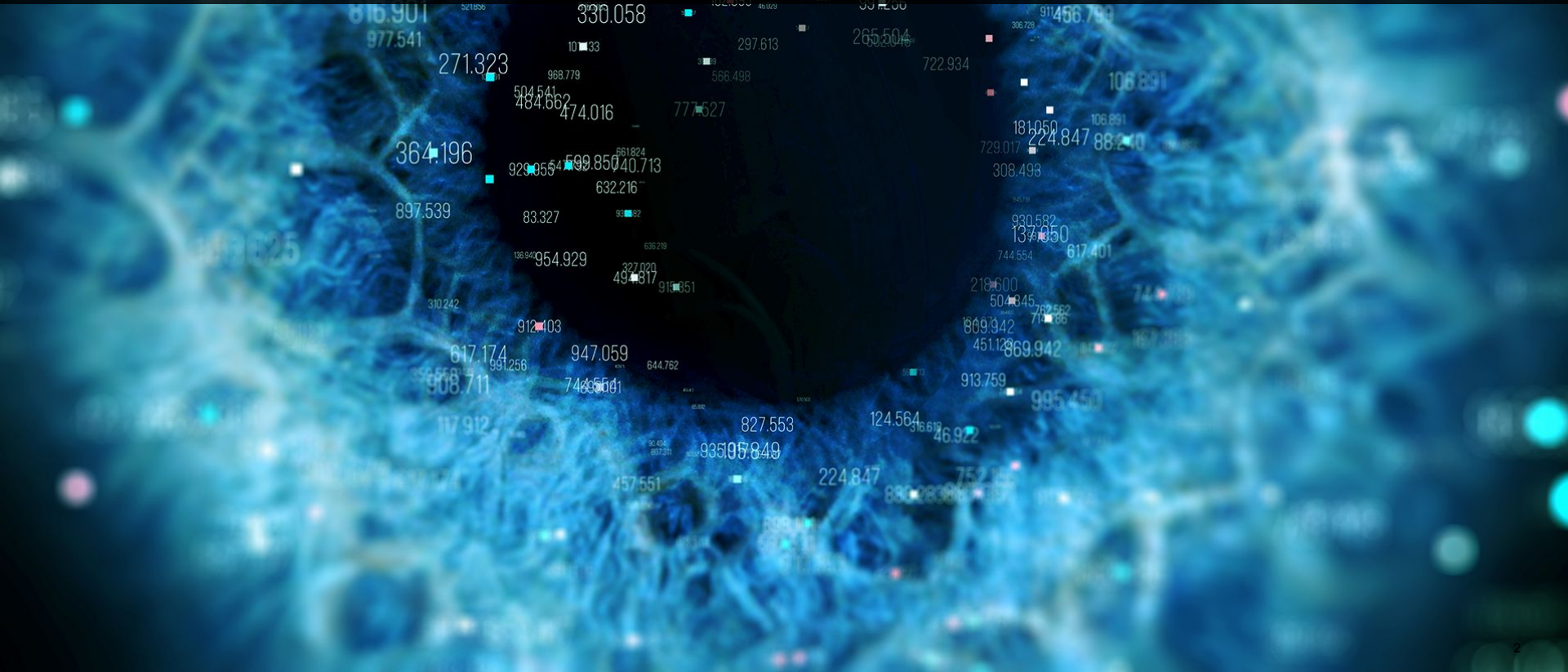
Introduction to Neural Networks

Data Boot Camp

Lesson 21.3

Blindness Detecting Neural Network

[Google’s DeepMind to Scan a Million Eyes to Fight Blindness with NHS]



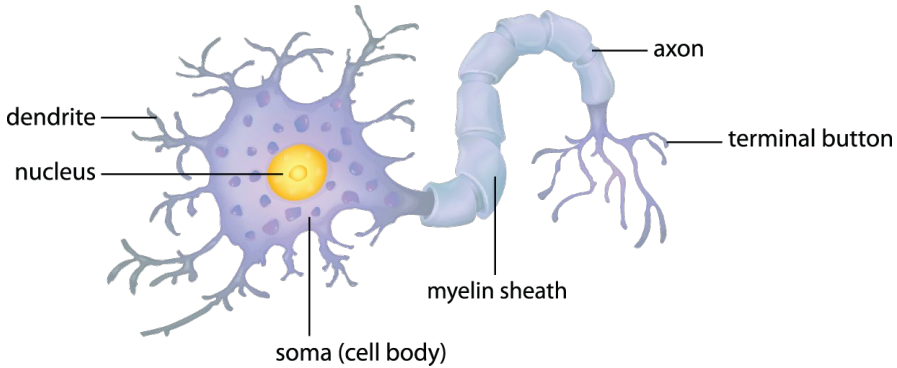
A Neural Network that merges the style of two images

