Introduction to ANN and Perceptron

Pavlos Protopapas

Outline

- Introduction
- Review of basic concepts
- Perceptron - Single neuron network
- Multi-Layer Perceptron (MLP)

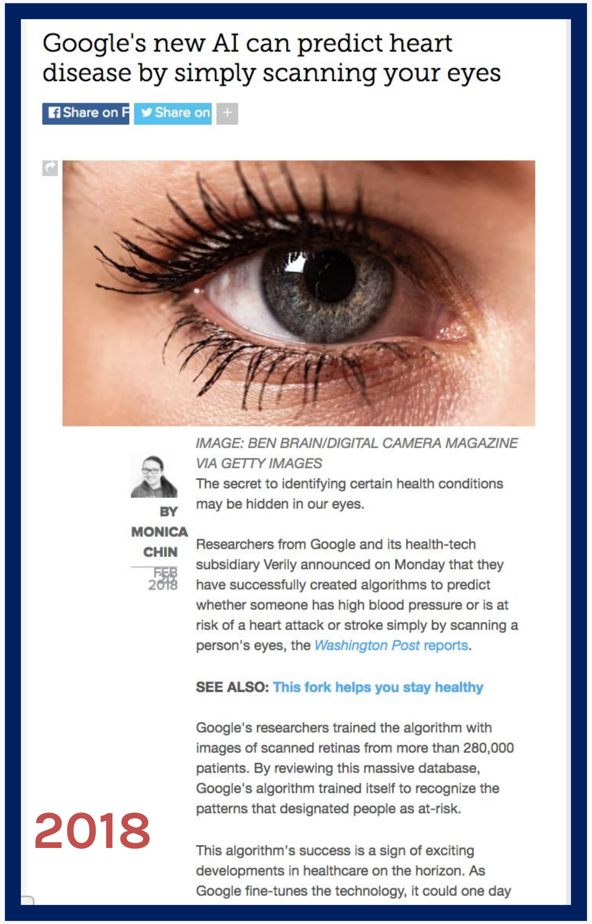
Outline

- **Introduction**
- Review of basic concepts
- Perceptron - Single neuron network
- Multi-Layer Perceptron (MLP)

Historical Trends

Game strategy

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

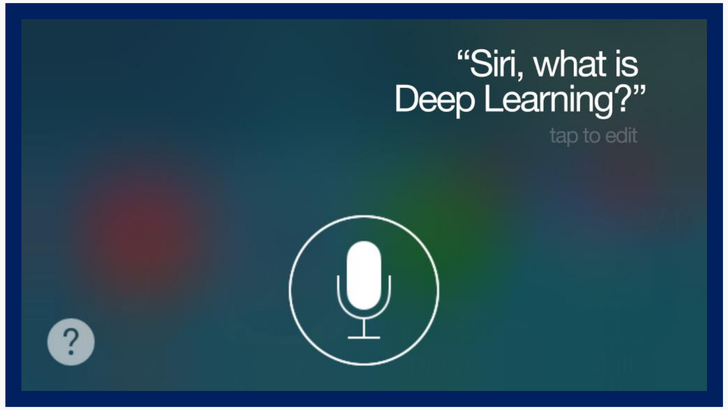

DeepMind

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

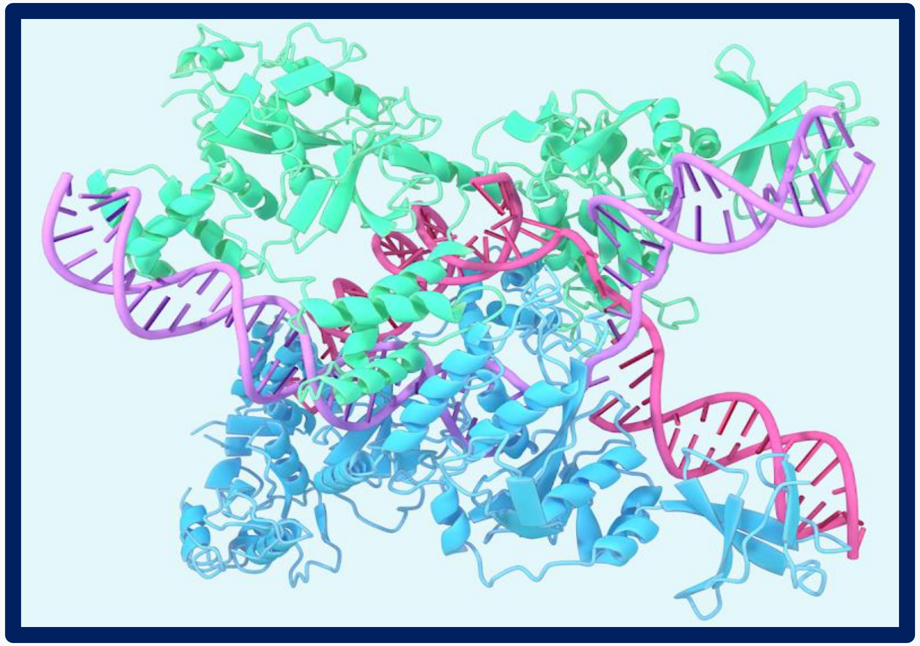
Natural Language Processing



2011

Recent developments

Protein folding



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

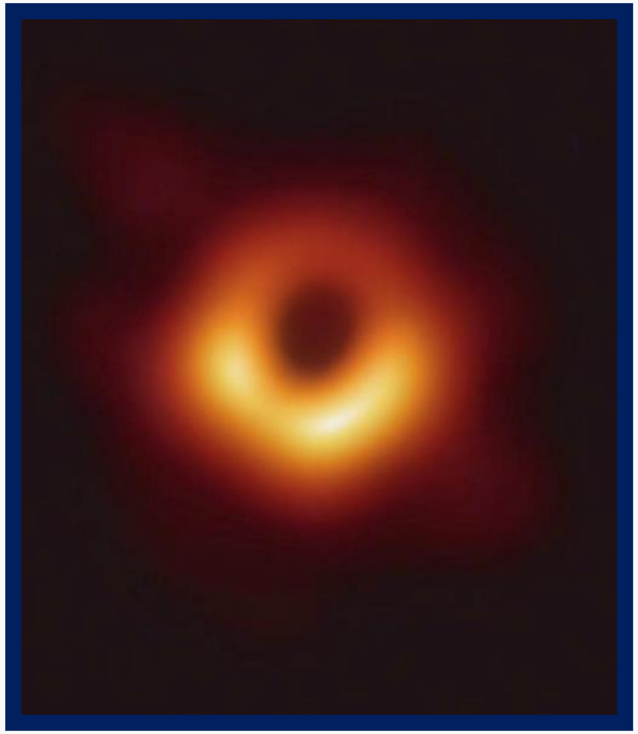
2020

Autonomous cars

AI Detecting objects to assist with autonomous driving.

