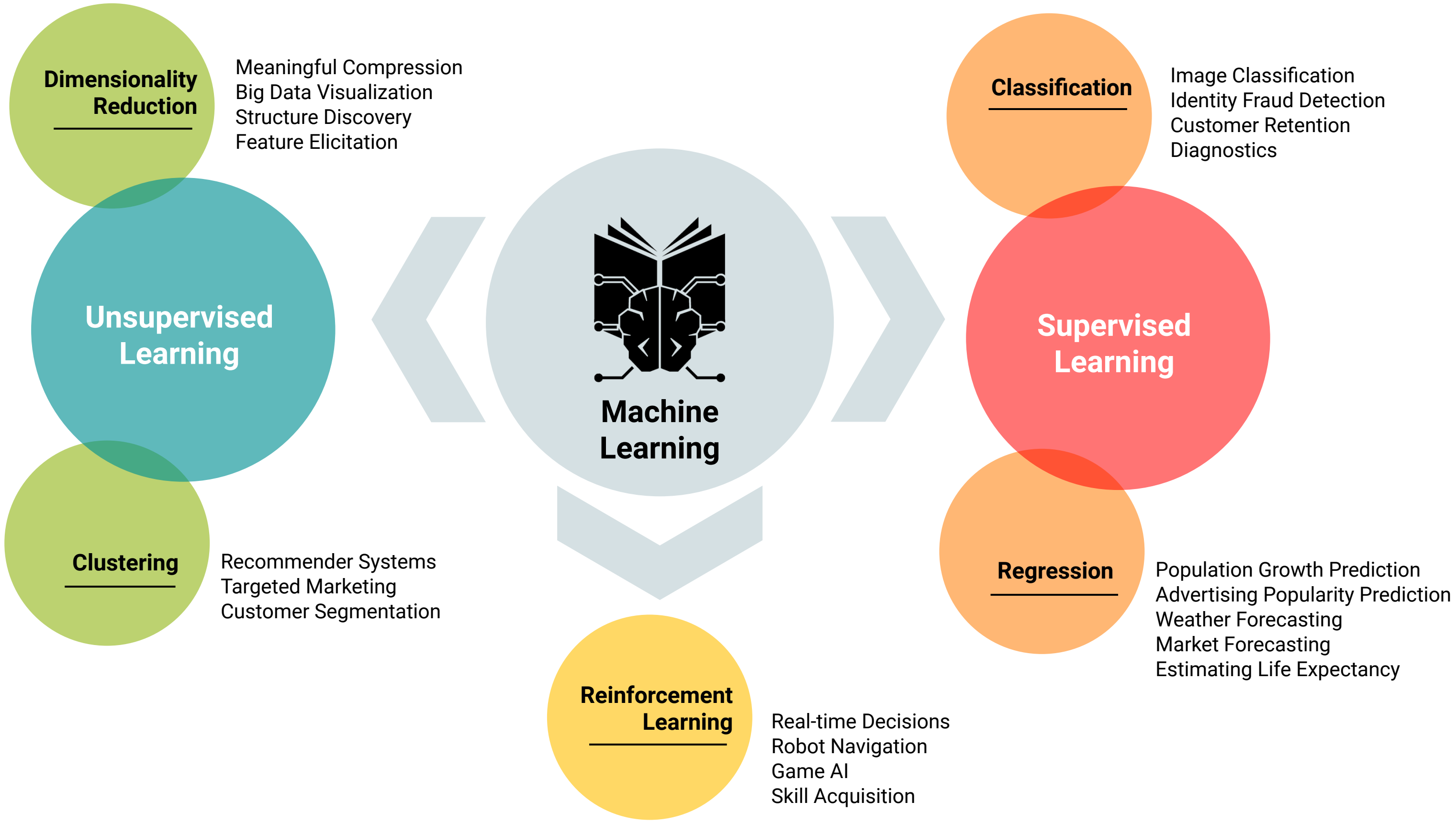
Neural Networks

FinTech
Lesson 14.1

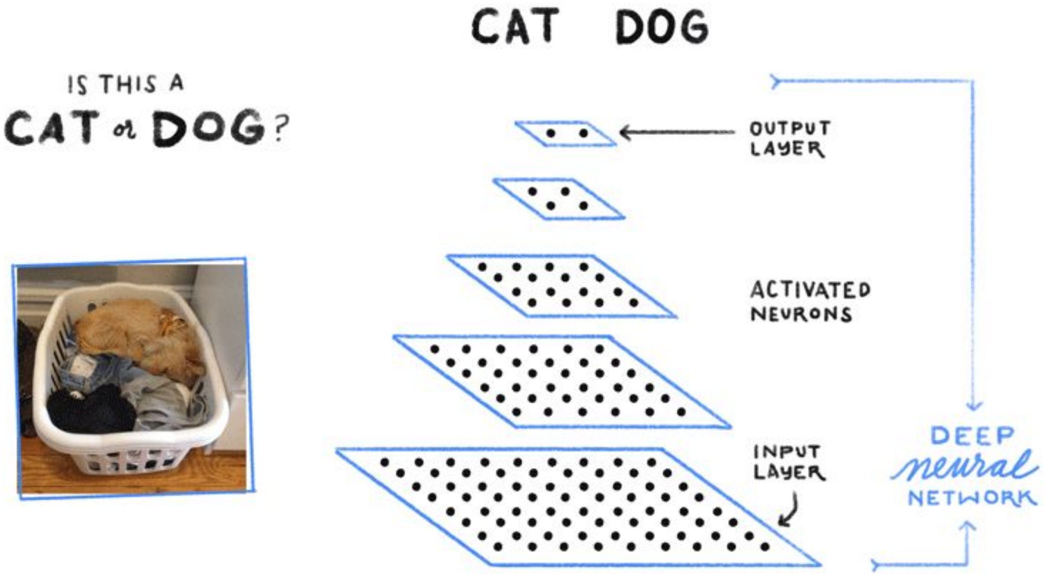
Machine Learning: Neural Networks



What’s a neural network?

Neural Networks

Neural networks are a set of algorithms that are modeled after the human brain. They're designed to recognize patterns and interpret sensory data through a kind of machine perception, labeling, or clustering raw input.



Neural Networks Are Cool!

Neural Networks

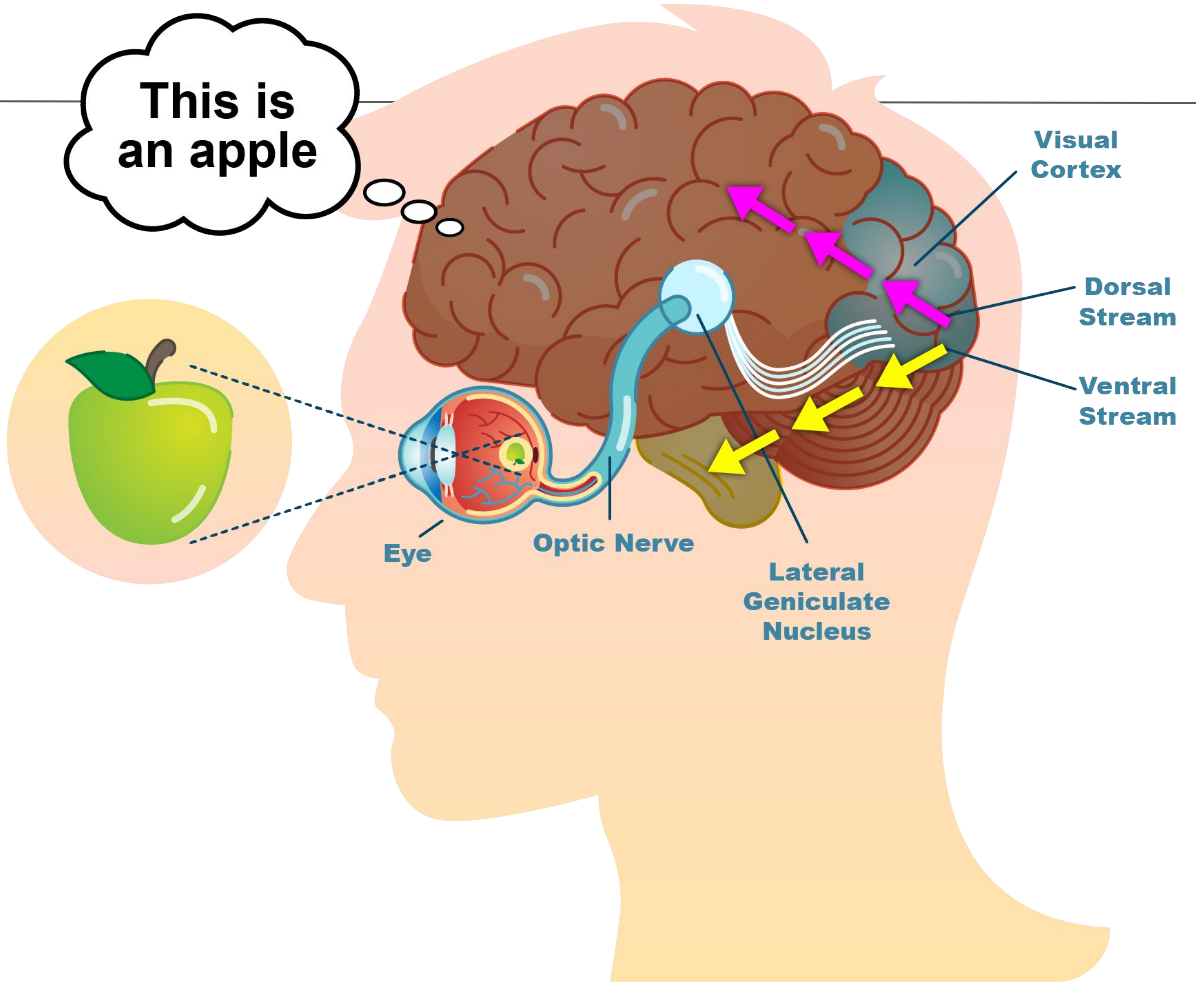
What do you see in this image?