deepdreamgenerator.com

Artificial Neural Networks

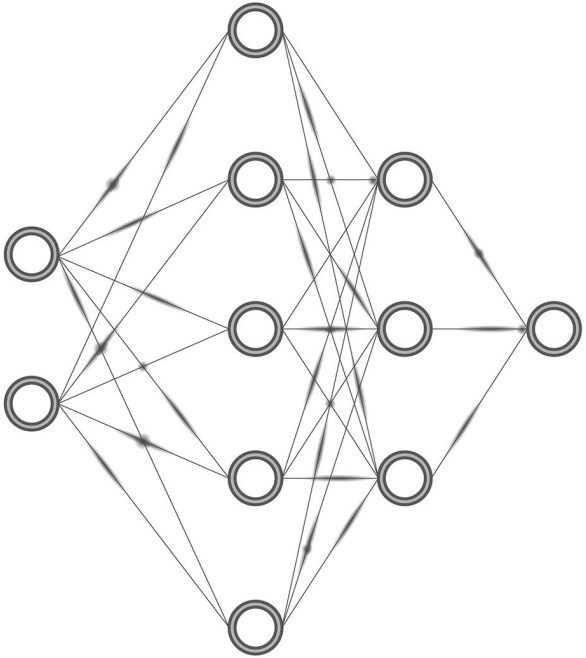
If this is what comes to mind when you think *neural*, you're on the right track:

- The development of neural nets were inspired by the arrangement of neurons in the human brain.
- The neuron's **axon** is what it uses to send signals to other neurons.
- **Synapses** are sensors that receive signals from other neurons.



Artificial Neural Networks

Neutral Networks

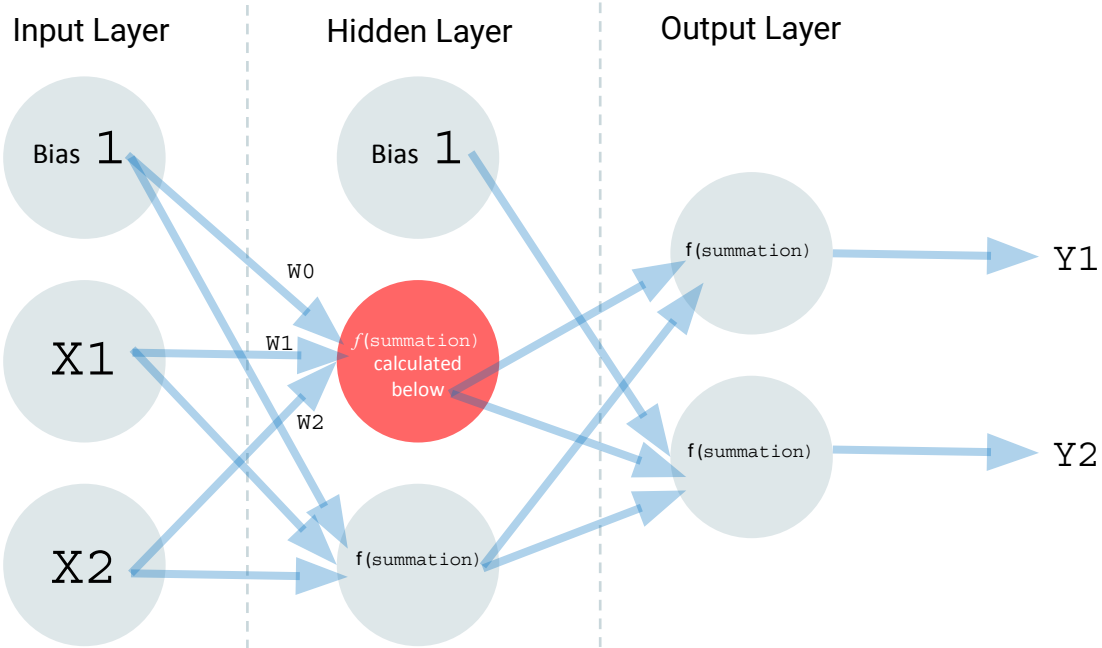


VS

The Brain

The Math:

Neural networks are just layers of neurons connected together.

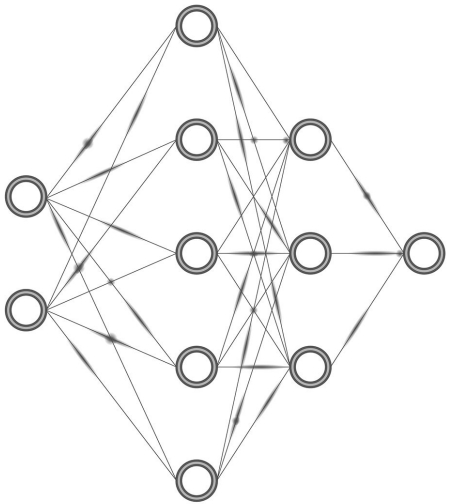


Output of highlighted neuron = f (summation) = f (w0.1 + w1.X1 + w2.X2)