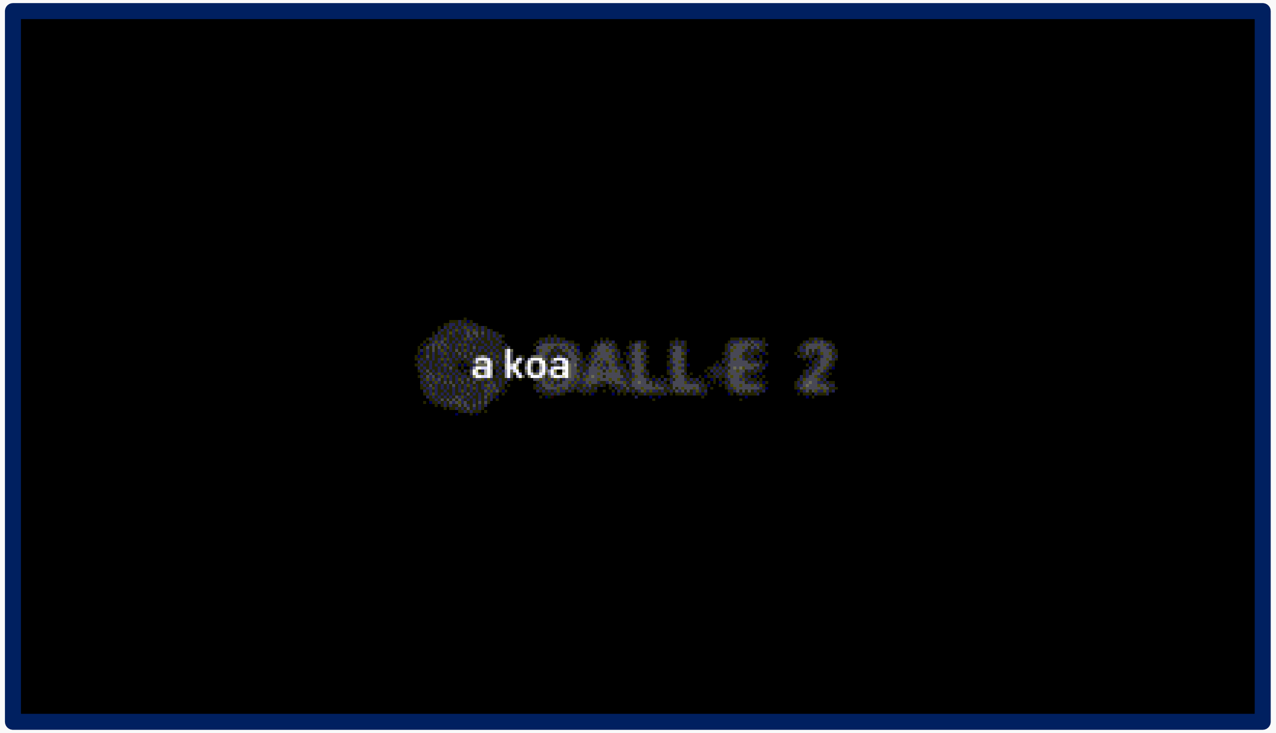
Recent developments

Image Reconstruction from Sparse Frequency Measurements



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

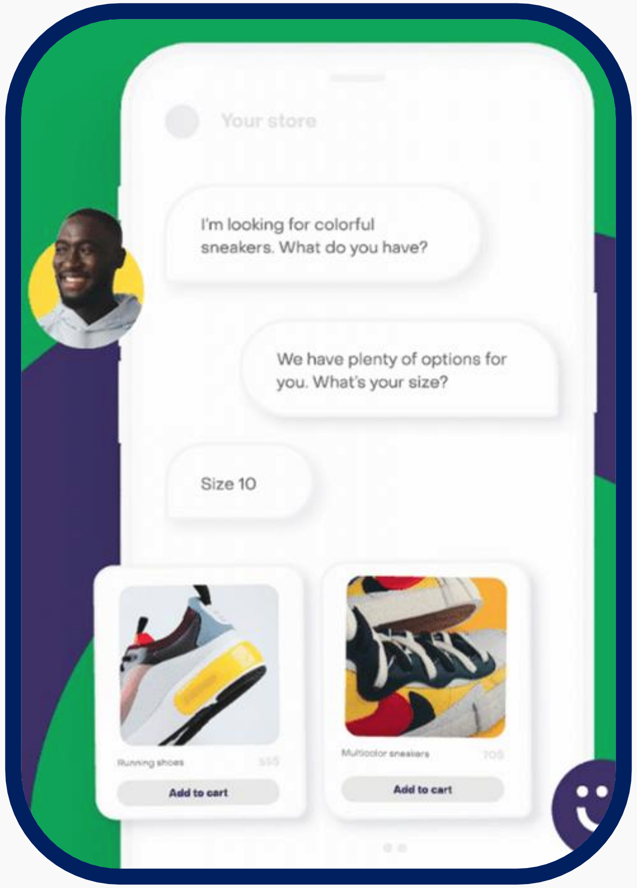
Text to Image Generation



DALL·E

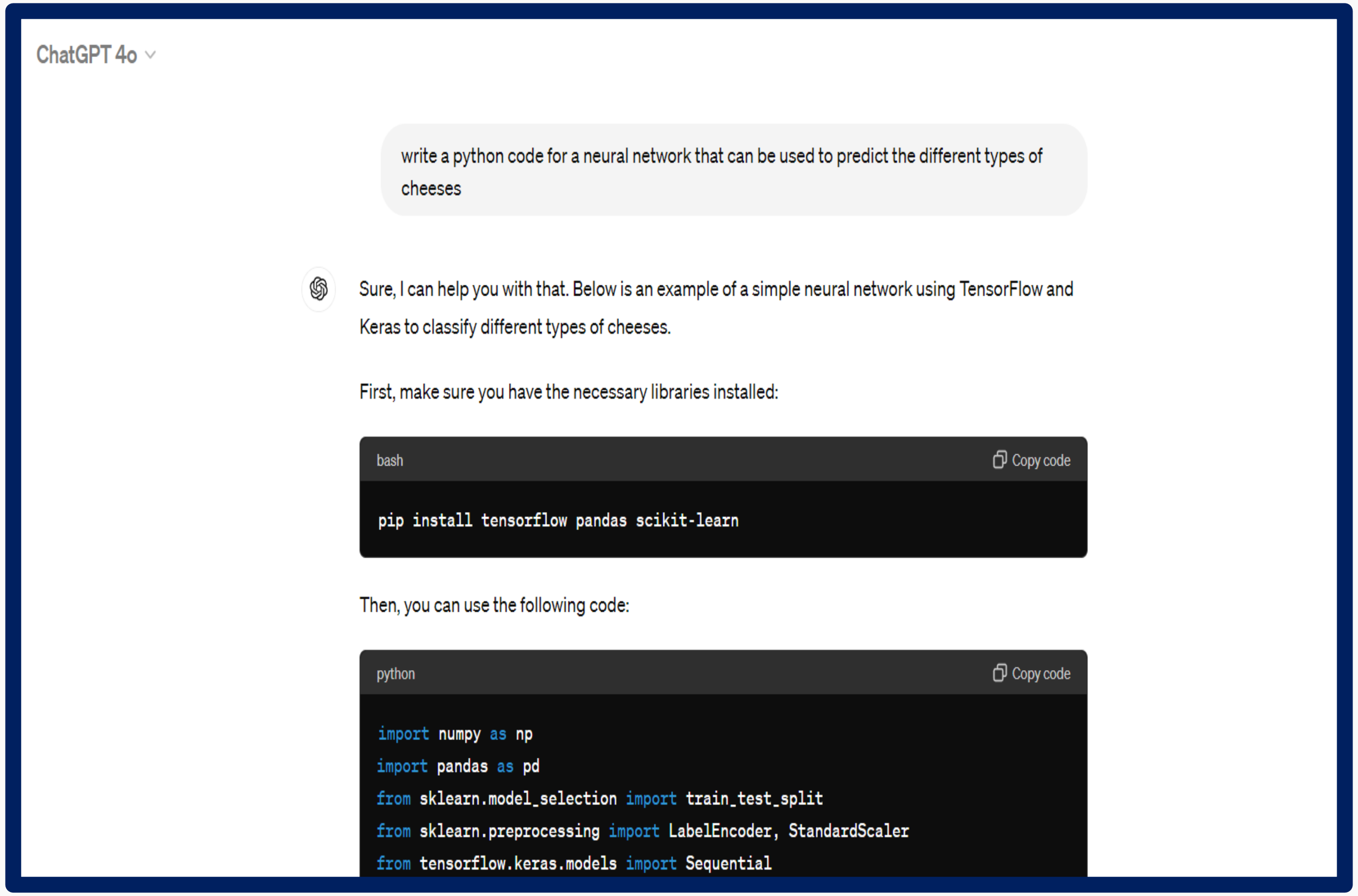
Recent developments

Personalized Customer Assistance



shopify

Computer Code Generation

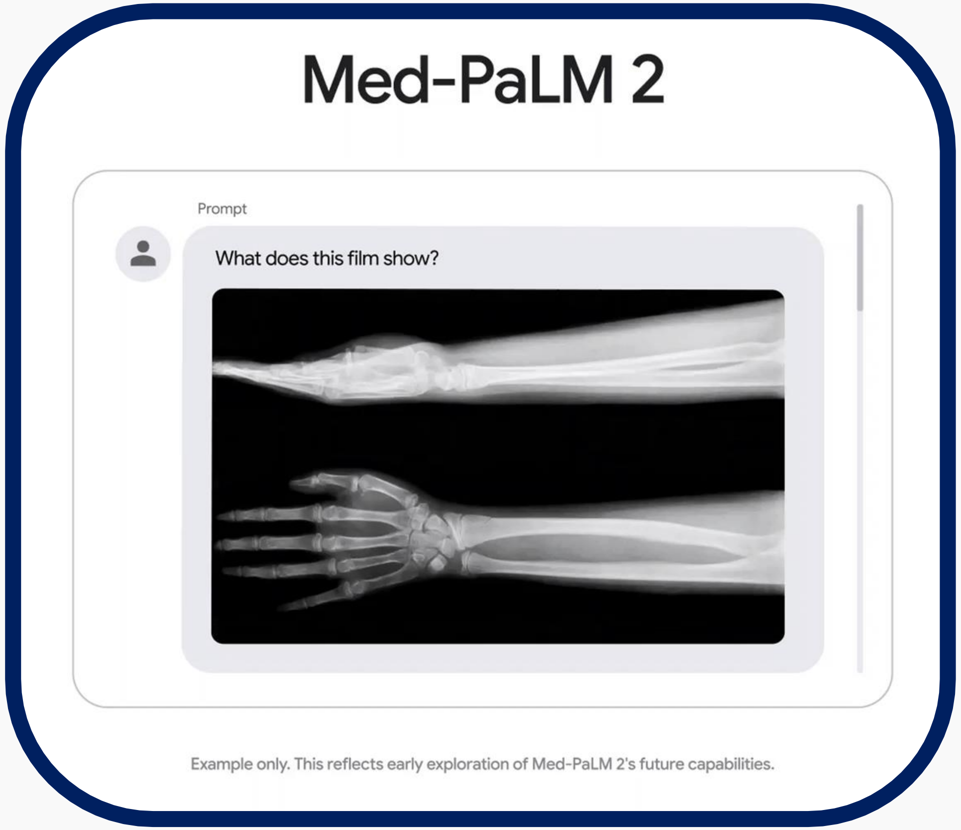


AI Conversational Assistant