Rorschach Blot

Neural Networks

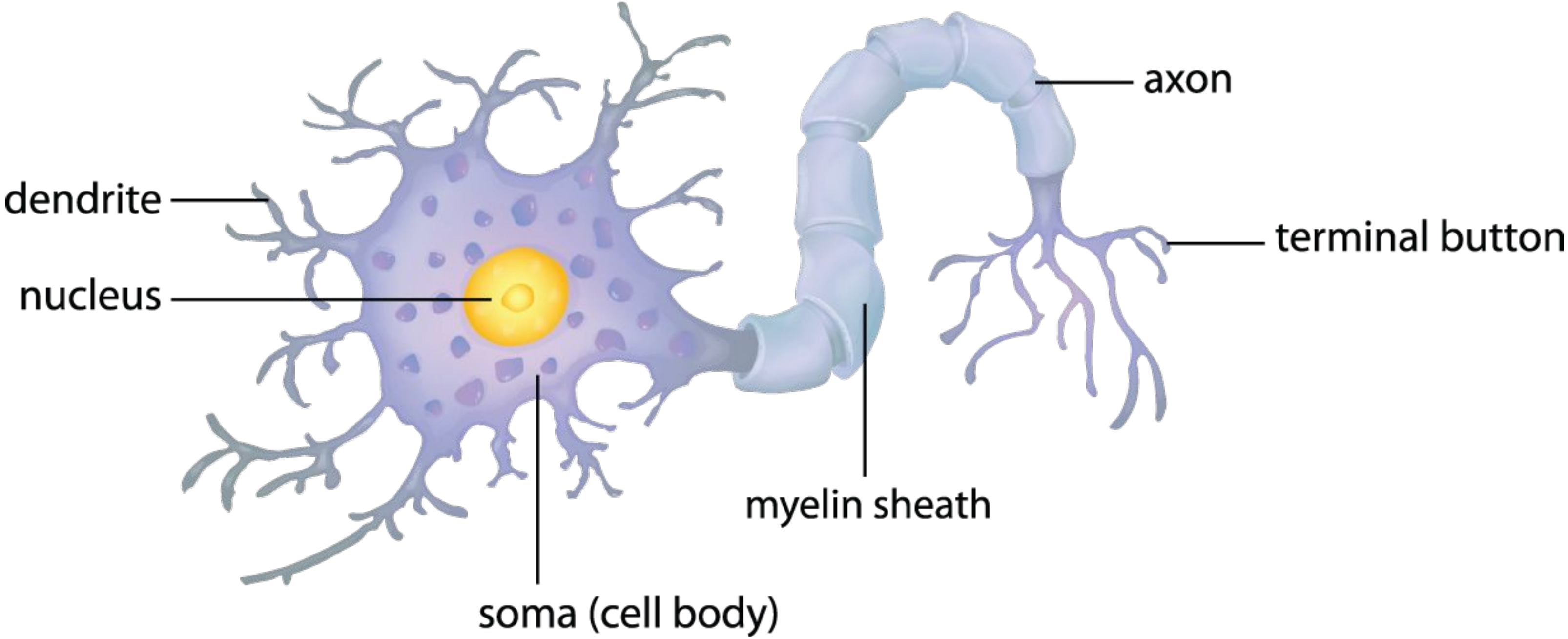
How our brain works

In order to recognize an image, our brain uses thousands of neuron connections to find a match between the visual input and a mental representation of an object.



Neural Networks

This power of our brain to process information and make predictions and interpretations inspired neurophysiologists and mathematicians to start the development of artificial neural networks (ANN).



The History of Neural Networks

Neural Networks

The History of Neural Networks: The 1940s to the 1970s

1943

Neurophysiologist **Warren McCulloch** and mathematician **Walter Pitts** wrote a paper on how neurons might work.

1949

Donald Hebb wrote *The Organization of Behavior*, which pointed out the fact that neural pathways are strengthened each time they are used.

1959

Bernard Widrow and **Marcian Hoff** of Stanford developed models called ADALINE and MADALINE.

1962

Widrow & **Hoff** developed a learning procedure that examines the value before the weight adjusts it (i.e., 0 or 1) according to the rule: Weight Change = (Pre-Weight line value).

1972

Teuvo Kohonen and **James A. Anderson** each developed a similar network independently of one another. They both used matrix mathematics to describe their ideas but did not realize that what they were doing was creating an array of analog ADALINE circuits.

(Source)

Neural Networks

The History of Neural Networks: The 1980s to the present

1982

John Hopfield of Caltech presented a paper to the National Academy of Sciences. His approach was to create more useful machines by using bidirectional lines. Previously, the connections between neurons was only one way.

1982

Joint US-Japan conference on **Cooperative/ Competitive Neural Networks**. Japan announced a new Fifth Generation effort on neural networks, and US papers generated worry that the US could be left behind in the field.

1986

Three independent groups of researchers, including **David Rumelhart**, a former member of Stanford's psychology department, came up with similar ideas which are now called back propagation networks.

1997

A recurrent neural network framework, LSTM was proposed by **Jürgen Schmidhuber** & **Sepp Hochreiter**.

Now

Neural networks discussions are prevalent; the future is here!

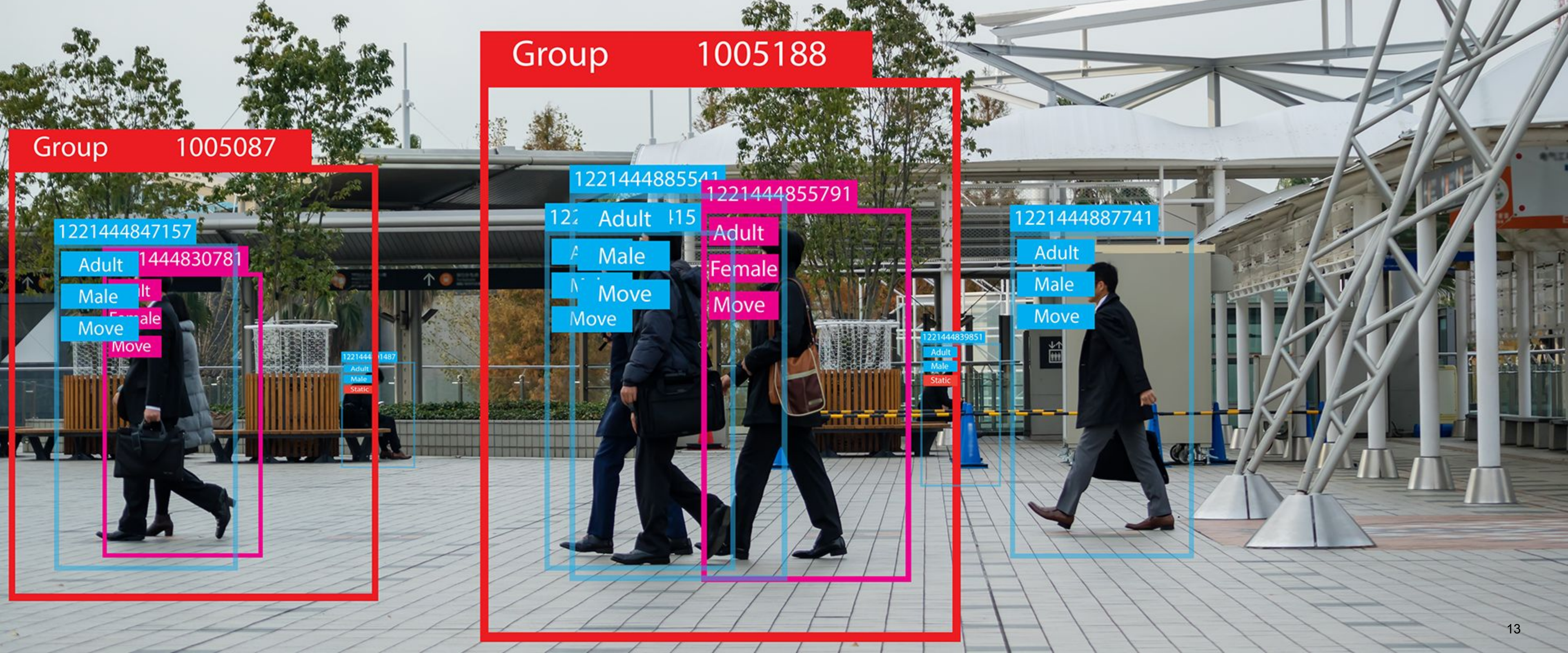
Examples:
self driving cars, algorithmic trading, robo advisors, customized marketing offers.

(Source 1; Source 2)

The Future Is Here: Applications of Neural Nets

Applications of Neural Nets

Neural networks are here to stay, and applications become more common every day.



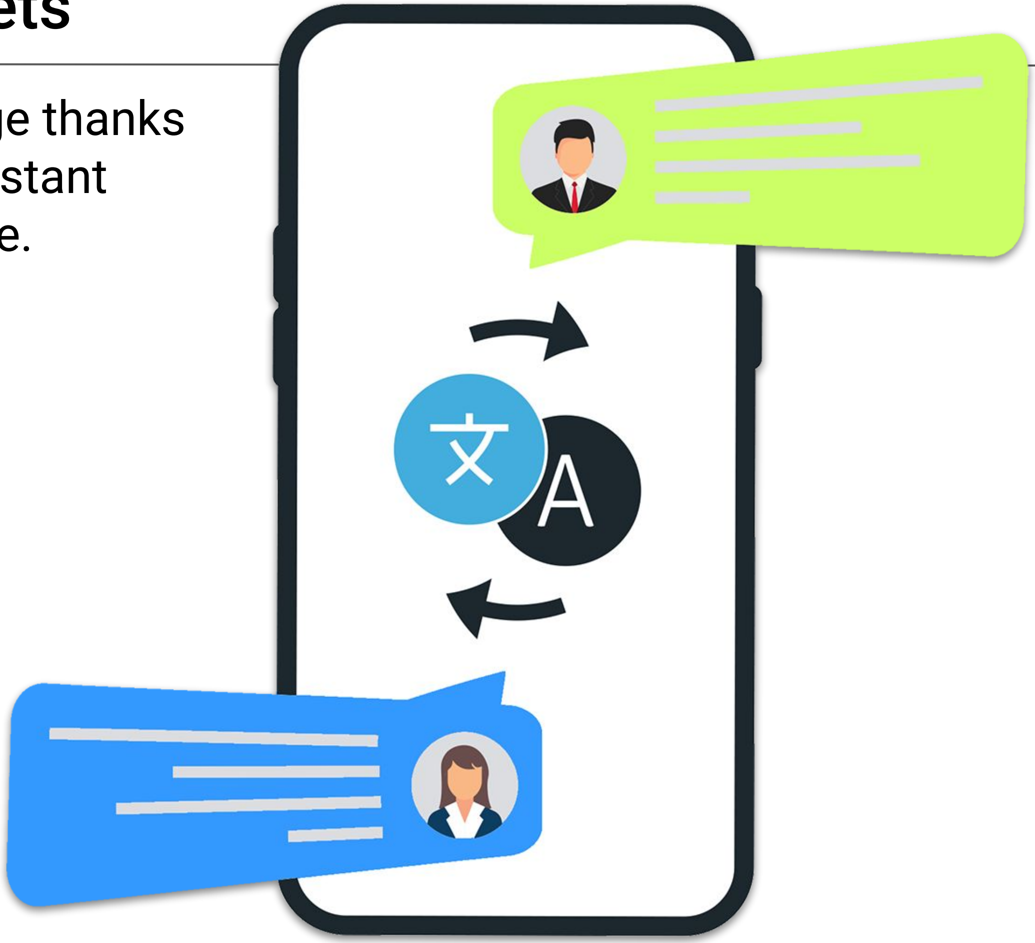
Applications of Neural Nets

Companies like Google and Tesla are using neural networks to develop self-driving cars.

tesla.com

Applications of Neural Nets

Today you can talk in any language thanks to the automated translation of instant messaging applications like Skype.