Artificial Neural Networks

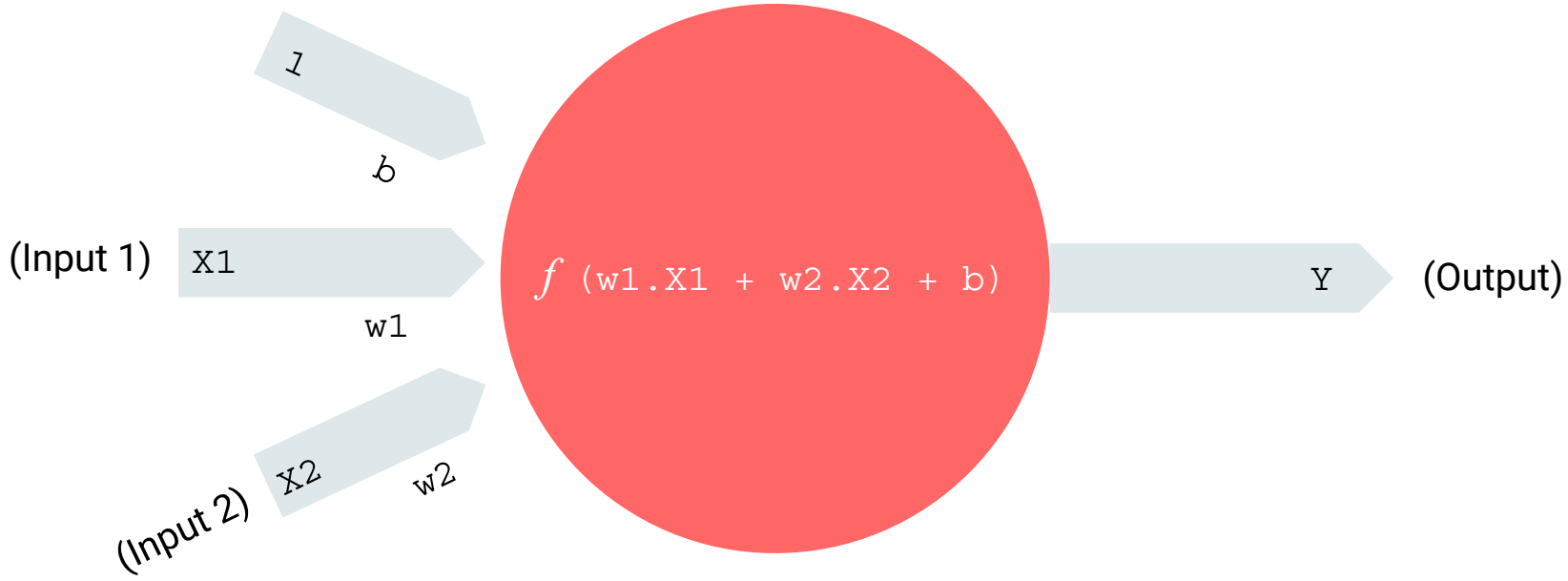
As it turns out, the neural networks we program find patterns in inputs in much the same way the brain does.

If you understand how the brain processes visual information, you'll understand how neural nets process *any* information.



The Math:

Neurons are kind of similar to logistic regression.

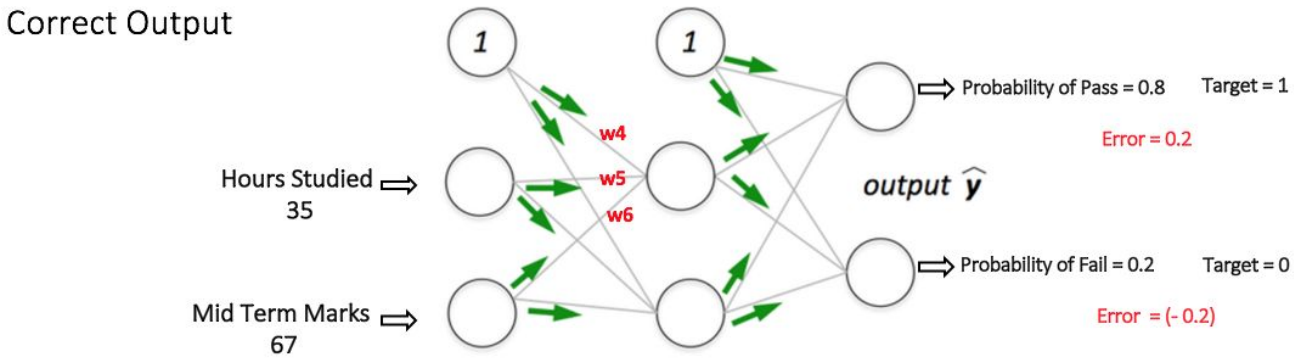


Output of neuron = $Y = f(w1.X1 + w2.X2 + b)$

The Math:

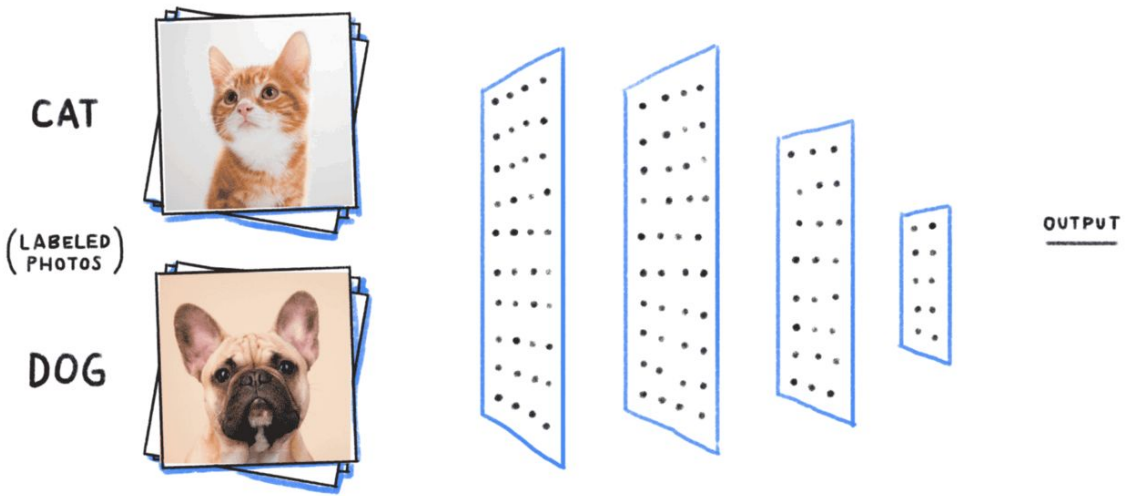
The most important mathematical detail to understand is that neural networks typically output **probabilities**.