ChatGPT

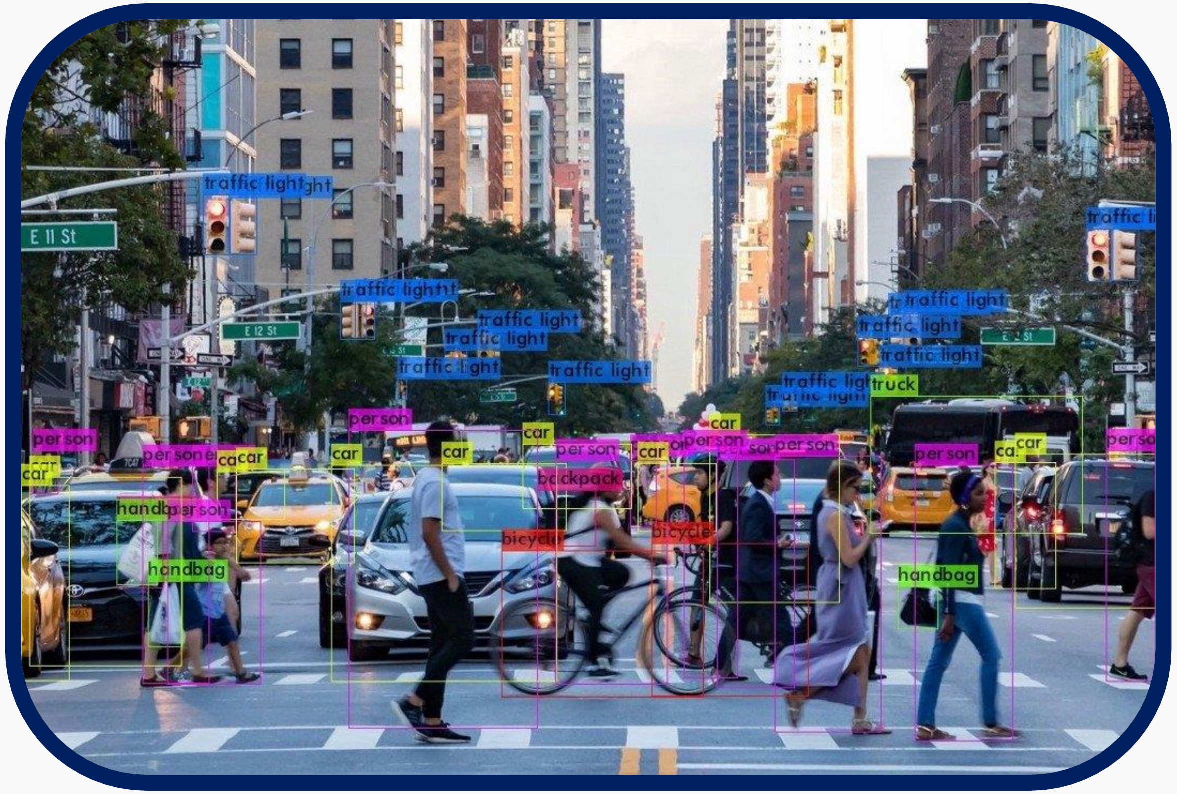
Recent developments

Disease Prediction



Google, 2023

Complex Object Detection



2024, YOLOv5

The potential challenges in Data Science

Gender Bias

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

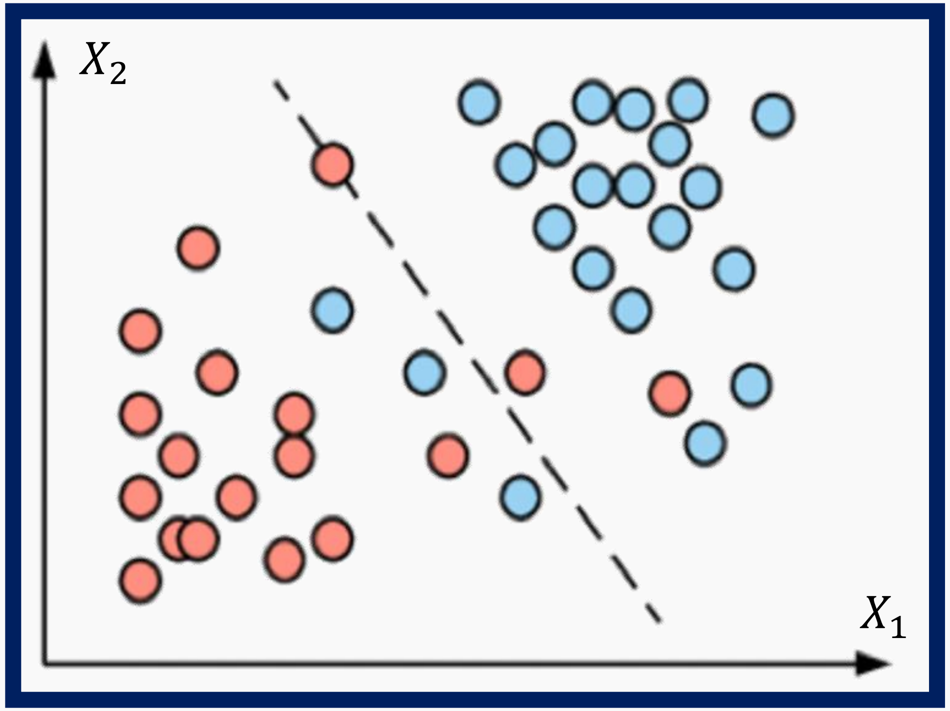
Racial Bias

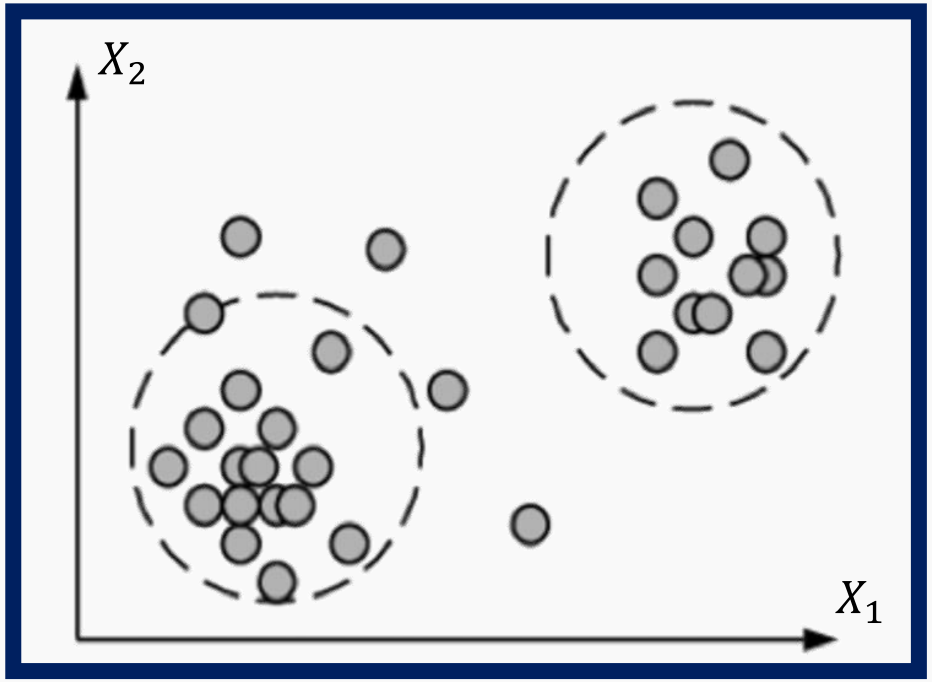
Risk models used in US courts have shown to be biased against non-white defendants