Applications of Neural Nets

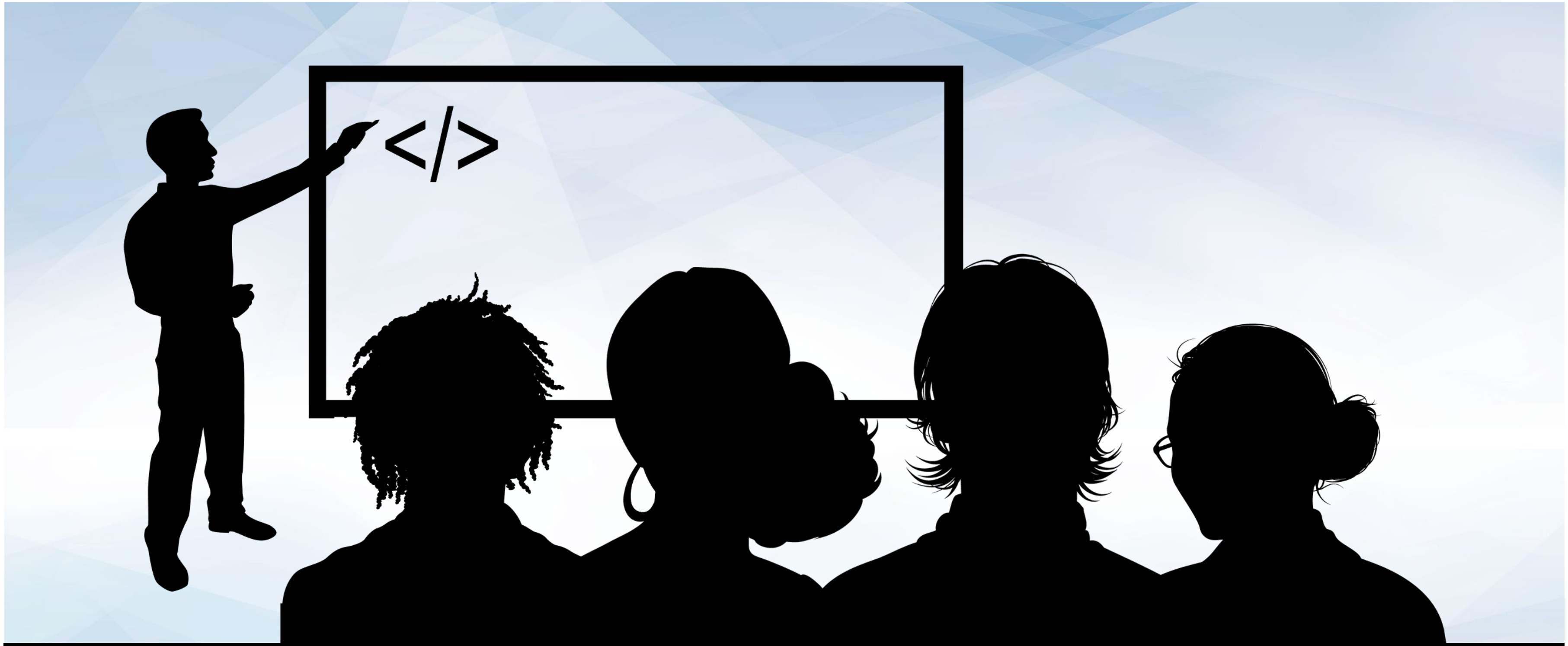
Memories from the past that were captured in black and white can take a new perspective through automatic image colorization.

Applications of Neural Nets

Neural networks are also used to create a better world, not only for humans but also for wildlife thanks to projects like the one led by the U.S. National Oceanic and Atmospheric Administration where they are saving whales by tracking the north Atlantic right whales population using neural networks for image recognition.

Neural Networks Applications in Finance

- Credit card fraud detection
- Foreign-Exchange-Rate forecasting
- Risk Management
- Algorithmic Trading
- Automated invoices processing
- Preventing Money Laundering



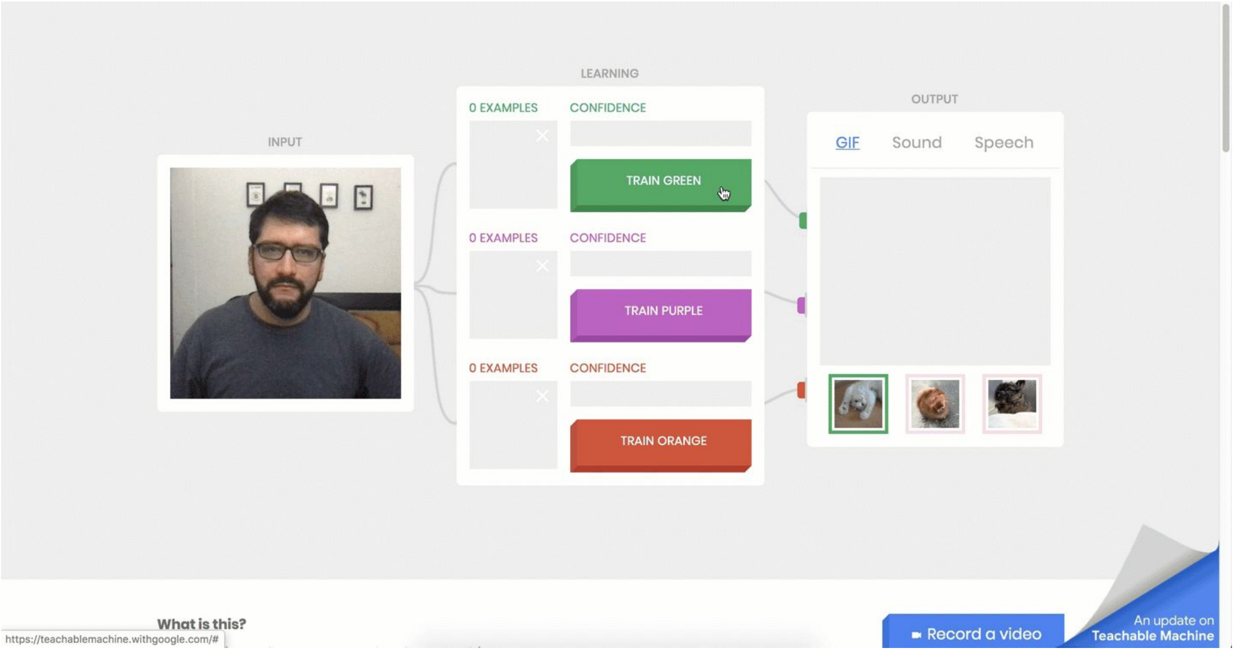
Instructor Demonstration

Teachable Machine

Teachable Machine in Action

Teachable Machine is an experiment that makes it easier for anyone to start exploring how machine learning works.

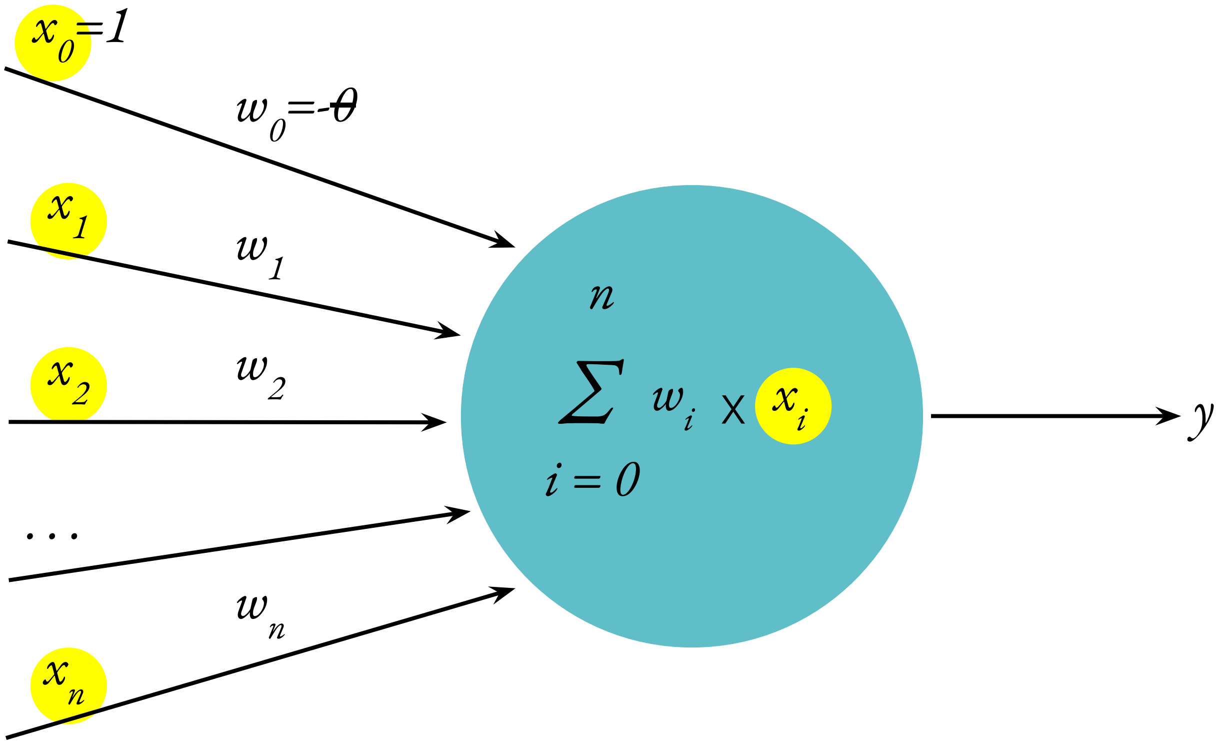
It lets you teach a machine using your camera—live in the browser, no coding required.



