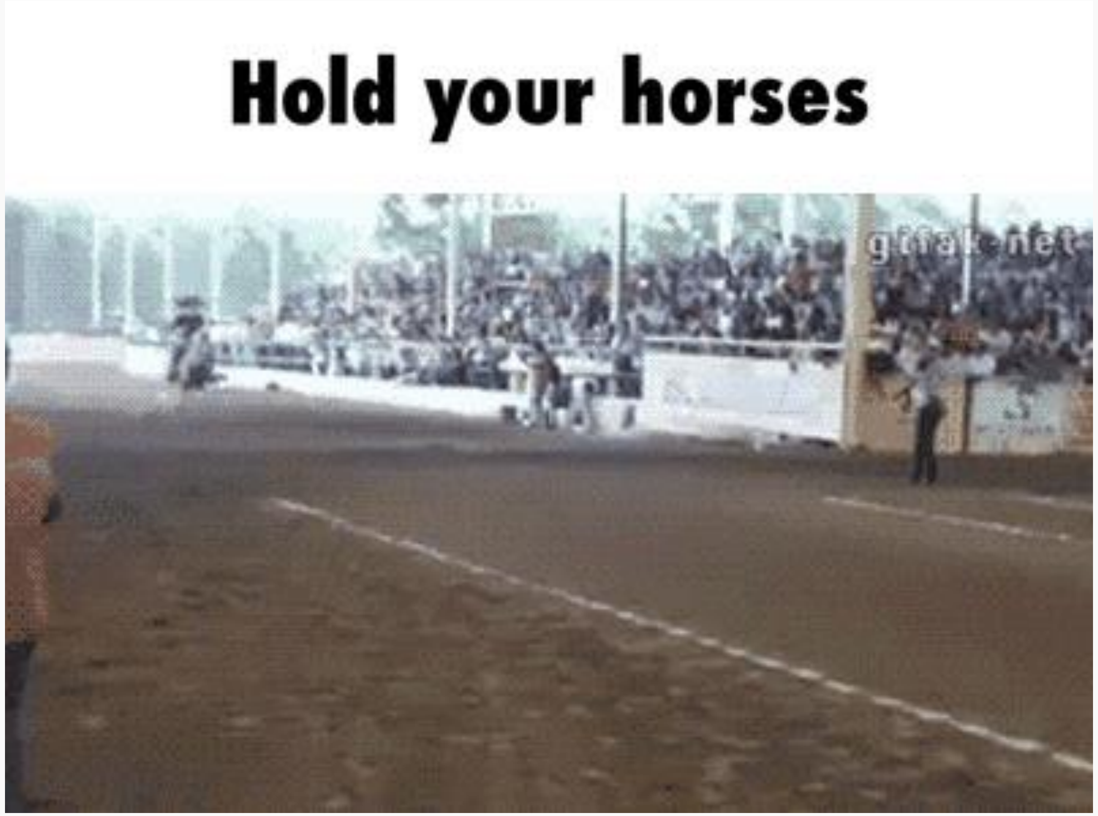
Analytical solution, Gradient Descent



Outline

- Introduction
- Review of basic concepts
- **Linear to Logistic Regression**
- Logistic Regression to ANN
- Perceptron - Single neuron network
- Multi-Layer Perceptron (MLP)

Linear to Logistic Regression



Before we dive into what an ANN looks like, let us review Logistic Regression

Linear to Logistic Regression

Linear Regression is like predicting how many points a Basketball Player will score in a game based on past performance!

Logistic Regression, however, predicts whether the same player will score above or below a certain number of points. Basically, a Yes or No by giving us the probability between 0 and 1

What is Classification?

Consider the dataset that contains a binary outcome AHD for 303 patients who presented with chest pain.

X
predictors

Y
Yes or No
response variable

Age	Sex	ChestPain	RestBP	Chol	MaxHR	ExAng	Thal	AHD
63	1	typical	145	233	150	0	fixed	No
67	1	asymptomatic	160	286	108	1	normal	Yes
67	1	asymptomatic	120	229	129	1	reversable	Yes
37	1	nonanginal	130	250	187	0	normal	No

What is Classification?

Consider the dataset that contains a binary outcome AHD for 303 patients who presented with chest pain.

X
predictors

Age	Sex	ChestPain	RestBP	Chol	MaxHR	ExAng	Thal	AHD
63	1	typical	145	233	150	0	fixed	No
67	1	asymptomatic	160	286	108	1	normal	Yes
67	1	asymptomatic	120	229	129	1	reversable	Yes
37	1	nonanginal	130	250	187	0	normal	No

Yes indicates **presence** of heart disease

What is Classification?

Consider the dataset that contains a binary outcome AHD for 303 patients who presented with chest pain.

X
predictors

No indicates **absence** of heart disease

Age	Sex	ChestPain	RestBP	Chol	MaxHR	ExAng	Thal	AHD
63	1	typical	145	233	150	0	fixed	No
67	1	asymptomatic	160	286	108	1	normal	Yes
67	1	asymptomatic	120	229	129	1	reversable	Yes
37	1	nonanginal	130	250	187	0	normal	No

Why not Linear Regression?