Outline

- Introduction
- **Review of basic concepts**
- Perceptron - Single neuron network
- Multi-Layer Perceptron (MLP)

Supervised v/s Unsupervised Machine Learning

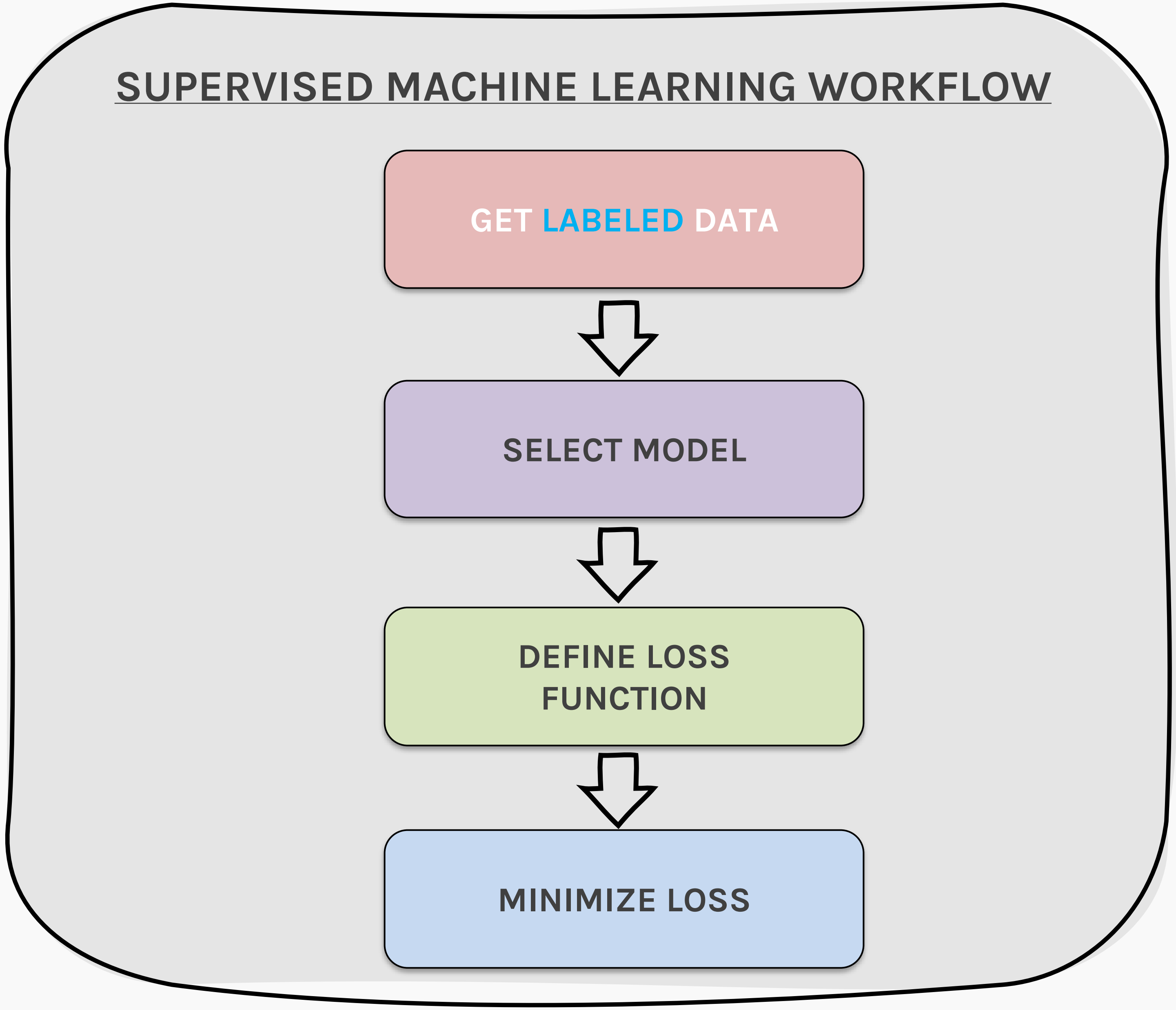




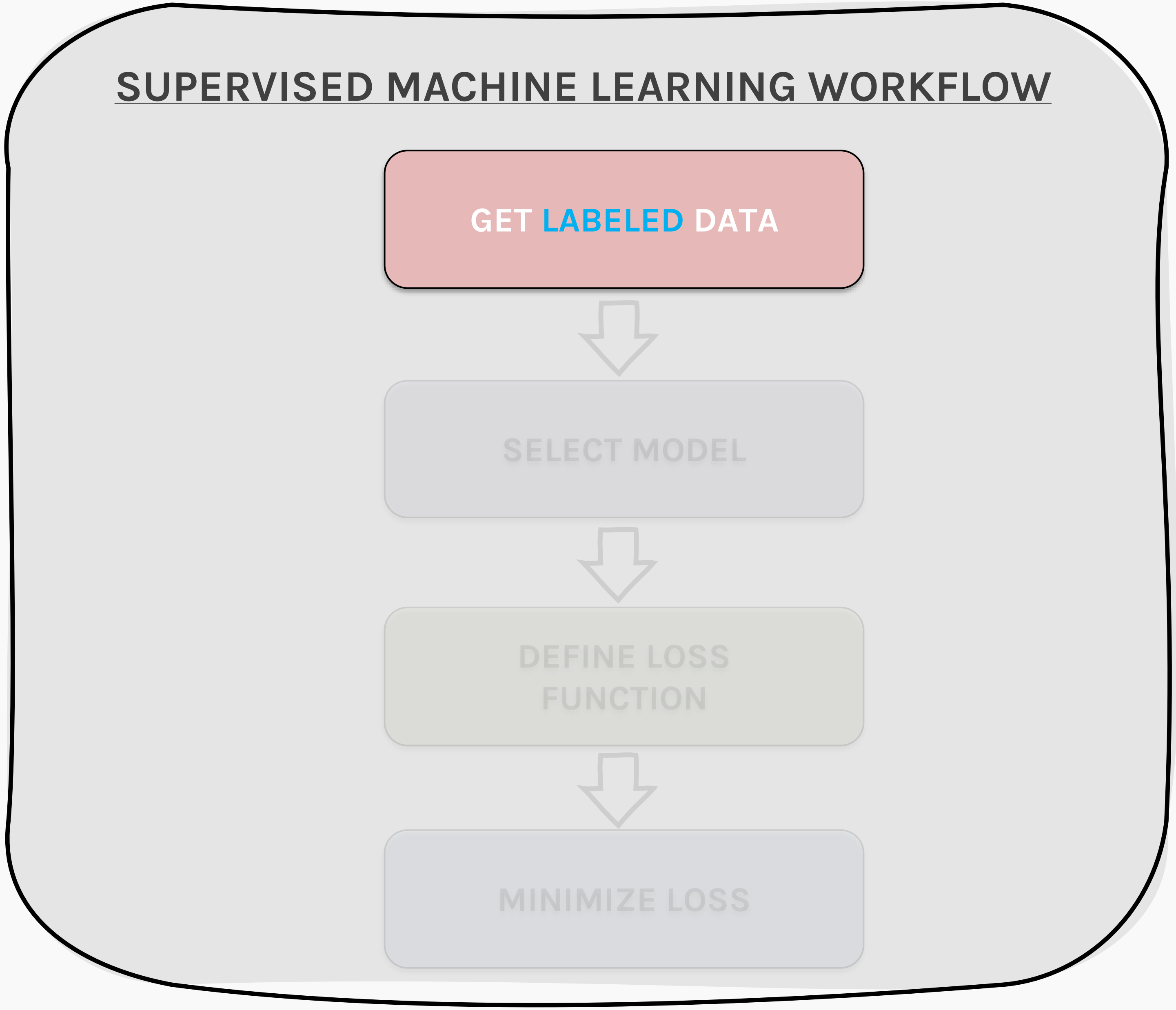
Supervised Learning: Learns with “labeled” data