Anatomy of a Neuron

Anatomy of a Neuron

The perceptron mimics the functioning of a biological neuron, it receives input data signals (Xn) that can take boolean or numeric values.



How Perceptron Works

How Perceptron Works

Consider the task of predicting whether or not a person would watch a random movie on Netflix using the behavioral data available.

How Perceptron Works

Let's assume the decision depends on 3 binary inputs (binary for simplicity).

01

`isNewRelease`

02

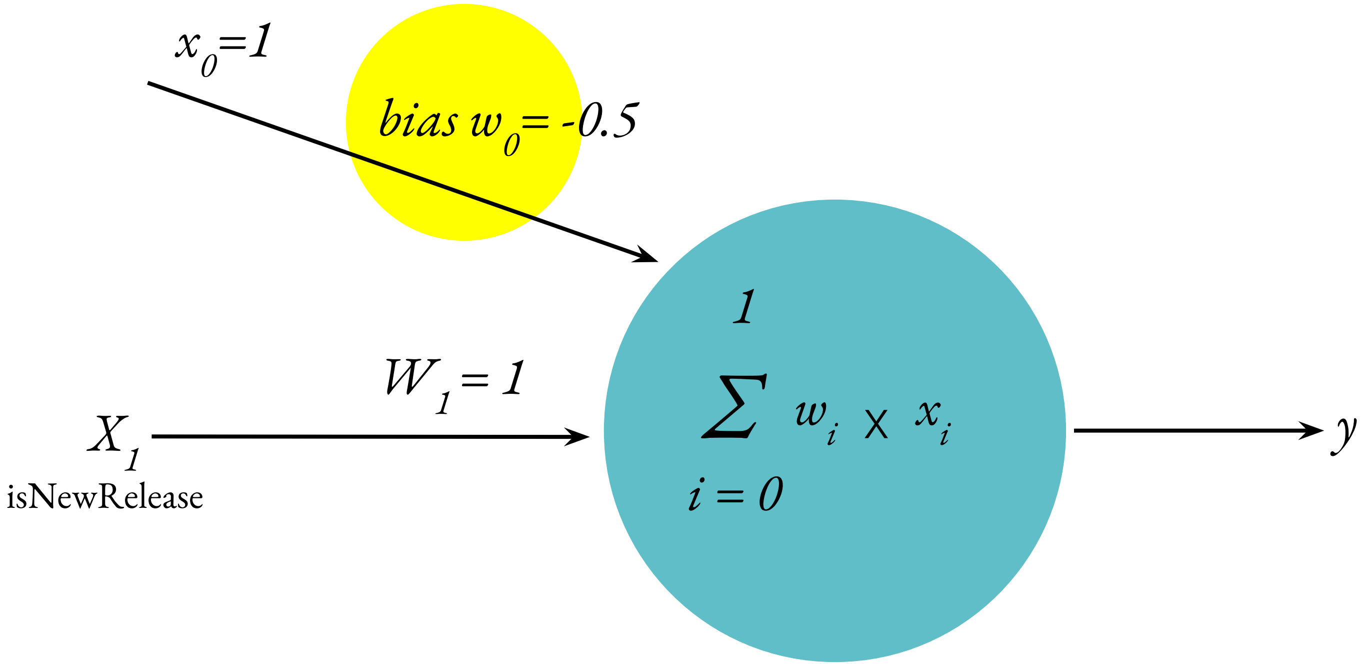
`isForChildren`

03

`isAwardWinning`

How Perceptron Works

In this perceptron model, w_0 (our arbitrary threshold θ) is called the bias because it represents the prejudice that can influence the final decision.



How Perceptron Works

A cinephile may have a low threshold and will be willing to watch any movie regardless of its genre, release date, or the awards the movie received.

$(\theta = 0)$

How Perceptron Works

In contrast, a father that wants to spend time with his daughter watching a movie may choose the latest children's film by default.

$(\theta = 2)$

How Perceptron Works

In the case of the father that wants to watch a movie with his daughter:

High Weight

`isNewRelease`

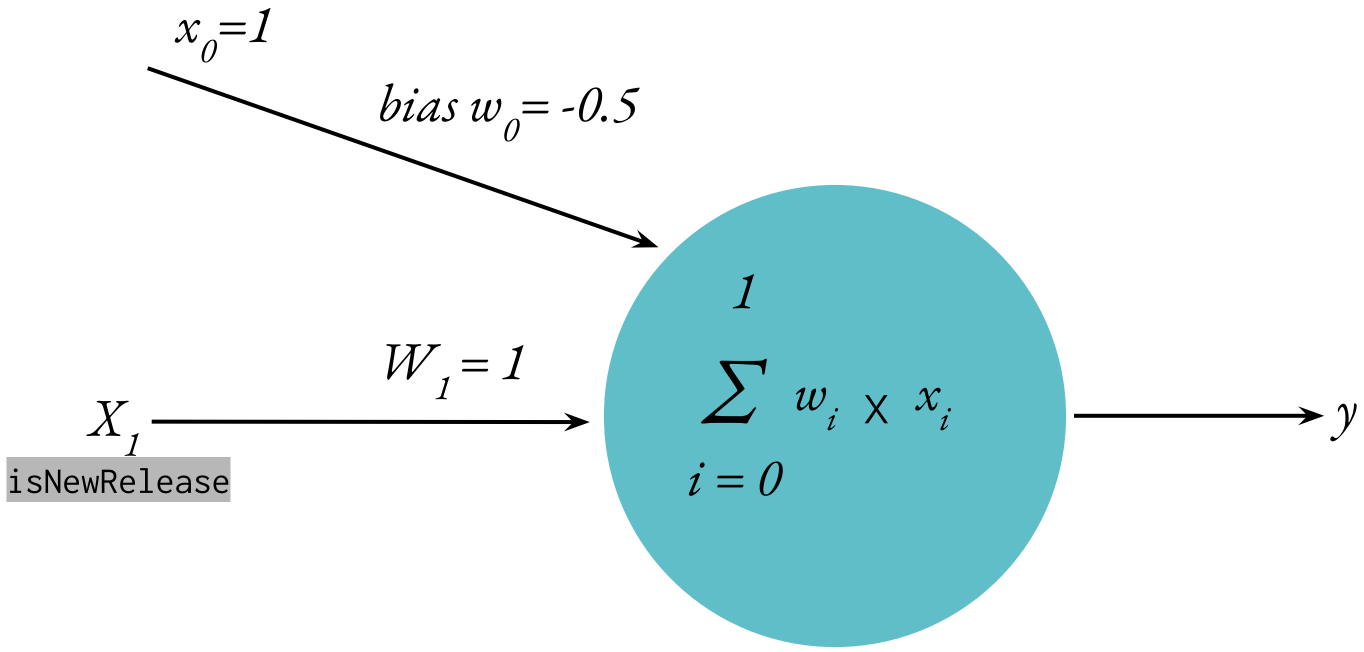
`isForChildren`

Penalize

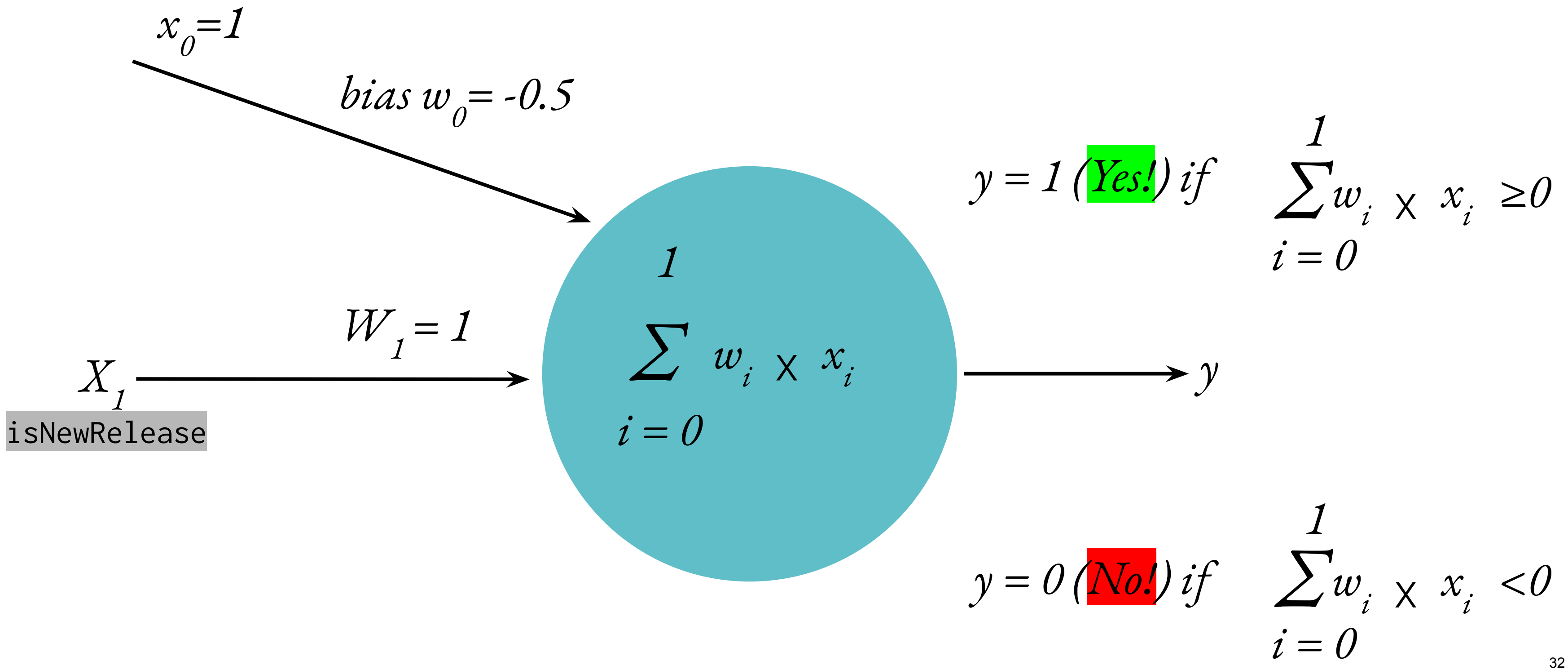
`isAwardWinning`

The weights and the bias
will depend on the
historical data

In real life, if you want to choose a movie, you're not as strict as the perceptron could be. For example, if you base your decision only in one input variable, such as `isNewRelease`, the bias is set to 0.5 and w1 = 1 then our perceptron would look as follows:



Using this model, the decision for a movie with `isNewRelease` = 1 will be Yes!, and the decision for a movie with `isNewRelease` = 0 will be No!.