- Here the response variable y has only two categories.

$$y = \begin{cases} 1, & \textit{has heart disease} \\ 0, & \textit{no heart disease} \end{cases}$$

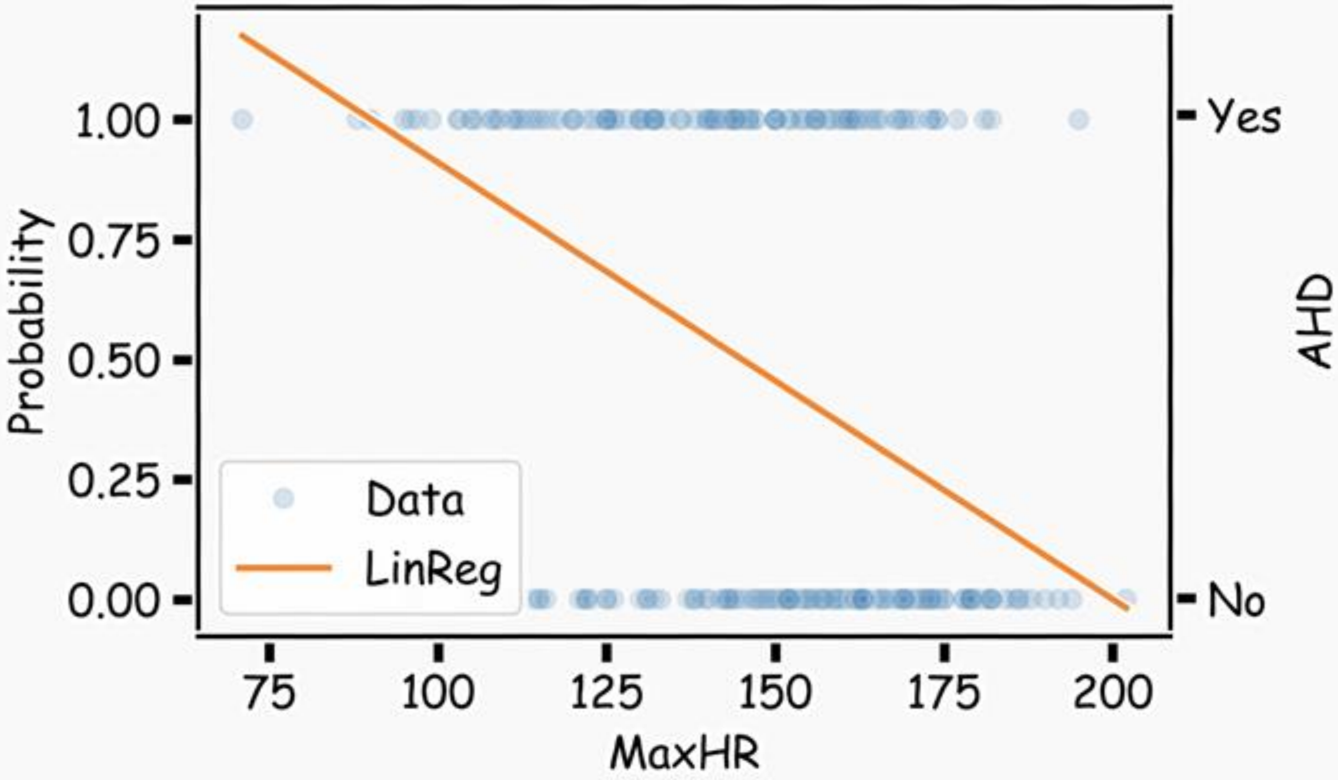
- Linear regression could be used to predict the probability $\mathrm{P}(y = 1)$ directly from a set of predictors such as sex, cholesterol levels, etc.

  If $\mathrm{P}(\mathrm{y} = 1) \geq 0.5$, we could predict that the patient has heart disease and predict otherwise if $\mathrm{P}(\mathrm{y} = 1) < 0.5$.

Why not Linear Regression?