Unsupervised Learning: Learns by clustering or association

Building blocks of **supervised** machine learning

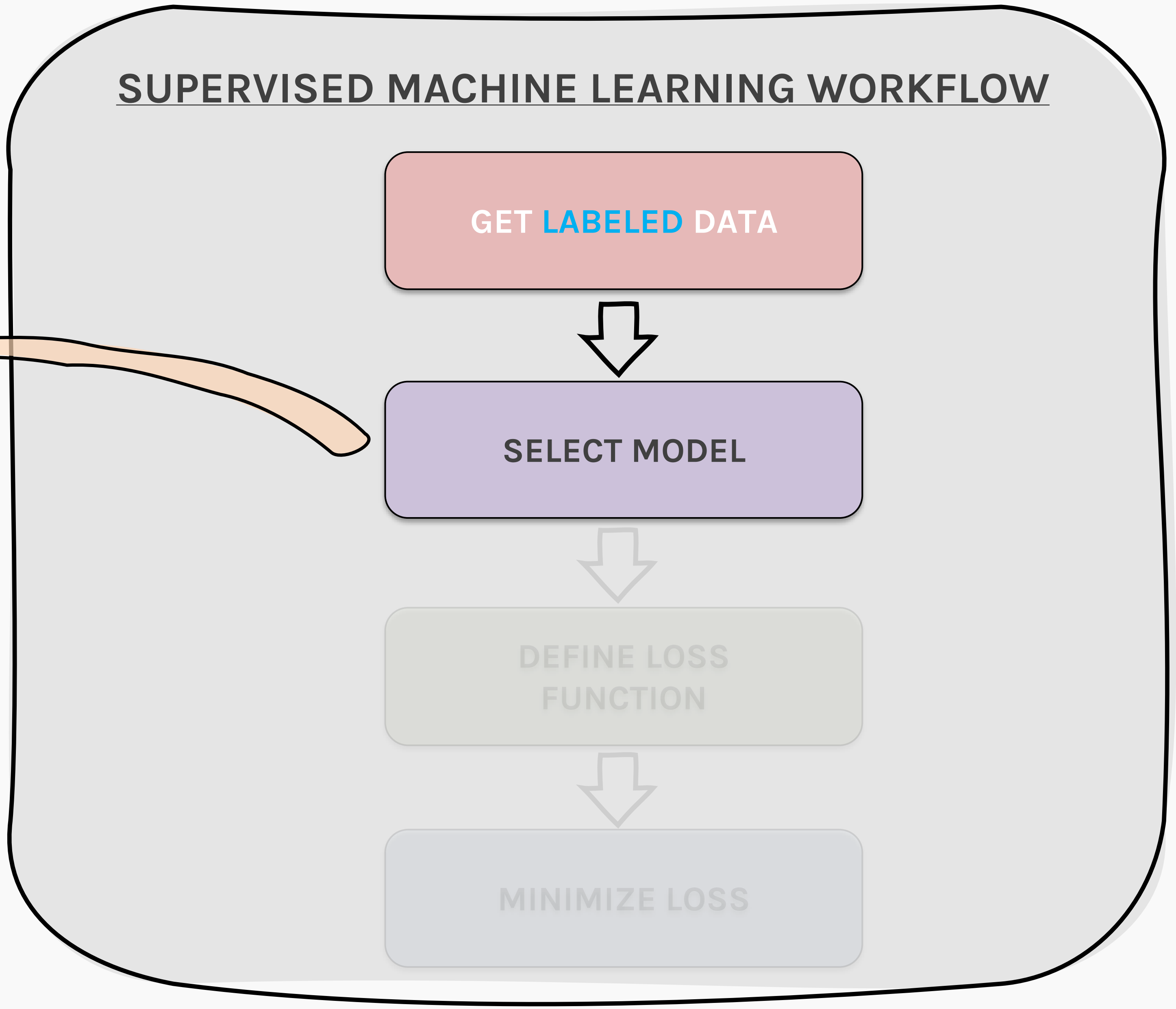


Building blocks of **supervised** machine learning



Building blocks of **supervised** machine learning

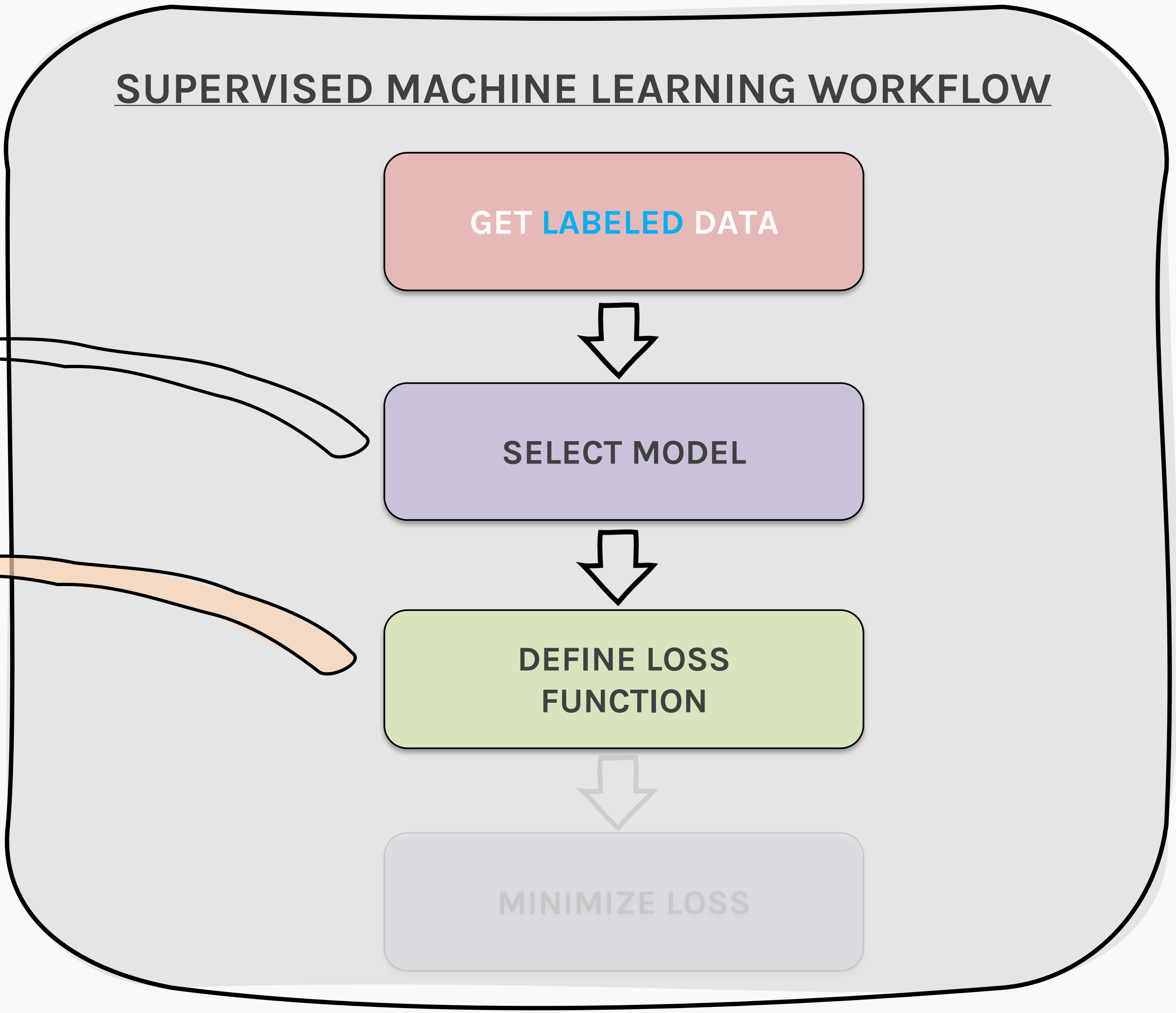
Regression, Classification, Object Detection