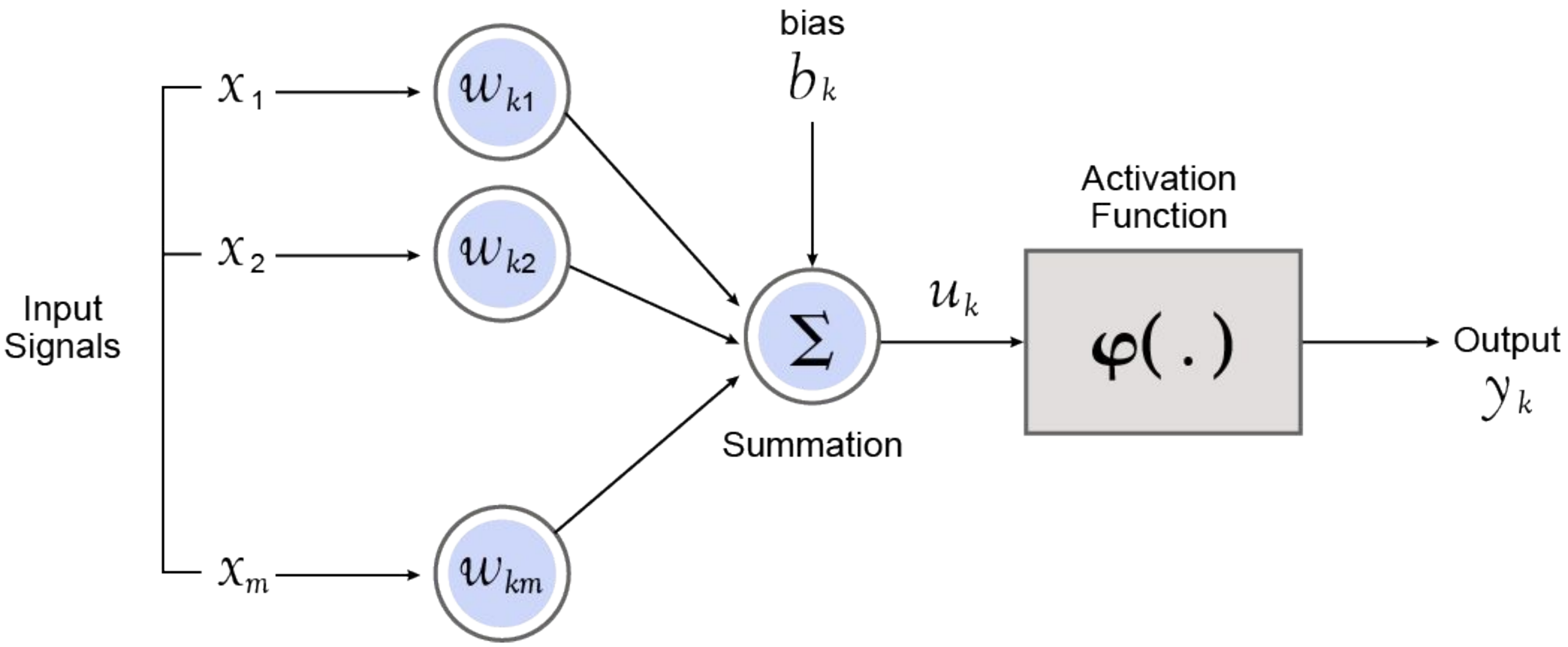
How Perceptron Works

This decision may be too harsh since we are losing the chance of enjoying classic movies, or the Oscar awarded films from last year; this is where the activation function comes to the scene.

Activation Function

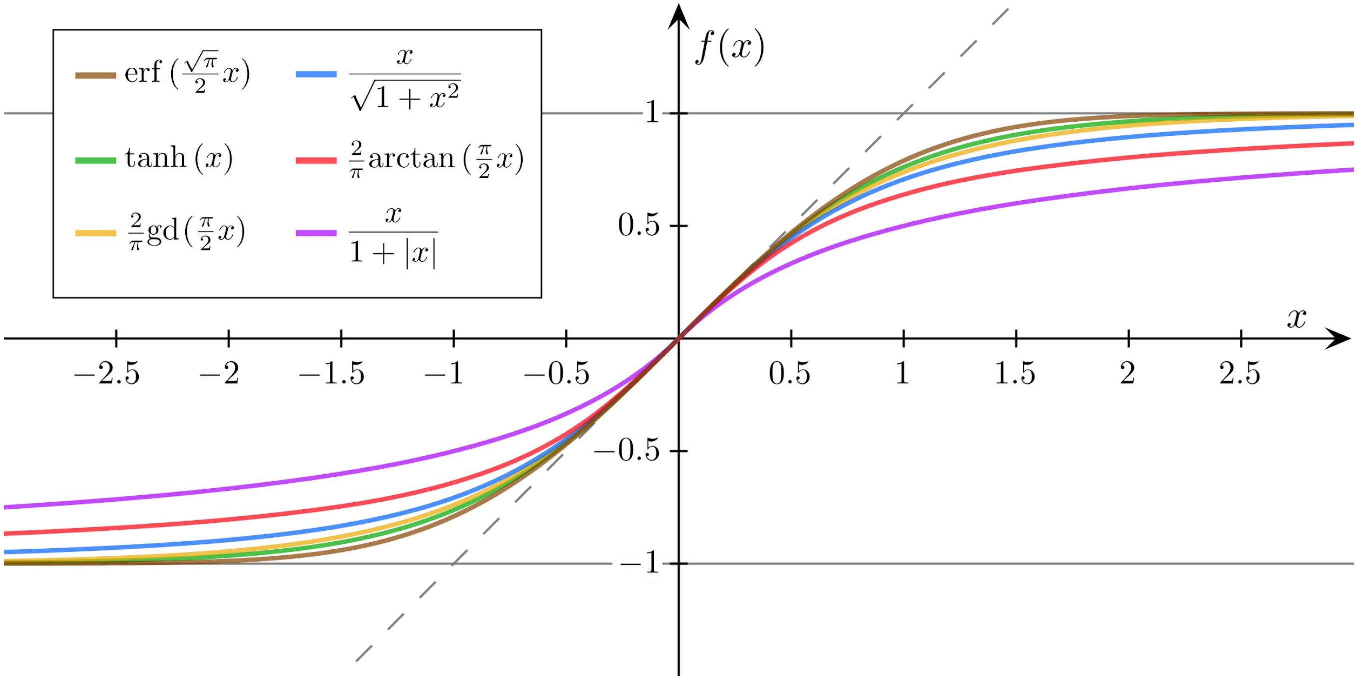
Introducing the Activation Function

Using an activation function, we normalize output to a range between -1 and 1, or 0 and 1, the latter being a representation of a probability.



Introducing the Activation Function

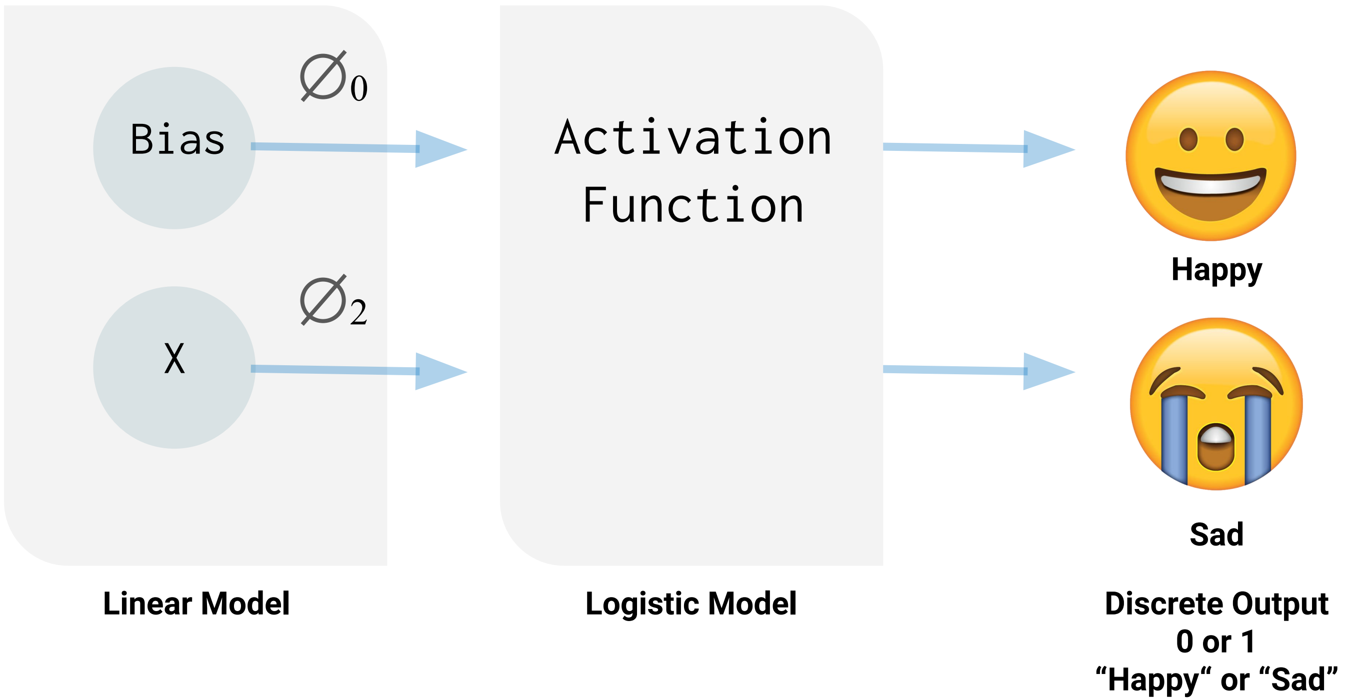
Using an activation function, we normalize output to a range between -1 and 1, or 0 and 1, the latter being a representation of a probability.