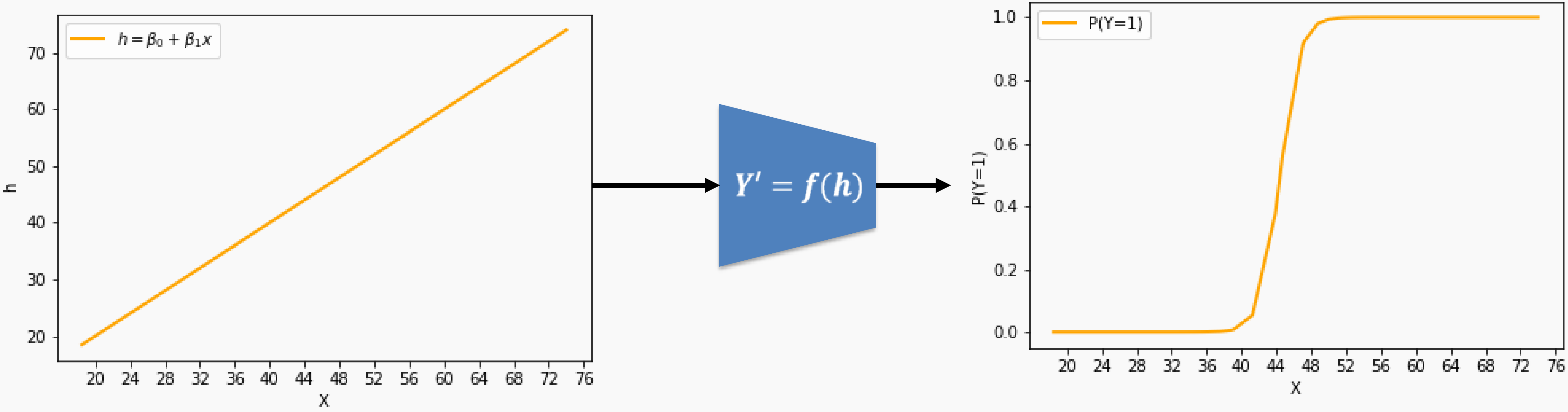
What could go wrong with this linear regression model?

Since this is modeling $P(y=1)$, values for $\hat{y}$ below 0 and above 1 would not make sense as a probability.



Logistic Regression

Now we know that linear regression yields values for probability that are larger than 1 or smaller than 0. So, what can we do to fix this?



Logistic Regression

We can use the sigmoid function:

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

Logistic Regression

- Logistic Regression addresses the problem of estimating a probability, $P(y = 1)$, to be outside the range of [0,1].

- The Logistic Regression model uses a function, called the **logistic** function, to model $P(y = 1)$:

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

This is the
same to the
sigmoid
function

Using Logistic Regression for Classification

How can we use a logistic regression model to perform classification?

That is, how can we predict when $Y = 1$ vs. when $Y = 0$?

We can classify all observations for which:

- Classify all observations with $\hat{P}(Y = 1) \geq 0.5$ to be in the group associated with $Y = 1$.
- Classify all observations with $\hat{P}(Y = 1) < 0.5$ to be in the group associated with $Y = 0$.

Loss Function

How do we find out if our predictions are any good?

We use a Loss Function!

A Loss Function compares the predicted output and the actual target values.

In the Linear Regression, we use a Mean Squared Error

Loss Function

What do we use in Logistic Regression?

We use Binary Cross Entropy

Actual Label

Predicted Probability

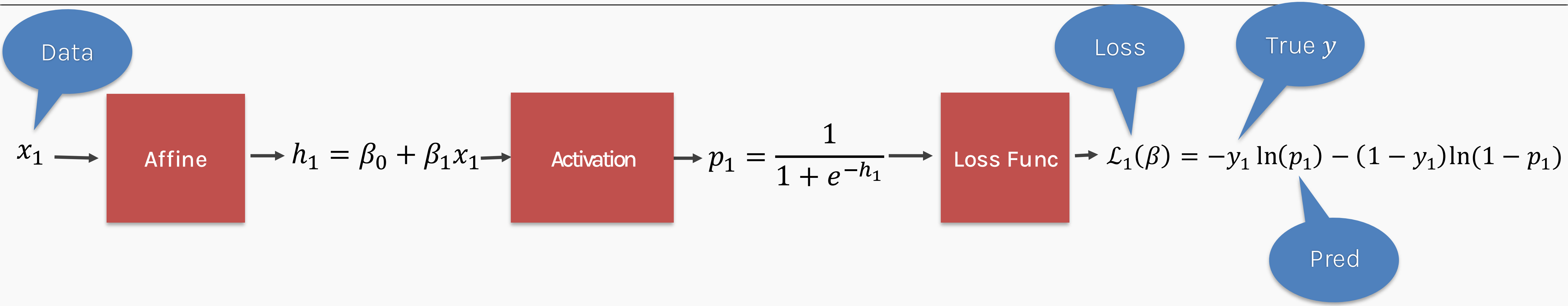
$$L_{BCE} = \sum_i -y_i \log p_i - (1 - y_i) \log(1 - p_i)$$

We subtract, as the goal is to minimize the loss

Outline

- Introduction
- Review of basic concepts
- Linear to Logistic Regression
- **Logistic Regression to ANN**
- Perceptron - Single neuron network
- Multi-Layer Perceptron (MLP)

Logistic Regression to ANN



Logistic Regression to ANN

$x_1 \rightarrow$ Affine $\rightarrow h_1 = \beta_0 + \beta_1 x_1 \rightarrow$ Activation $\rightarrow p_1 = \dfrac{1}{1 + e^{-h_1}} \rightarrow$ Loss Func $\rightarrow \mathcal{L}_1(\beta) = -y_1 \ln(p_1) - (1 - y_1)\ln(1 - p_1)$

$x_2 \rightarrow$ Affine $\rightarrow h_2 = \beta_0 + \beta_1 x_2 \rightarrow$ Activation $\rightarrow p_2 = \dfrac{1}{1 + e^{-h_2}} \rightarrow$ Loss Func $\rightarrow \mathcal{L}_2(\beta) = -y_2 \ln(p_1) - (1 - y_2)\ln(1 - p_2)$