Building blocks of **supervised** machine learning

Regression, Classification, Object Detection

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

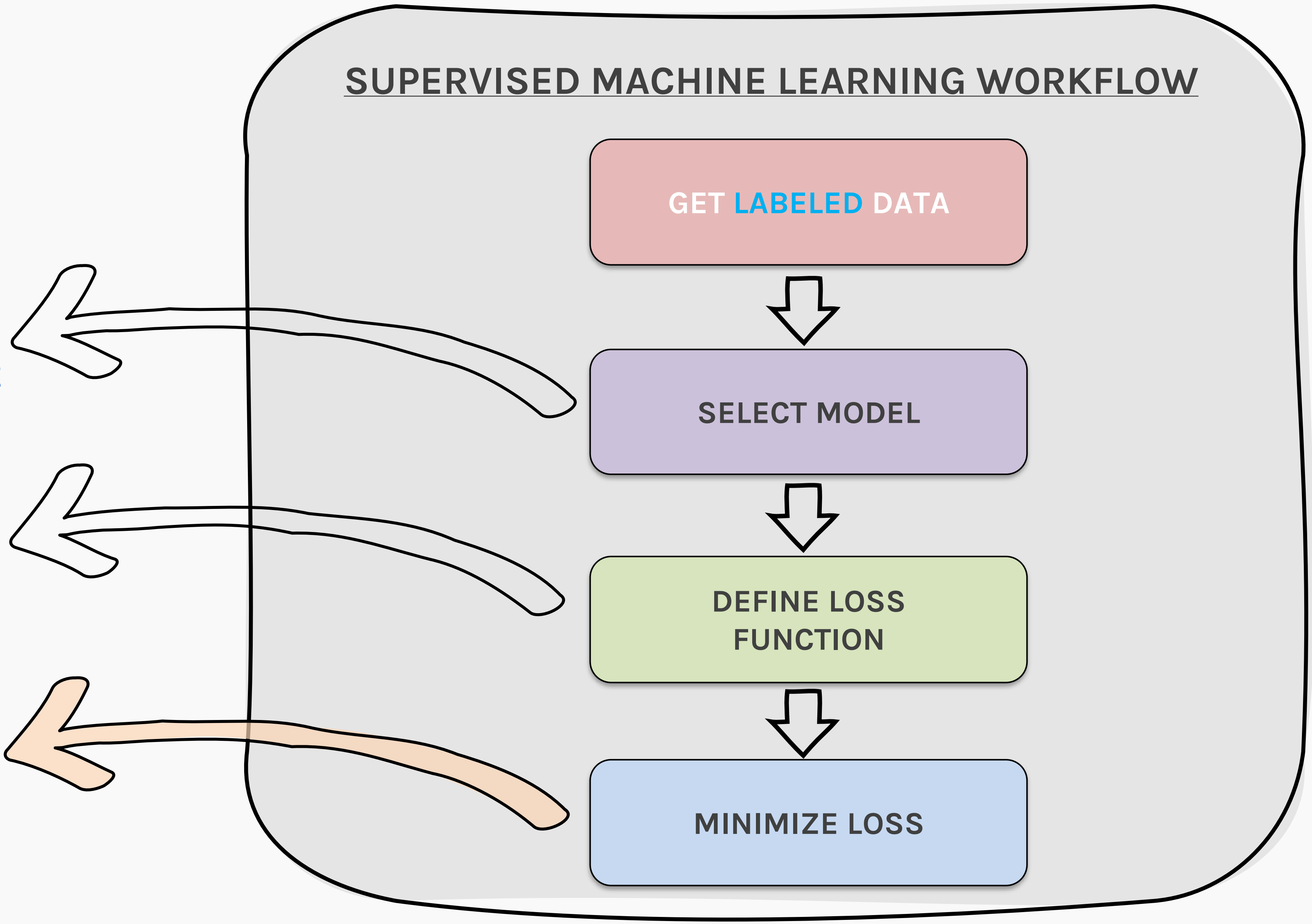


Building blocks of **supervised** machine learning

Regression, Classification, Object Detection

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

Analytical solution, Gradient Descent



Outline

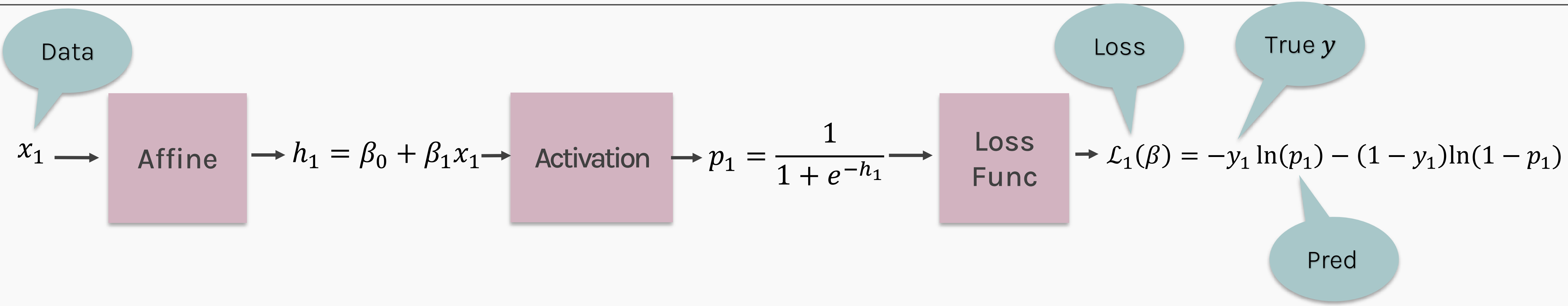
- Introduction
- Review of basic concepts
- **Perceptron - Single neuron network**
- Multi-Layer Perceptron (MLP)

Logistic Regression Revisited

Before we understand what, a perceptron is, let's look at a machine learning model which we had talked about in a previous lecture.

Logistic Regression

Logistic Regression Revisited

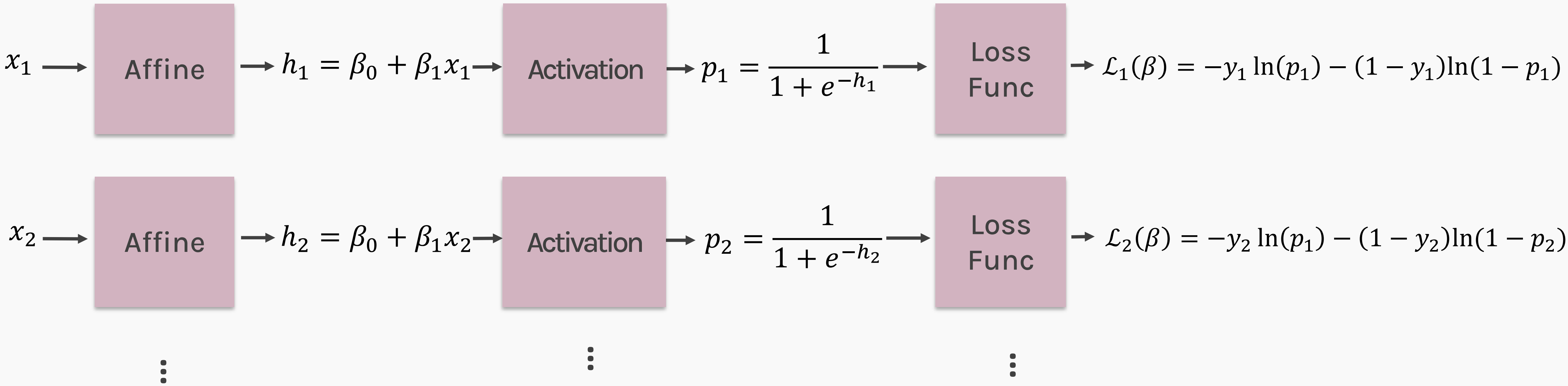


Logistic Regression Revisited

$x_1 \rightarrow$ 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 Revisited



Logistic Regression Revisited

$x_1 \rightarrow$ 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 $\rightarrow \mathcal{L}_2(\beta) = -y_2 \ln(p_1) - (1-y_2)\ln(1-p_2)$

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

$x_n \rightarrow$ Affine $\rightarrow h_n = \beta_0 + \beta_1 x_n \rightarrow$ Activation $\rightarrow p_n = \dfrac{1}{1+e^{-h_n}} \rightarrow$ Loss Func $\rightarrow \mathcal{L}_n(\beta) = -y_n \ln(p_n) - (1-y_n)\ln(1-p_{1n})$