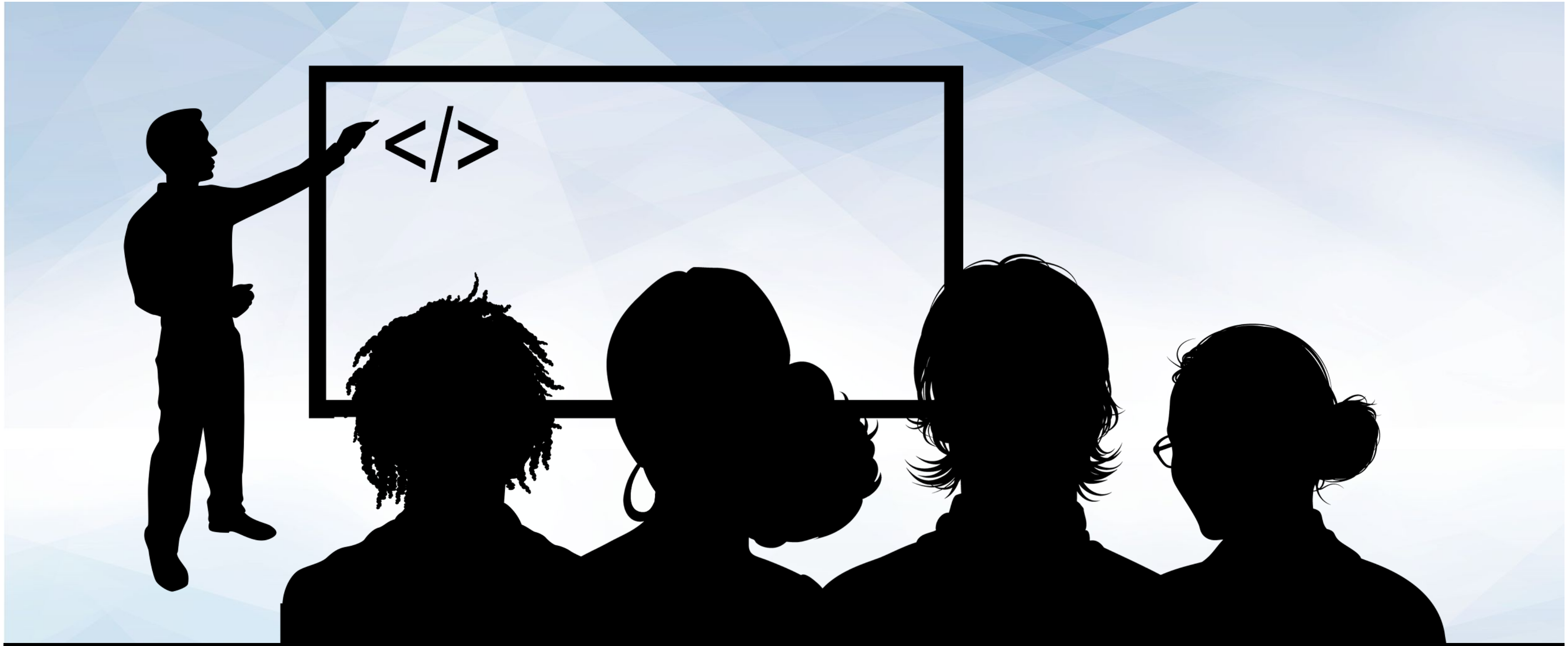


(Source)

Activation Function

Instead of yes/no decision, with an activation function, we get the probability of yes, similar to using logistic regression.



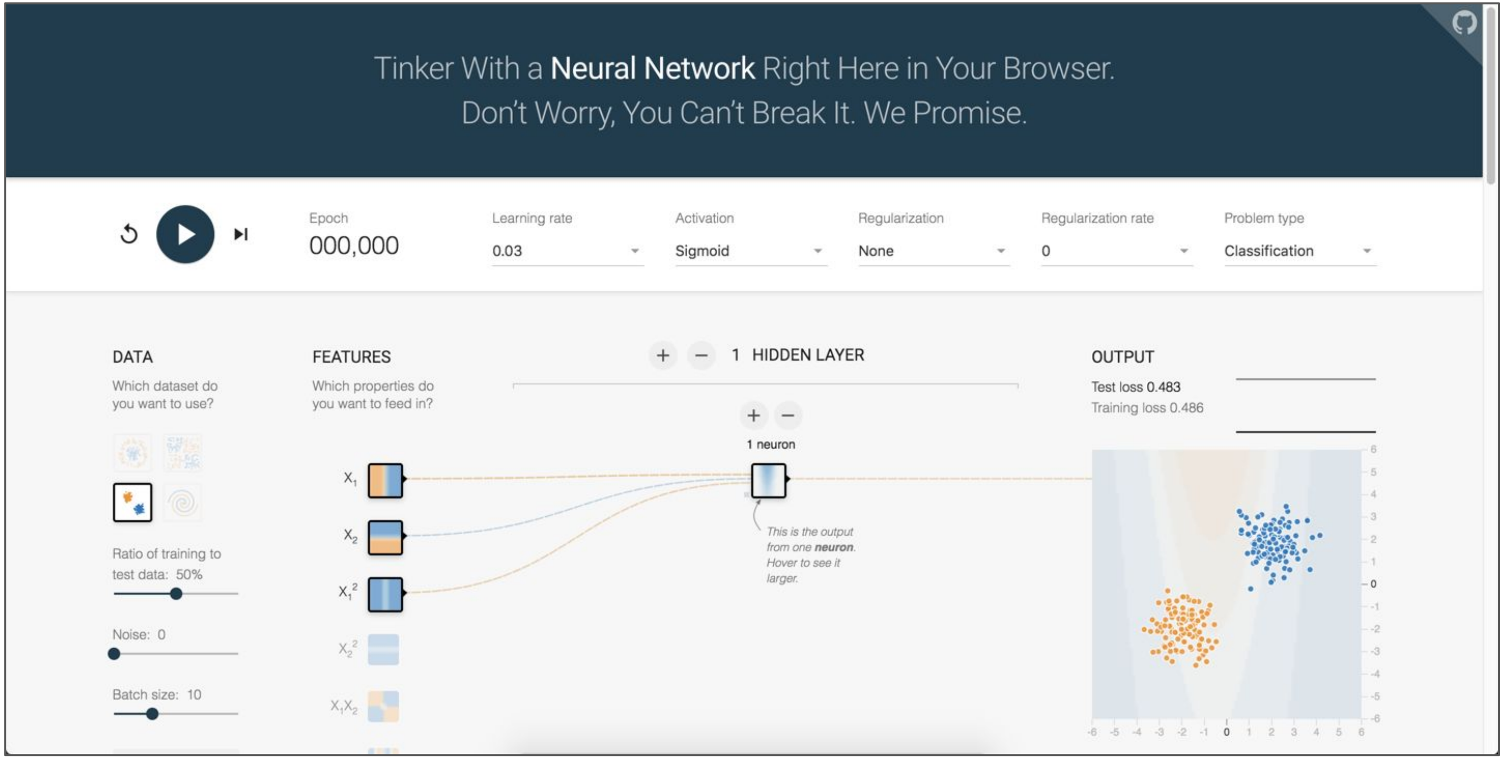


Instructor Demonstration

TensorFlow Playground

Introducing the Activation Function

Tensorflow Playground



(Source)

Activating Your First Artificial Neuron

Activating Your First Artificial Neuron

There are two ways to code a neuron:

01 New Code

- Complete the math and code it from scratch using Python, Pandas, and NumPy

02 Industry-Standard API

- Speed up your implementation
- Focus your efforts on improving your model
- Have a better understanding of the business problem you want to solve

Activating Your First Artificial Neuron

We are going to use [TensorFlow] and [Keras] to build our Neural Networks.

TensorFlow

TensorFlow is an end-to-end open-source platform for machine learning, that allows us to run our code across multiple platforms in a highly efficient way.

Keras

Keras is an abstraction layer on top of TensorFlow that makes it easier to build models. You can relate this to using Plotly Express to create charts instead of using the more verbose Matplotlib library.

Activating Your First Artificial Neuron

TensorFlow

- Machine learning library
- Used to build neural networks
- Runs on multiple platforms
- Supports distributed computing
- Allows detailed fine-tuning

Activating Your First Artificial Neuron

Why Keras

- It’s an API designed for human beings
- Runs on top of TensorFlow, CNTK, or Theano
- Facilitates to turn models into products
- It has broad adoption in the industry and the research community

Keras + TensorFlow

Keras allows interaction with TensorFlow through a simplified interface

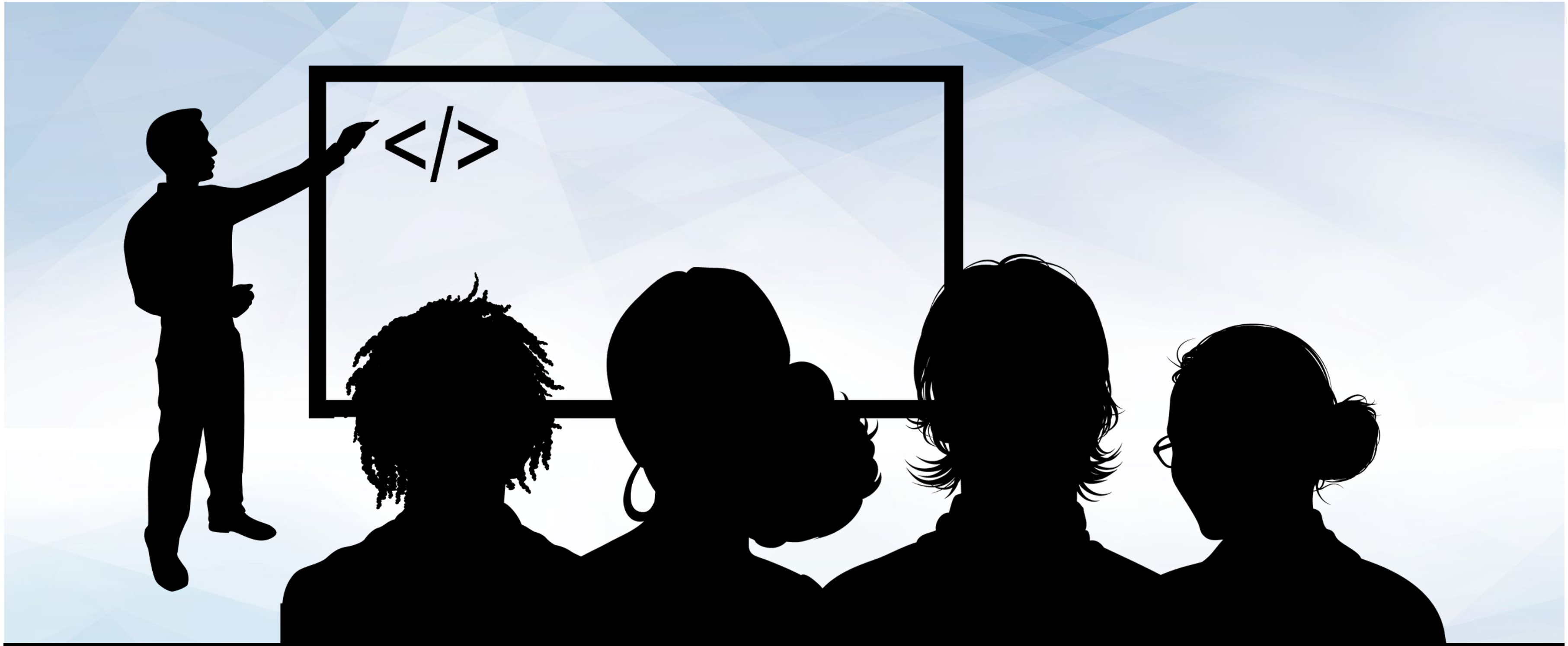
Model → Fit → Predict (with a few other steps)

```
from keras.models import Sequential

model = Sequential()
```