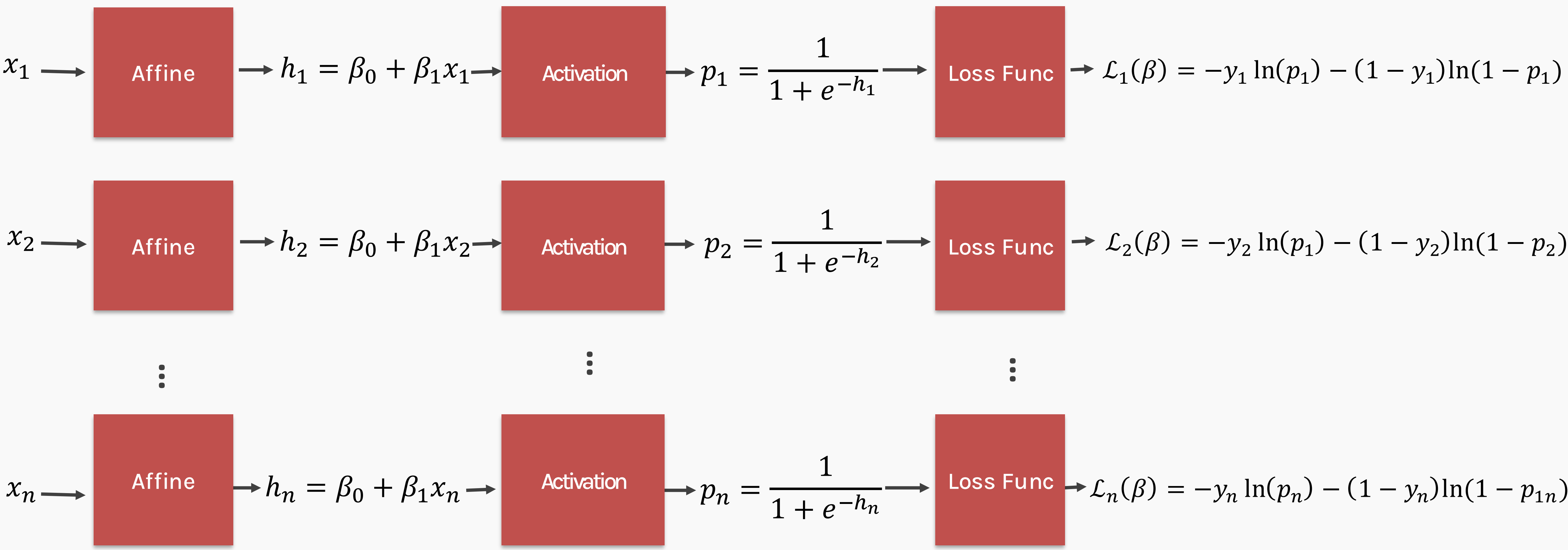
Logistic Regression to ANN

$x_1 \rightarrow$ Affine $\rightarrow h_1 = \beta_0 + \beta_1 x_1 \rightarrow$ Activation $\rightarrow p_1 = \dfrac{1}{1 + e^{-h_1}} \rightarrow$ Loss Func $\rightarrow \mathcal{L}_1(\beta) = -y_1 \ln(p_1) - (1 - y_1)\ln(1 - p_1)$

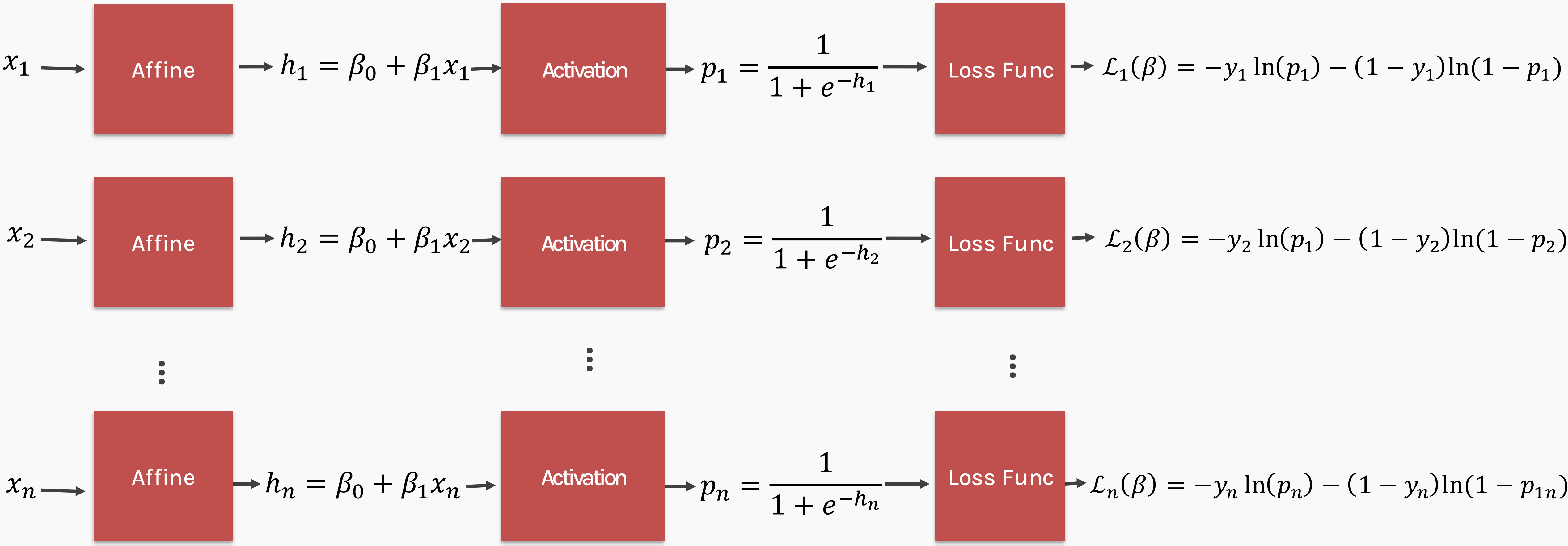
$x_2 \rightarrow$ Affine $\rightarrow h_2 = \beta_0 + \beta_1 x_2 \rightarrow$ Activation $\rightarrow p_2 = \dfrac{1}{1 + e^{-h_2}} \rightarrow$ Loss Func $\rightarrow \mathcal{L}_2(\beta) = -y_2 \ln(p_1) - (1 - y_2)\ln(1 - p_2)$