In other words, if we were to tell it how many hours we studied for a test, and our grade on the midterm, it will spit out a *probability* that we'll pass (or fail!) the final.



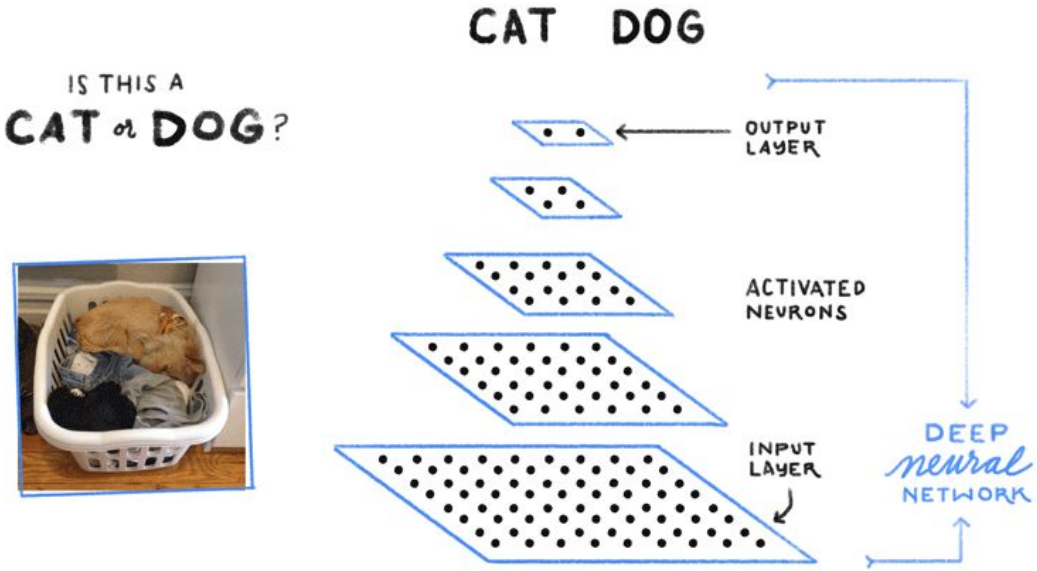
How Does It Work?

Neural networks are trained by feeding it (**forward propagation**) a bunch of inputs (like pictures of cats and dogs) and comparing the outputs to a known label (the output was a cat). The error in the output probability is then fed back (**back propagation**) to update the weights in each neuron. This is repeated until the neural network learns how to provide an accurate probability for a given input.



How Does It Work?

A trained neural network can take a new input (picture of a dog) and feed it through the network to predict the output label (cat or dog).



https://www.google.co.in/about/main/machine-learning-qa/

In a Nutshell:

If at first you don’t succeed, try, try again.

Questions?

Why Keras? What Is Keras?

Keras

A layer of abstraction

Simplifies deep learning modeling

Allows rapid prototyping

Works on top of TensorFlow (among other libraries)

TensorFlow

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

Keras + TensorFlow

Keras allows interaction with TensorFlow through a simplified interface similar to Scikit-Learn.