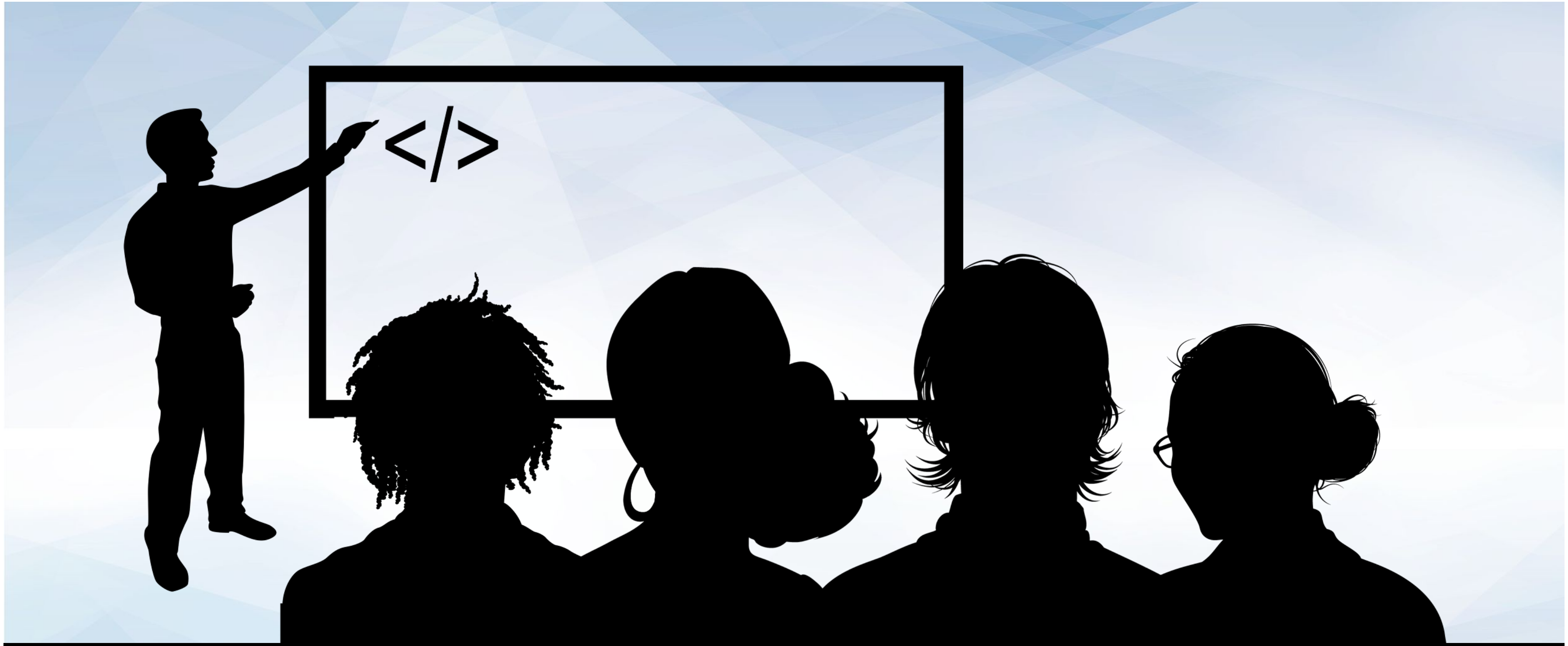
```
# x_train and y_train are Numpy arrays, just like in the Scikit-Learn API.
model.fit(x_train, y_train, epochs=5, batch_size=32)
```

```
classes = model.predict(x_test, batch_size=128)
```



Instructor Demonstration

Neural Network with a Single Neuron Using Keras



Instructor Demonstration

Connecting Neurons

How do you think we can improve the model’s accuracy?

Model Accuracy

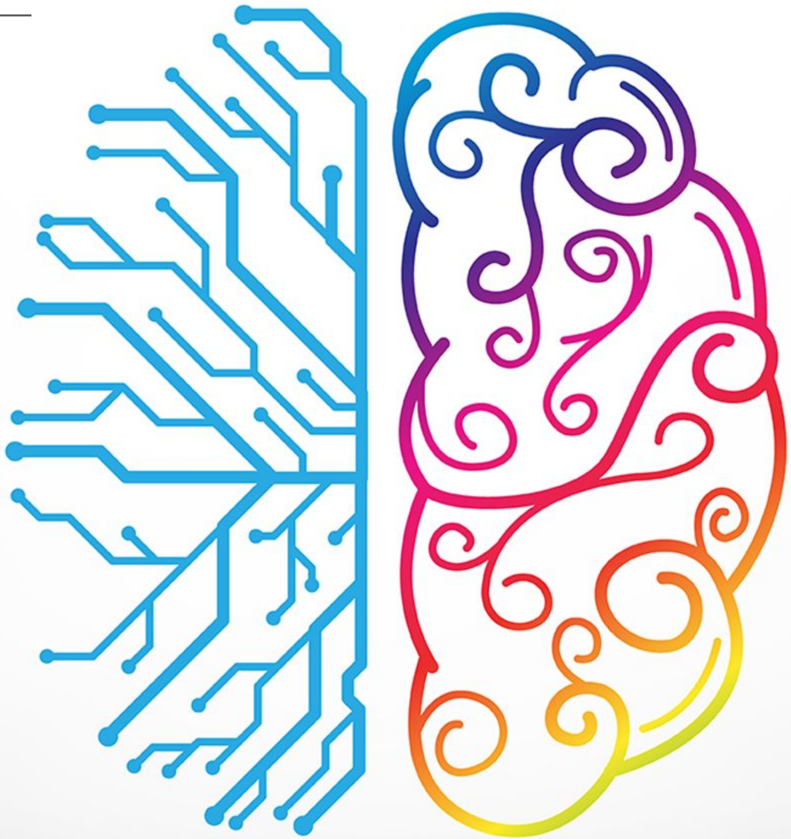
Possibilities include:

Adding more neurons to the hidden layer	Adding more neurons to the model is a possible solution; however, we can overfit the model.
Adding a second hidden layer	Adding a second layer is also a suitable solution; this is part of deep learning, we will cover this in the next class.
Testing with different activation functions	Testing with different activation functions is one of the most used initial solutions, especially when dealing with non-linear data.
Using more epochs	Using more epochs for training is another strategy to improve the model's accuracy.

Modeling Neural Networks

Modeling neural networks is part science and part art.

The best model will be the result of several tests that play around with the number of neurons and testing different activation functions.



Activity: Preventing credit card defaults with neural networks

In this activity, you'll train a neural network model to predict whether a credit card holder will default in the next month.

Suggested Time:
20 Minutes

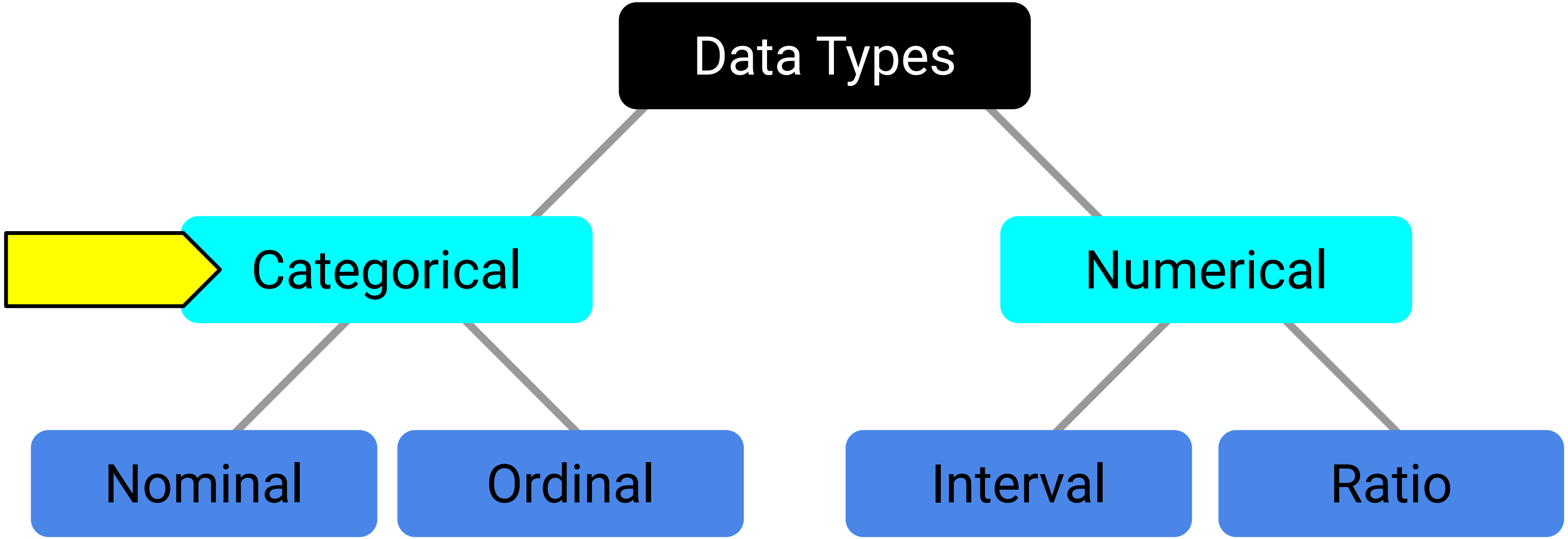
Time’s Up! Let’s Review.