$\vdots \qquad \vdots \qquad \vdots$

Logistic Regression to ANN



Logistic Regression to ANN



$$\mathcal{L}(\beta) = \sum_i^n \mathcal{L}_i(\beta)$$

Outline

- Introduction
- Review of basic concepts
- Linear to Logistic Regression
- Logistic Regression to ANN
- **Perceptron - Single neuron network**
- Multi-Layer Perceptron (MLP)

Build our first ANN

n_p :number of predictors

n_o: number of observations

matrix: $n_o \times n_p$

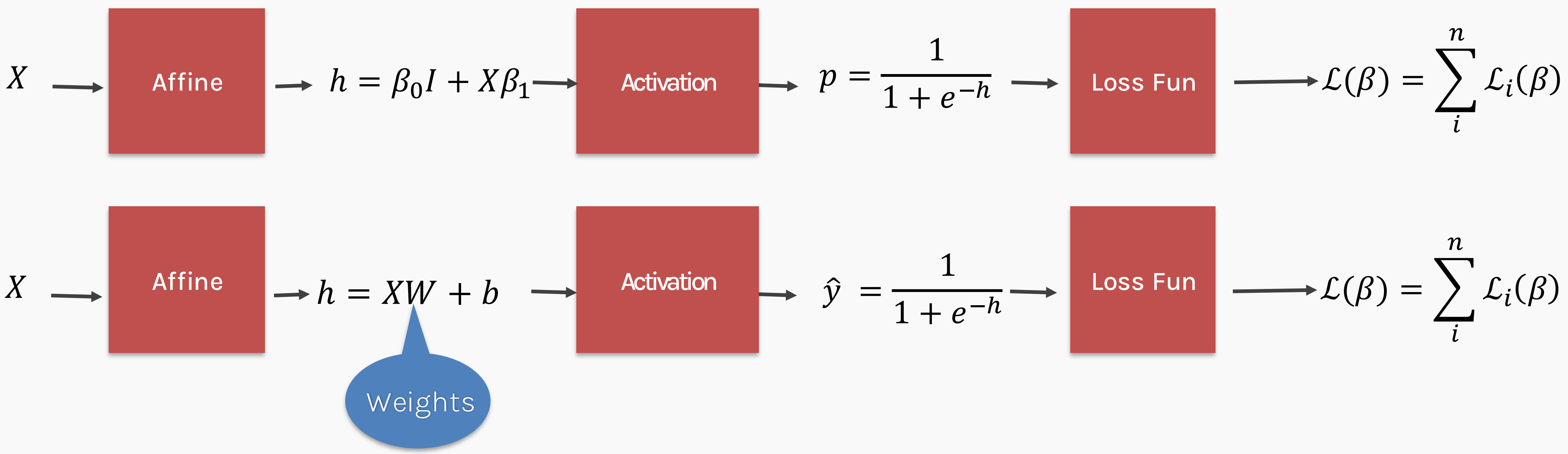
vector: $n_o \times 1$

vector: n_p

scalar

X → Affine → $h = \beta_0 I + X\beta_1$ → Activation → $p = \dfrac{1}{1+e^{-h}}$ → Loss Fun → $\mathcal{L}(\beta) = \sum_i^n \mathcal{L}_i(\beta)$

Build our first ANN



Build our first ANN

$X \rightarrow$ Affine $\rightarrow h = \beta_0 I + X\beta_1 \rightarrow$ Activation $\rightarrow p = \dfrac{1}{1+e^{-h}} \rightarrow$ Loss Fun $\rightarrow \mathcal{L}(\beta) = \sum_i^n \mathcal{L}_i(\beta)$