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

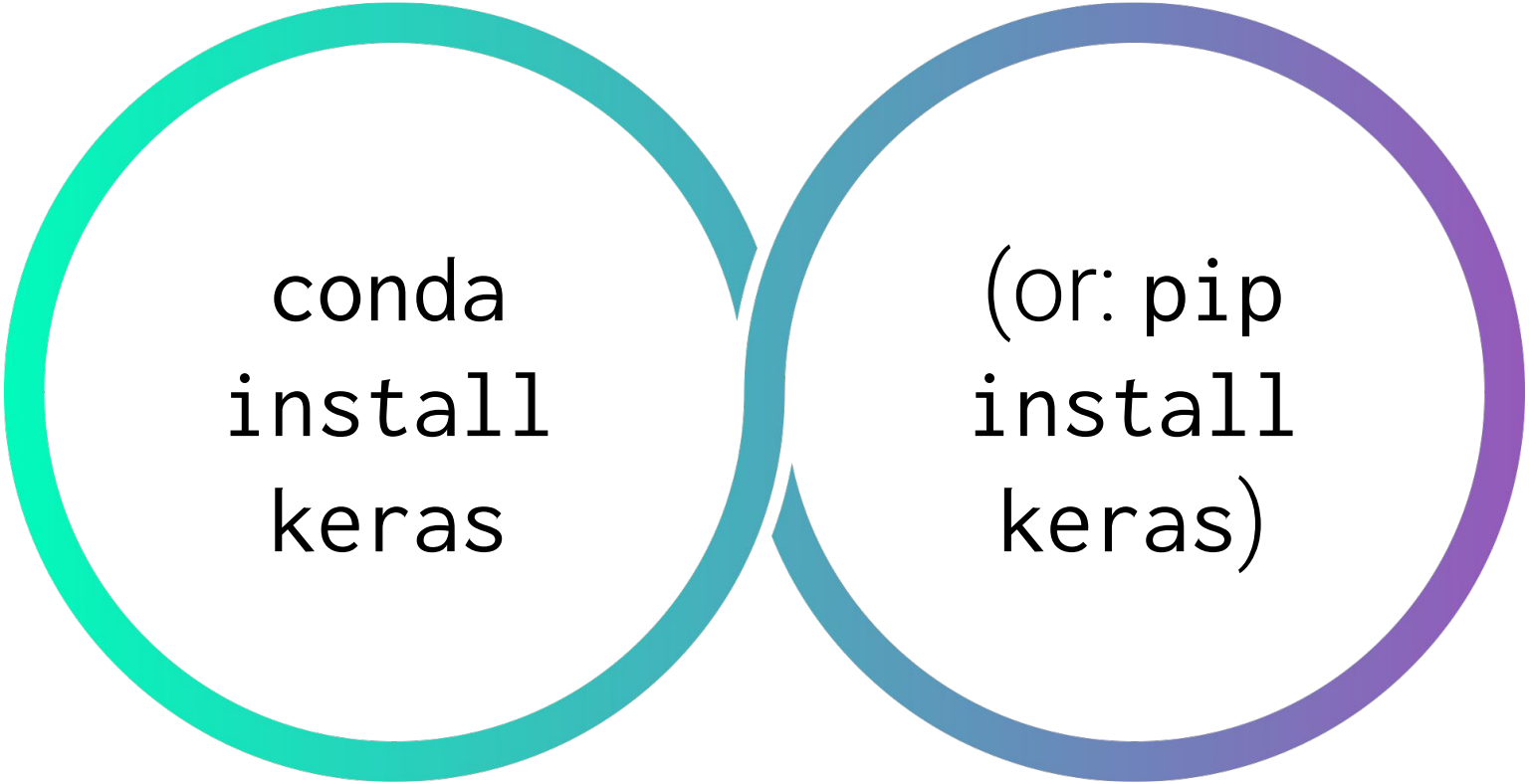
```
from keras.models import Sequential

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)
```

Installation

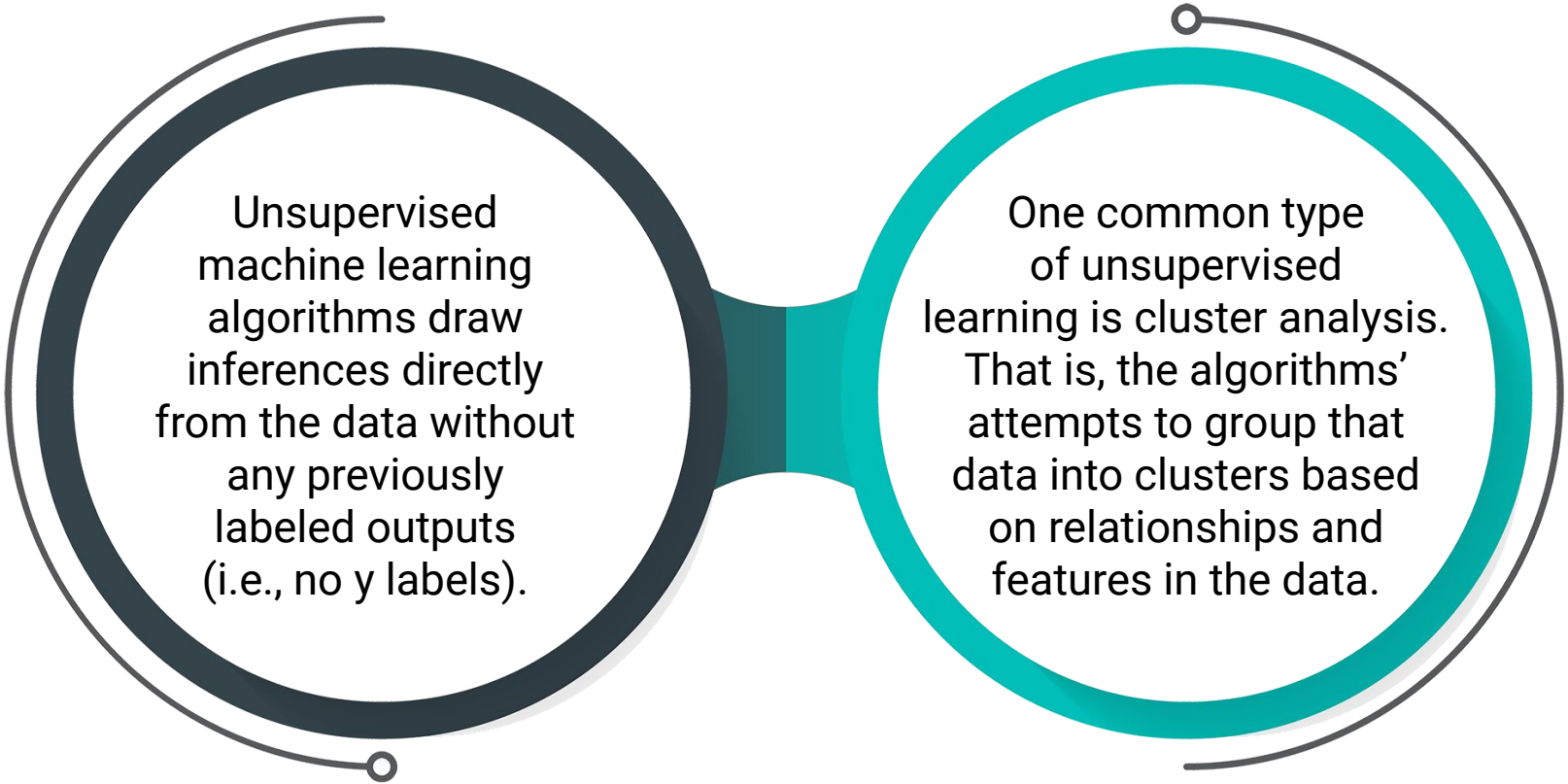
Installation



Questions?