Break

Preparing Data for Neural Networks

Preparing Data for Neural Networks

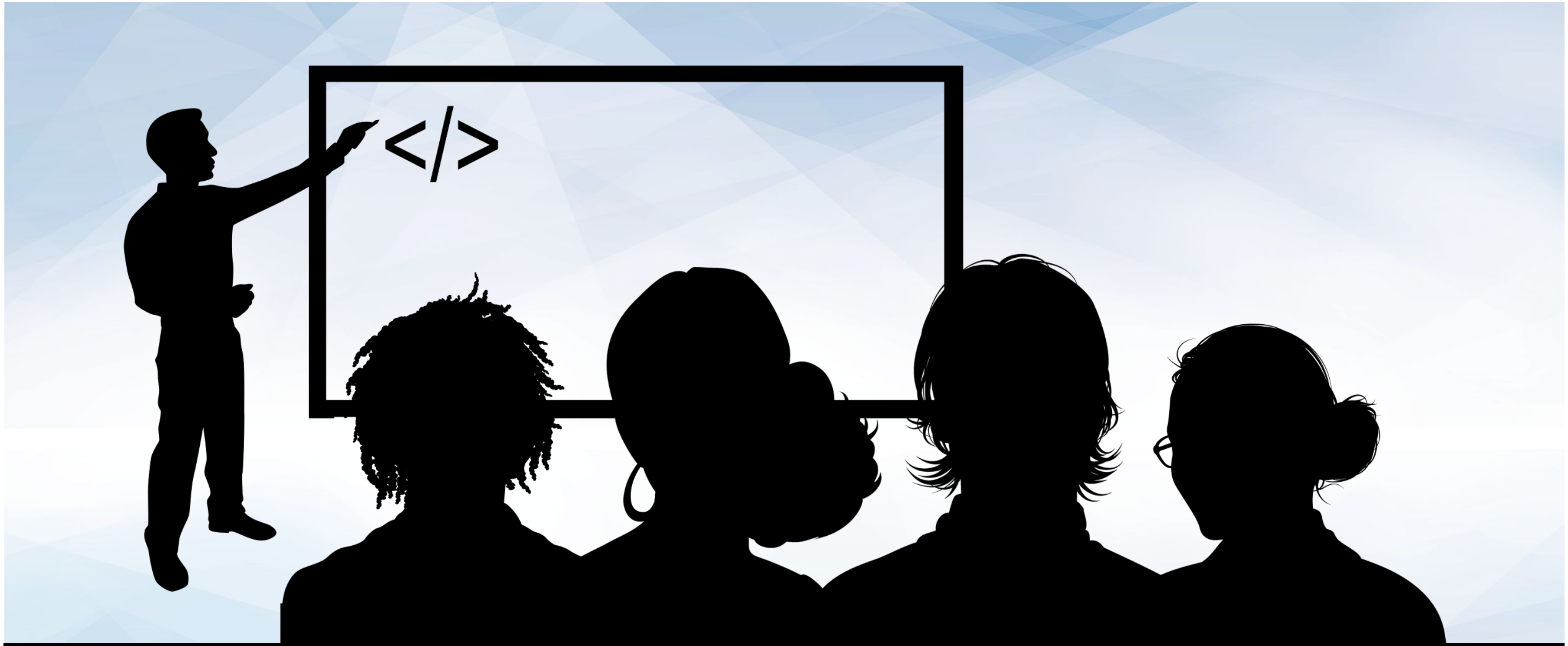
Neural networks cannot deal with categorical variables in their raw forms.



One-hot Encoding

One-hot encoding involves taking a categorical variable, such as color and creating three new variables of the colors. With each instance of the data now showing a 1 if it corresponds to that color and 0 if it does not.

	A	B	C	D	E	F	G	H	I
1	Original data:			One-hot encoding Format					
2	id	Color		id	White	Red	Black	Purple	Gold
3	1	White		1	1	0	0	0	0
4	2	Red		2	0	1	0	0	0
5	3	Black		3	0	0	1	0	0
6	4	Purple		4	0	0	0	1	0
7	5	Gold		5	0	0	0	0	1

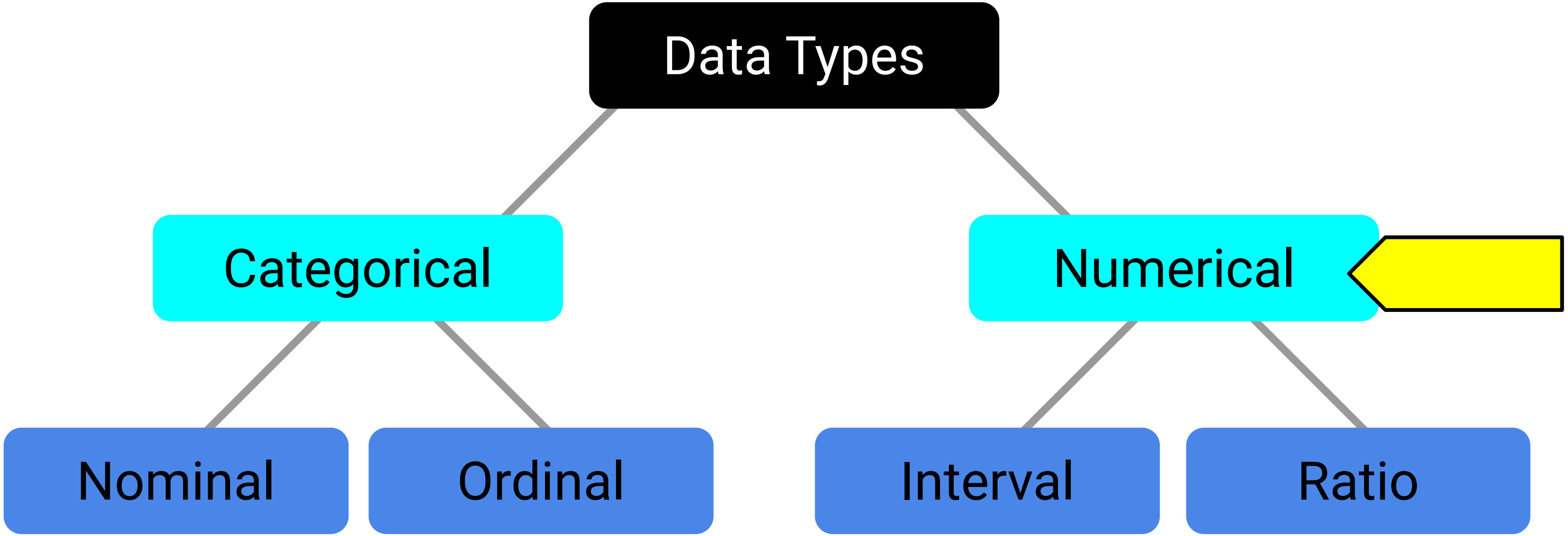


Instructor Demonstration

One-hot encoding and model scaling

Data Standardization

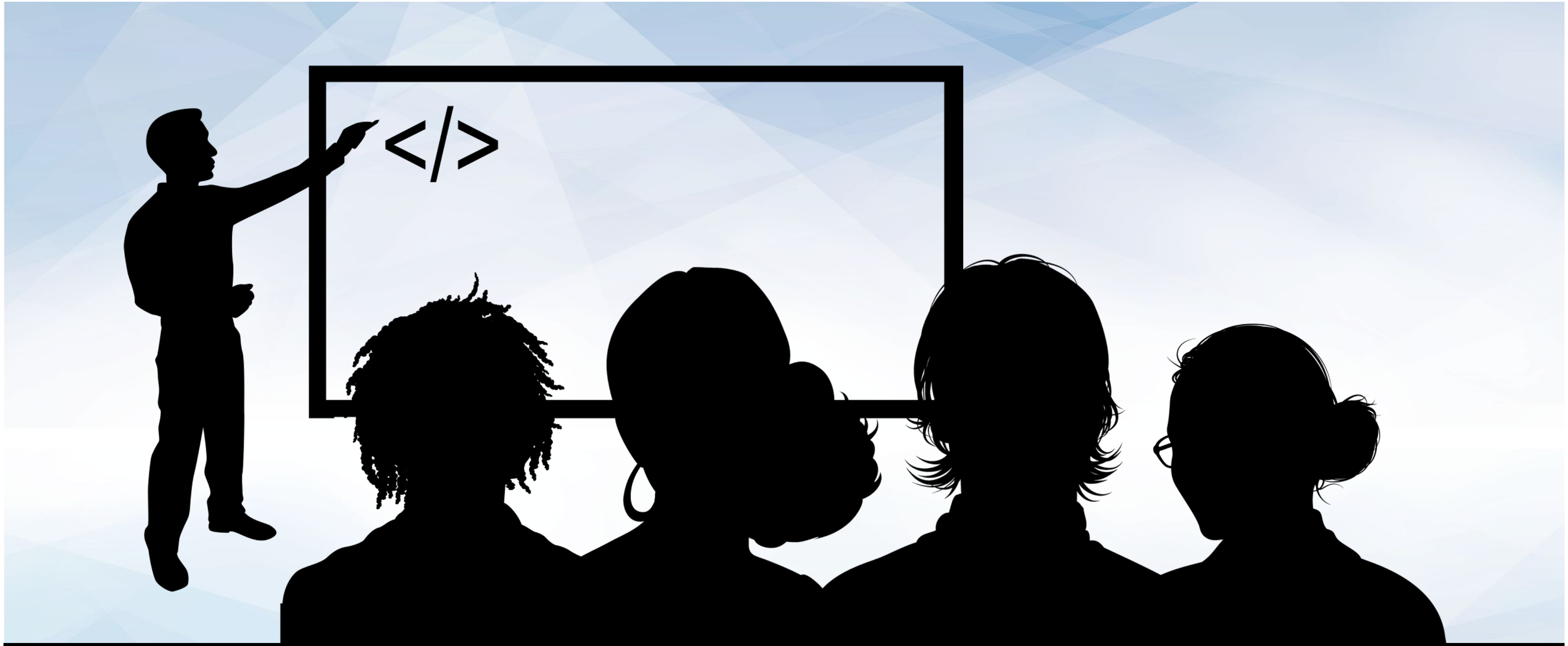
Another critical preprocessing task is to standardize the numerical variables in the dataset.



Data Standardization

Standardization involves demeaning the variables; that is, making it so that each numerical variable has a mean of 0 and a constant variance of 1.

Variables are all approximately the same size, this reduces the likelihood that outliers or variables in different units will negatively affect model performance.



Instructor Demonstration

Data Standardization

Activity:
Smartphone Activity Detector

In this activity, you'll create a neural network to predict a user's activity based on their smartphone accelerometer data.

Suggested Time:
30 Minutes

Time's Up! Let's Review.

Homework Demo

01

Cryptocurrencies are gaining the attention of investors. However, due to their volatile nature, conventional indicators and metrics are not always suitable.

02

The Crypto Fear and Greed Index (FNG) is an instrument used to assess cryptocurrencies.

03

FNG attempts to use a variety of data sources to produce a daily value for cryptocurrency, based on sentiment from social media and news articles to help guide trading strategies.

Questions?

The End