Logistic Regression Revisited

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

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

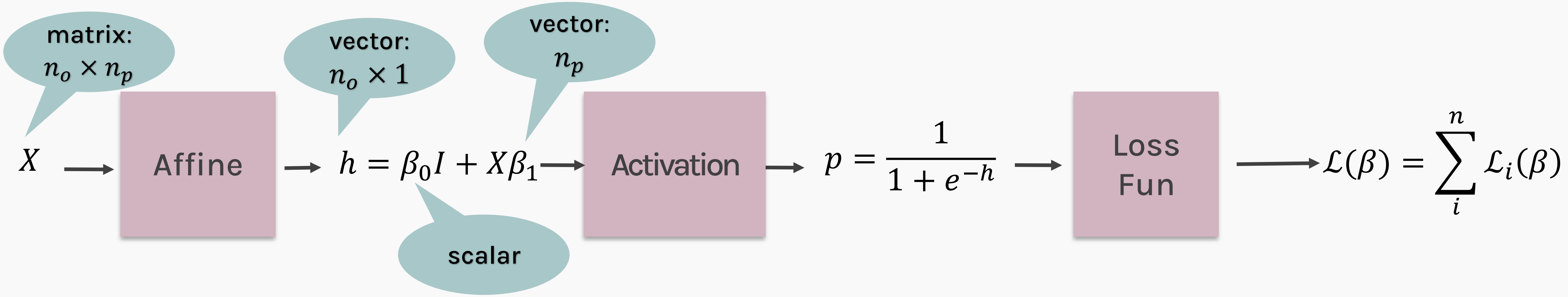
$\vdots$

$x_n \longrightarrow$ **Affine** $\longrightarrow h_n = \beta_0 + \beta_1 x_n \longrightarrow$ **Activation** $\longrightarrow p_n = \dfrac{1}{1 + e^{-h_n}} \longrightarrow$ **Loss Func** $\longrightarrow \mathcal{L}_n(\beta) = -y_n \ln(p_n) - (1 - y_n)\ln(1 - p_{1n})$

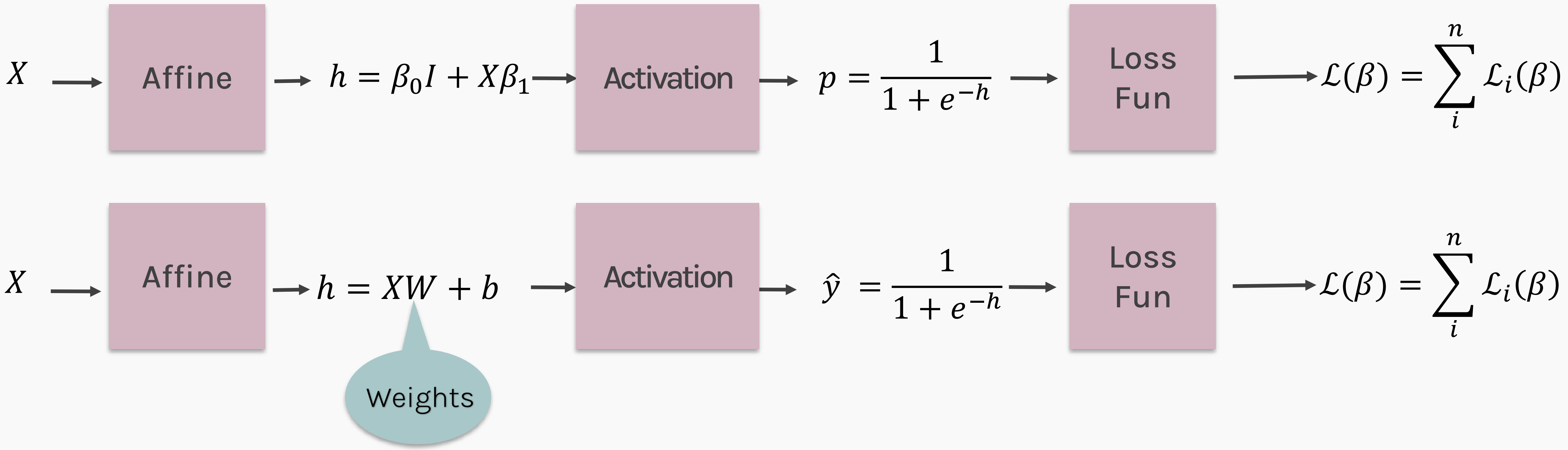
$$\boxed{\mathcal{L}(\beta) = \sum_{i}^{n} \mathcal{L}_i(\beta)}$$

Build our first ANN

n_p: number of predictors
n_o: number of observations



Build our first ANN



Build our first ANN

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

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

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

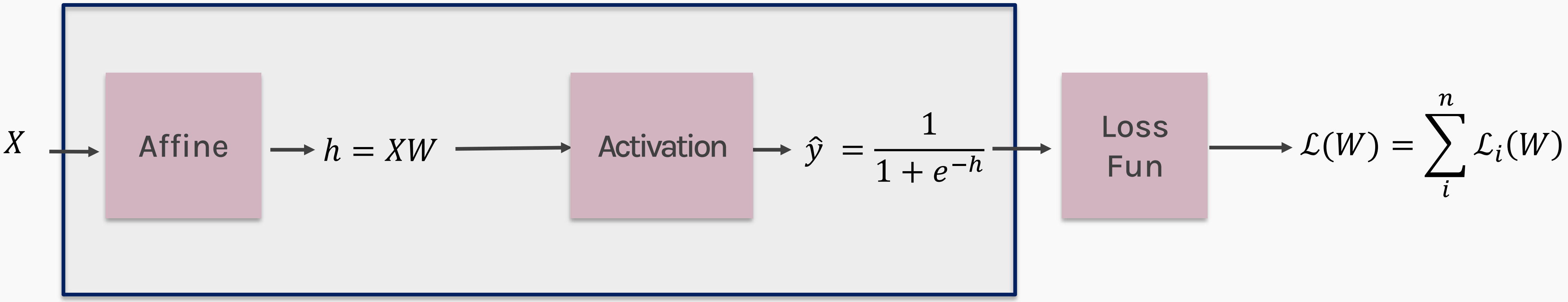
$$X = \begin{bmatrix} 1 & X_{11} & \dots\ X_{1p} \\ 1 & \vdots & \dots\ \ \vdots \\ 1 & X_{o1} & \dots\ X_{op} \end{bmatrix} \quad \mathrm{W} = \begin{bmatrix} b \\ W_1 \\ \vdots \\ W_p \end{bmatrix}$$

$\boxed{b \text{ is included in W}}$

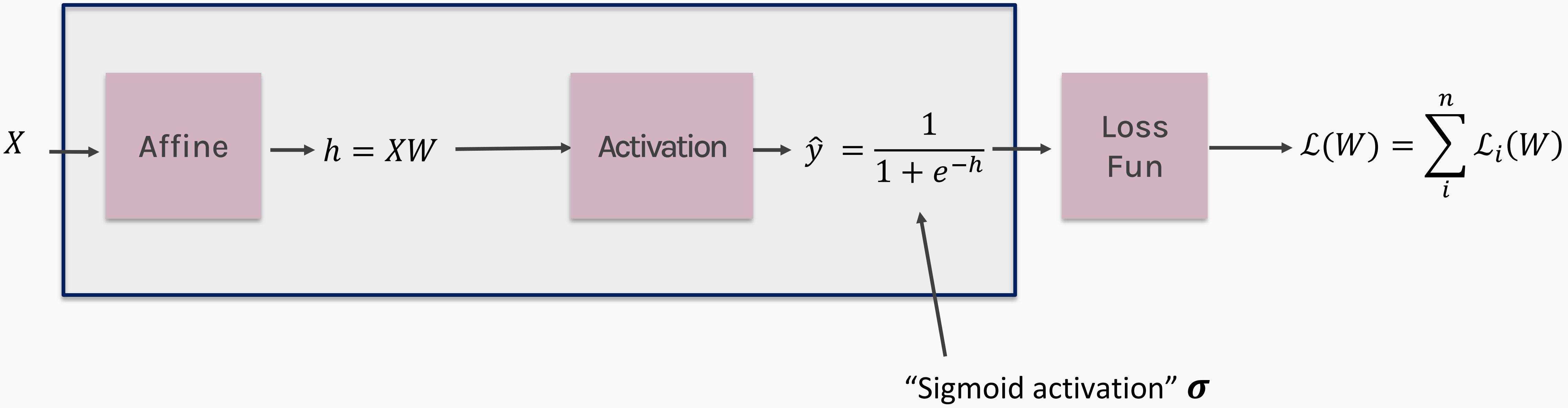
Build our first ANN

$$X \longrightarrow \boxed{\text{Affine}} \longrightarrow h = XW \longrightarrow \boxed{\text{Activation}} \longrightarrow \hat{y} = \frac{1}{1+e^{-h}} \longrightarrow \boxed{\text{Loss Fun}} \longrightarrow \mathcal{L}(W) = \sum_{i}^{n} \mathcal{L}_i(W)$$