k-means

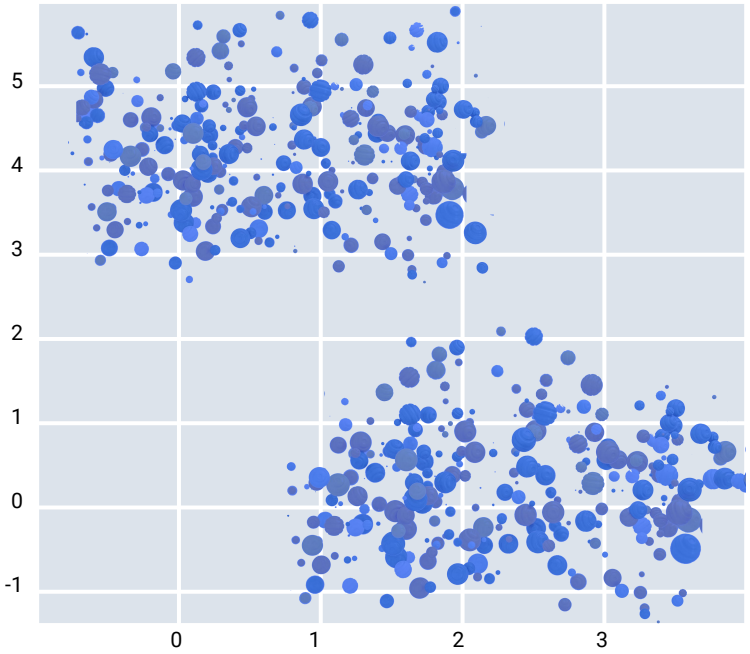
Unsupervised Learning



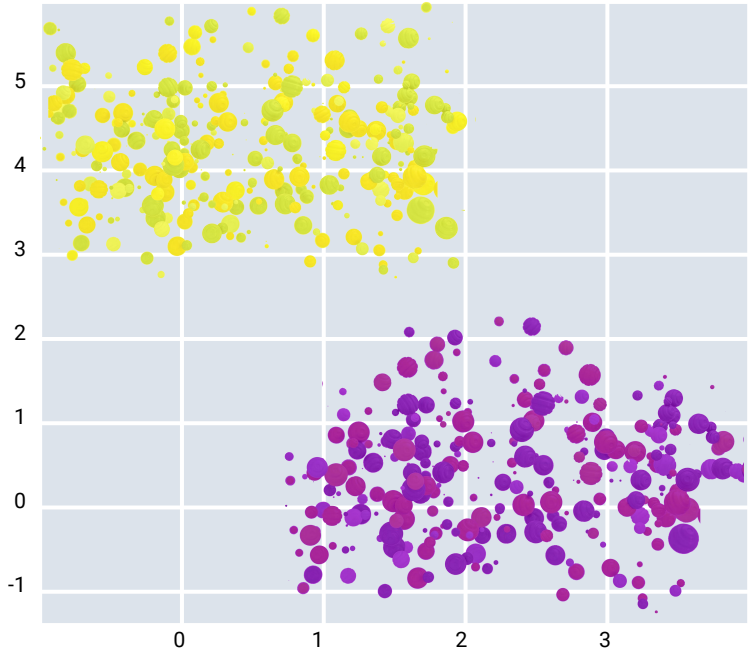
Clustering

While clustering may be intuitive to humans in this case, clustering algorithms have to decide which data points belong together.