$X \rightarrow$ Affine $\rightarrow h = XW + b \rightarrow$ Activation $\rightarrow \hat{y} = \dfrac{1}{1+e^{-h}} \rightarrow$ Loss Fun $\rightarrow \mathcal{L}(\beta) = \sum_i^n \mathcal{L}_i(\beta)$

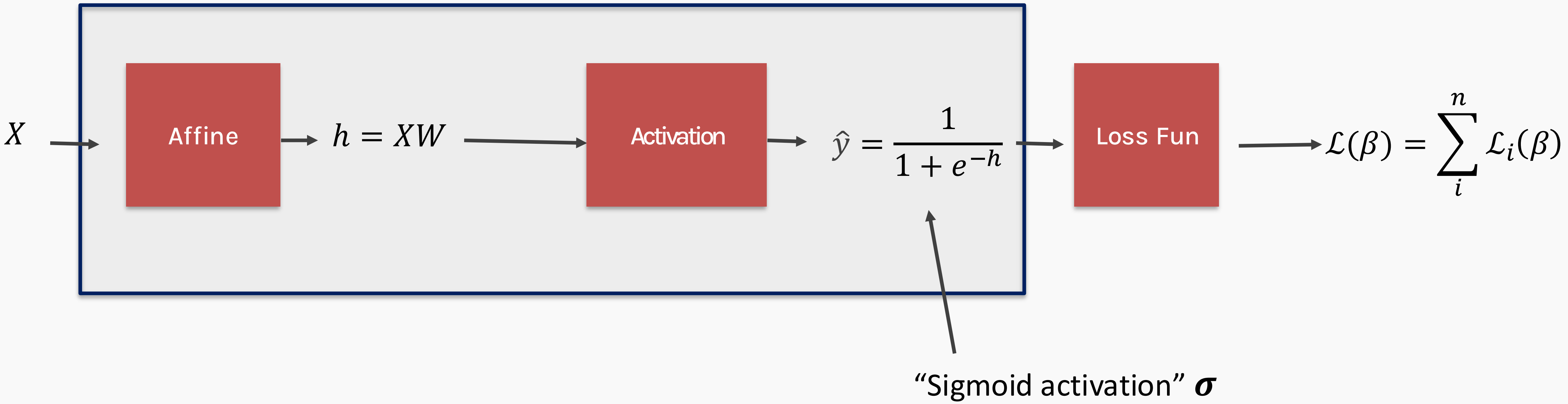
$X \rightarrow$ Affine $\rightarrow h = XW \rightarrow$ Activation $\rightarrow \hat{y} = \dfrac{1}{1+e^{-h}} \rightarrow$ Loss Fun $\rightarrow \mathcal{L}(\beta) = \sum_i^n \mathcal{L}_i(\beta)$