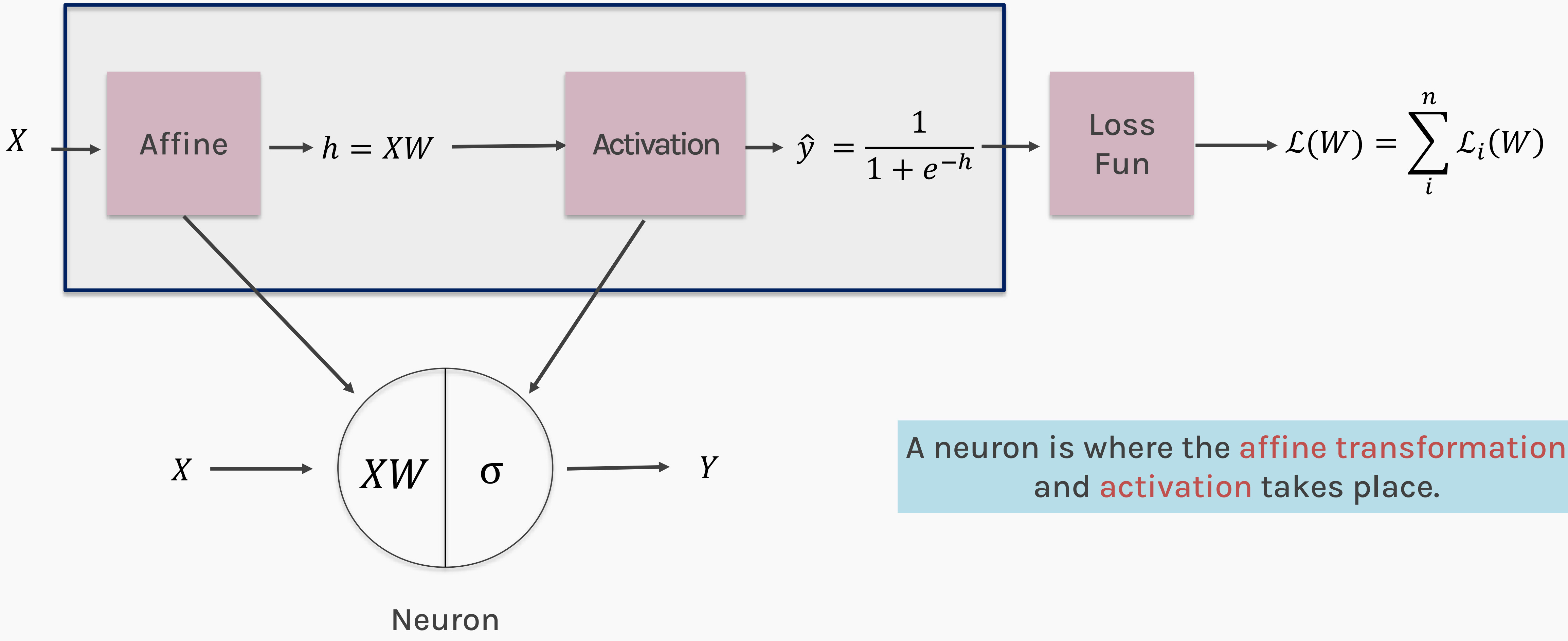
Build our first ANN



Build our first ANN

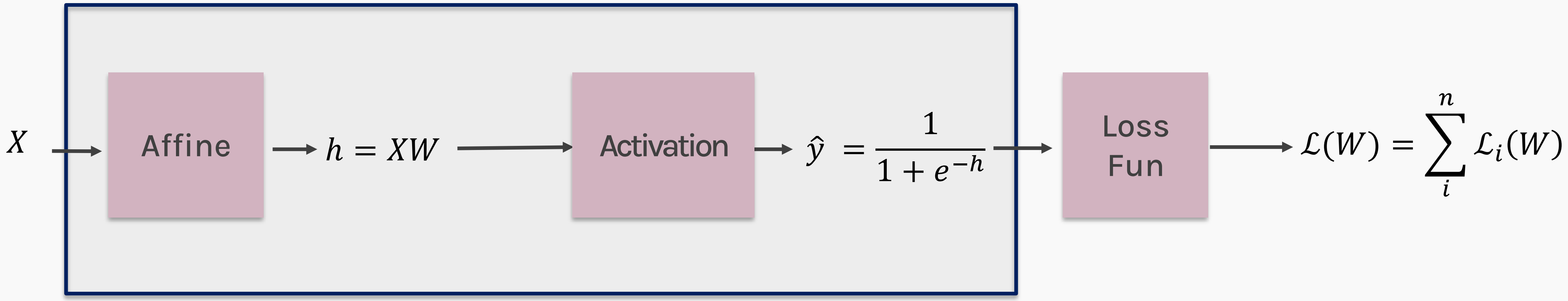


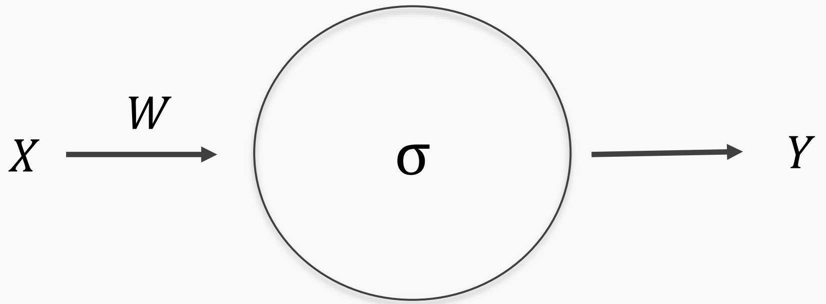
Build our first ANN



A neuron is where the affine transformation and activation takes place.

Build our first ANN



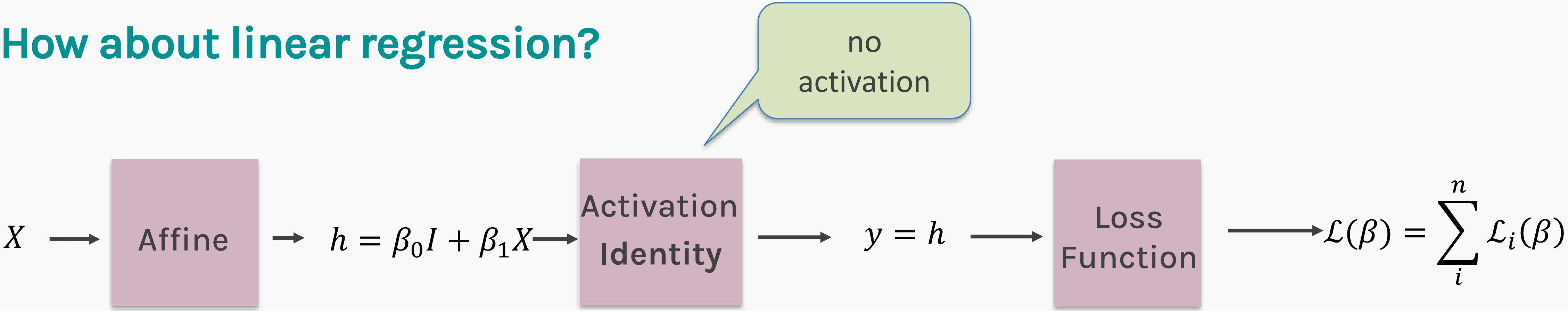


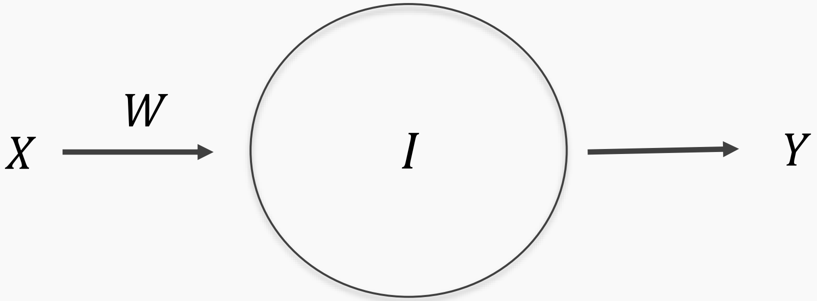
Single Neuron Neural "Network"

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