Unclustered Data

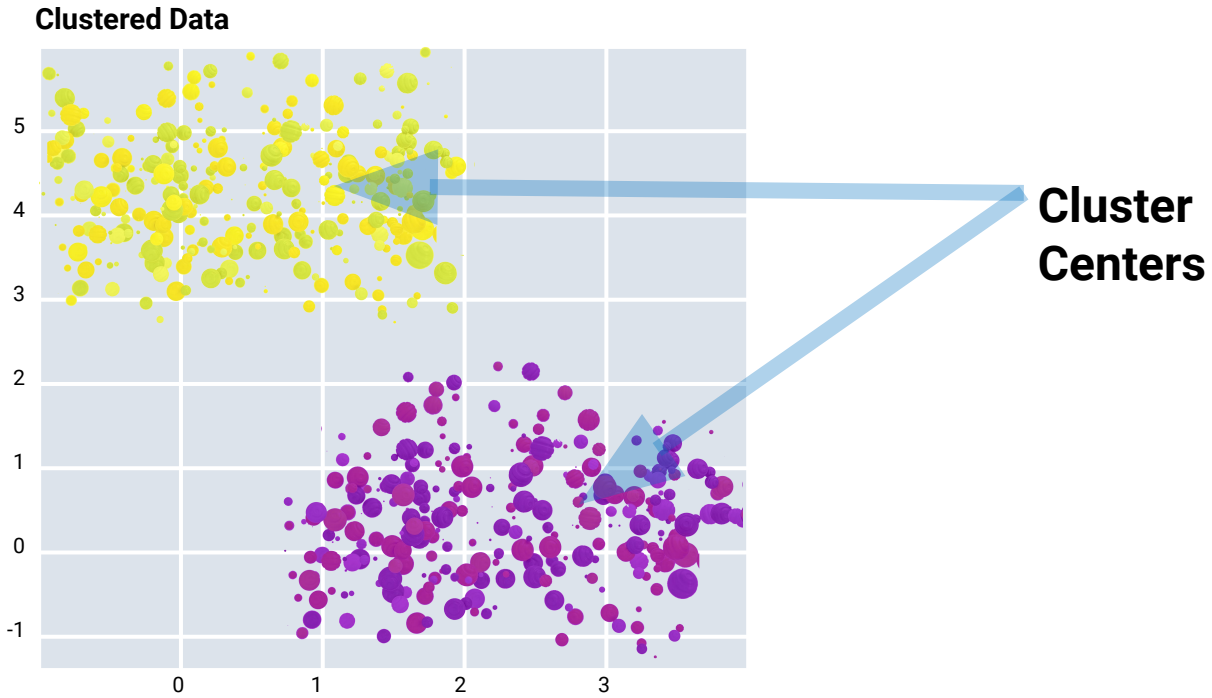


Clustered Data



k-means Clustering Will Group the Data Into k Groups

The cluster center is the mean of all the points belonging to that cluster. Therefore, each point is closer to its own cluster's center than it is to other cluster centers.



k-means

k-means clustering will group the data into k number of groups.
A small k will create larger clusters.

K = 2

k-means

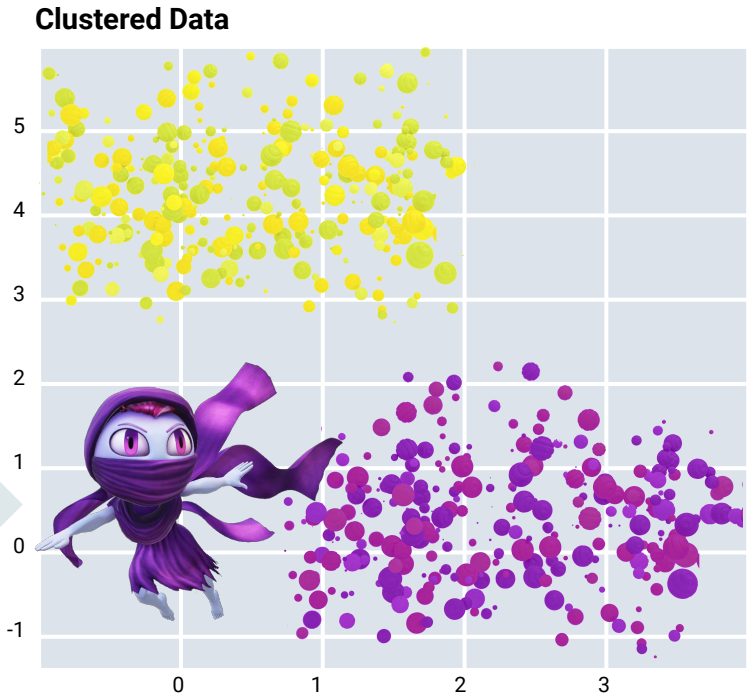
A large k will create smaller clusters.

K = 6

k-means

Predictions using a trained k-means model

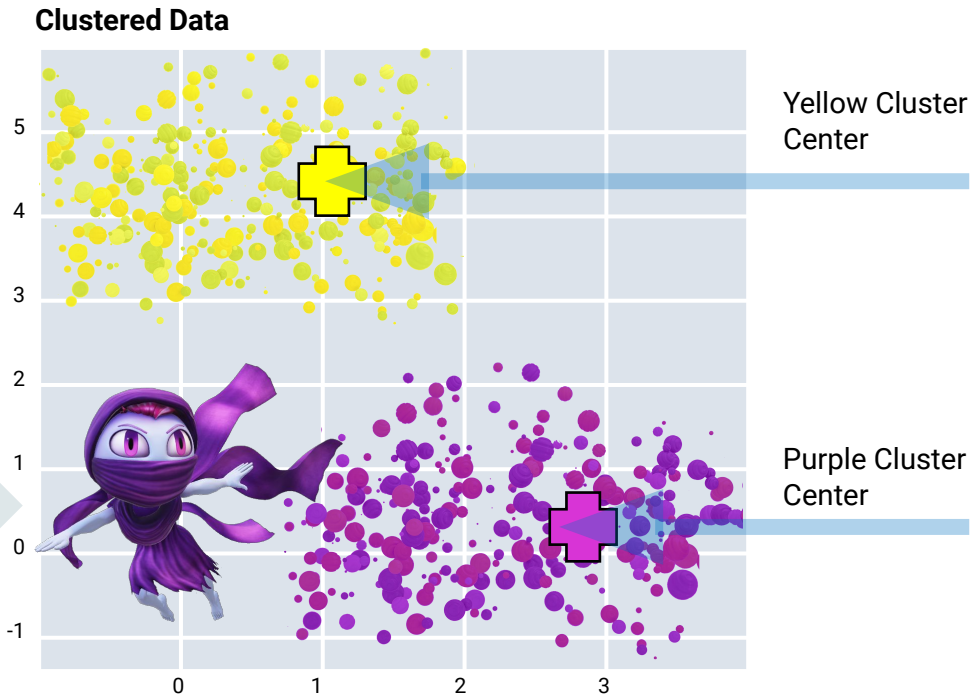
Hi guys! Where do you want me?