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

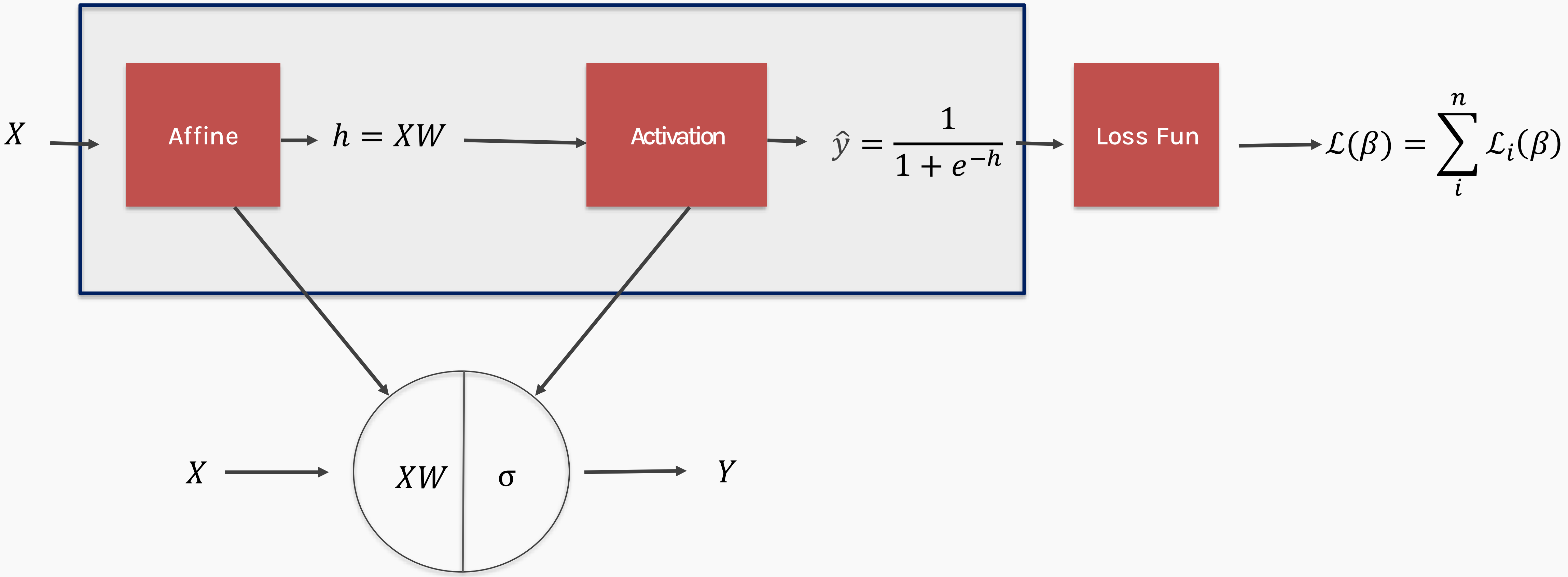
Build our first ANN

X → **Affine** → $h = XW$ → **Activation** → $\hat{y} = \dfrac{1}{1+e^{-h}}$ → **Loss Fun** → $\mathcal{L}(\beta) = \sum_{i}^{n} \mathcal{L}_i(\beta)$