Predictions Using a Trained k-means Model

What group would this new data point belong to?

Hi guys! Where do you want me?



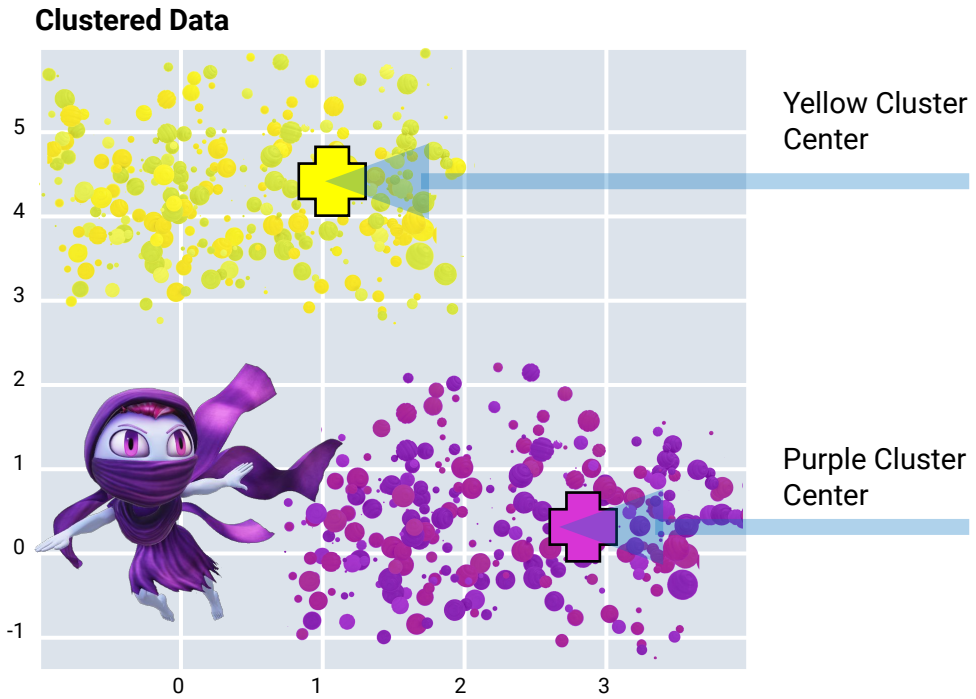
Predictions Using a Trained k-means Model

We should use Python to figure out which cluster center is the closest:

```
np.sqrt(sum((x - y) ** 2))
```

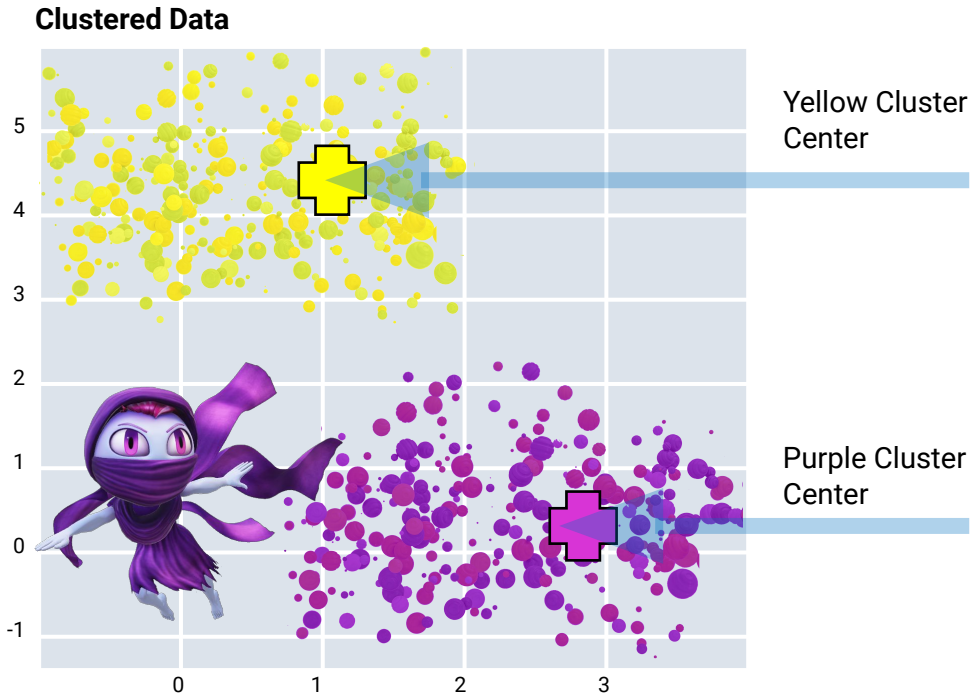
We can also figure this out using:

```
kmeans.predict(new_data)
```



Predictions Using a Trained k-means Model

Purple is closer!



Questions?