Build our first ANN

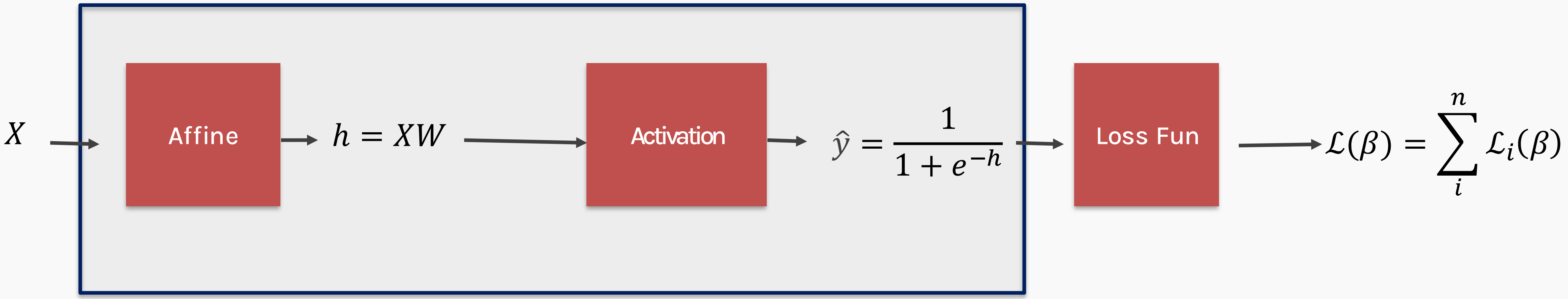


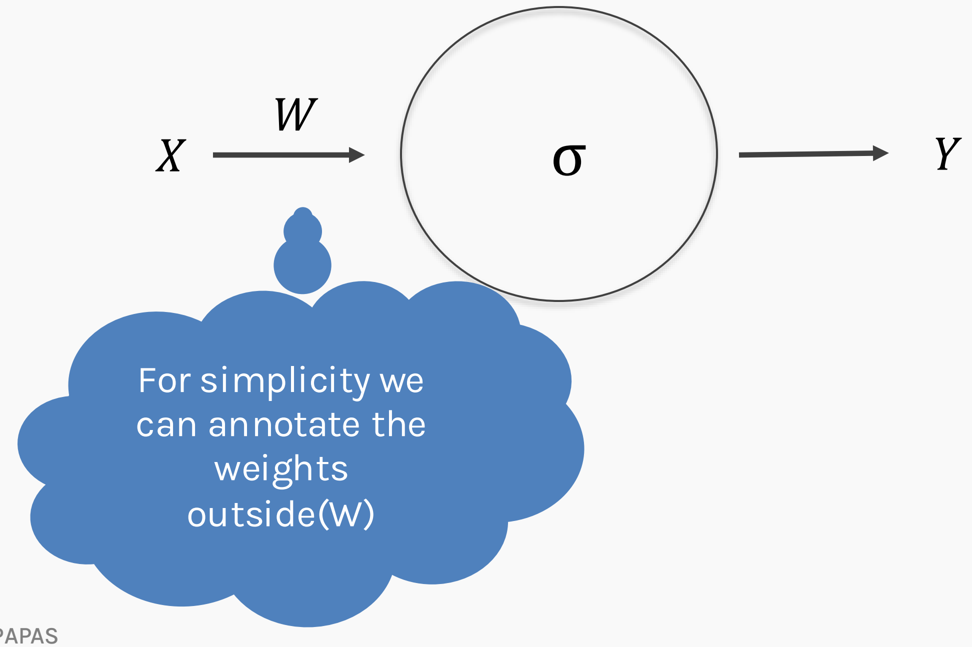
Build our first ANN



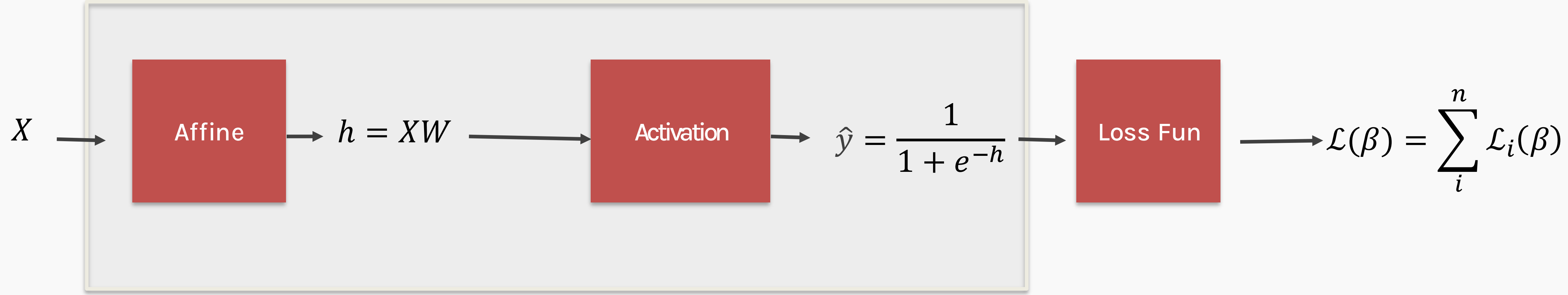
A neuron essentially applies an affine transformation on the input, followed by an activation function

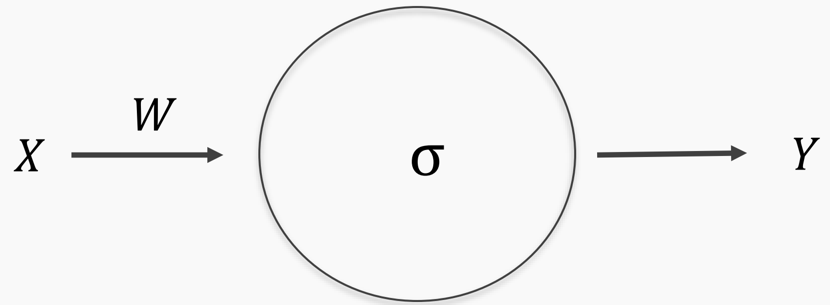
Build our first ANN





Build our first ANN



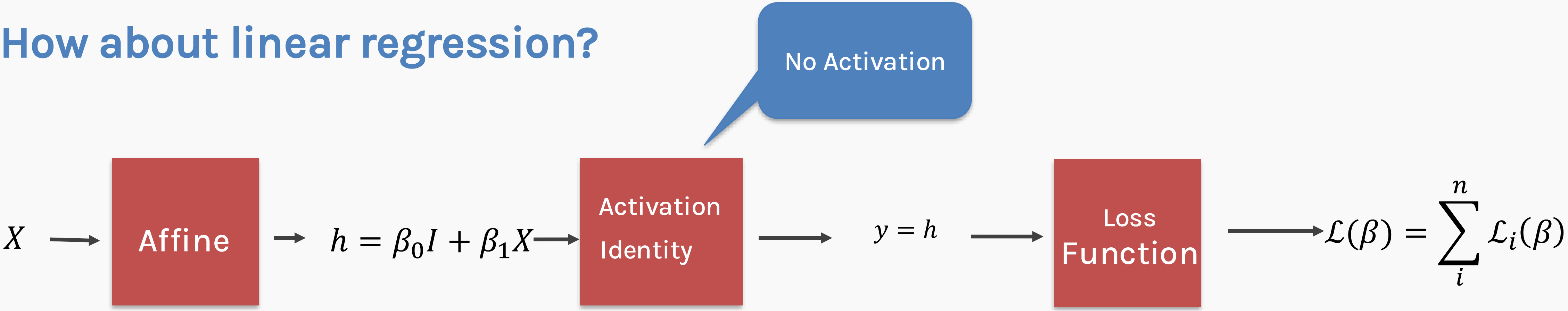


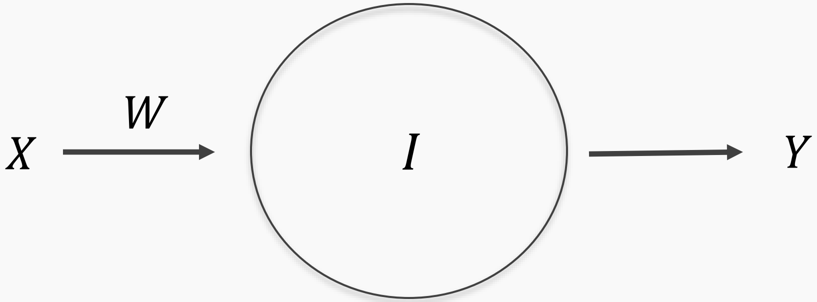
Single Neuron Neural "Network"

A single neuron

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

Summary

So far:

- A single neuron is simply an affine transformation followed by an activation function.
- A single neuron with sigmoid activation is equivalent to the logistic regression and a single neuron with no activation is equivalent to linear regression.