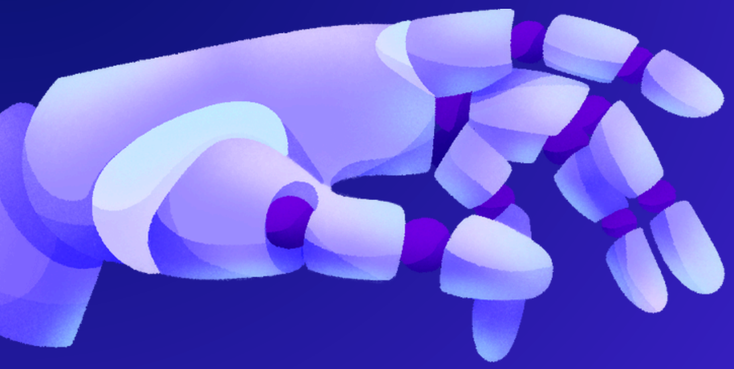
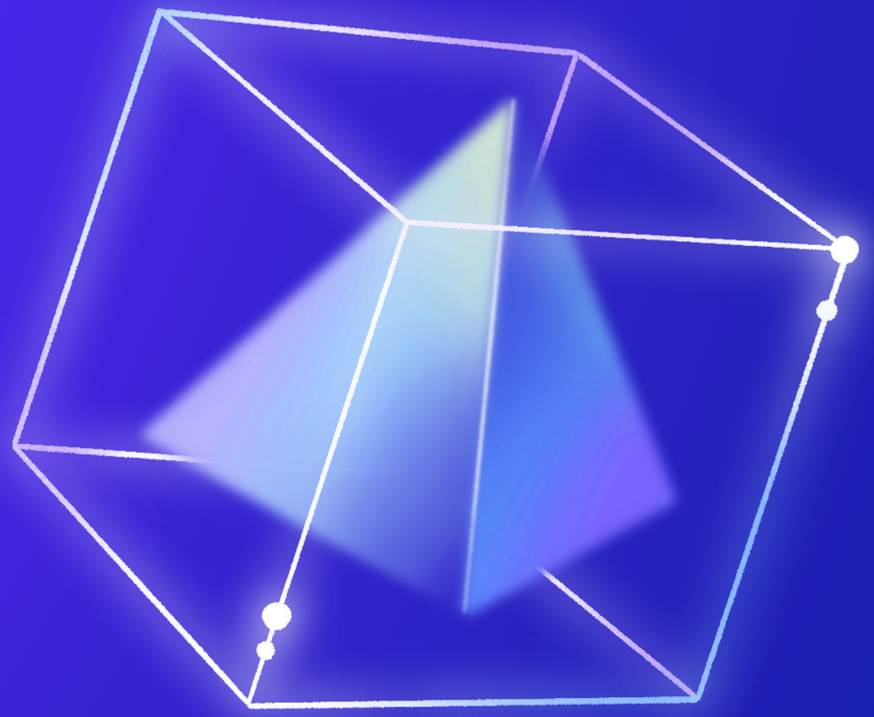

NAISS MACHINE LEARNING BOOTCAMP
SESSION 1
WHITEBOX MODELS



AGENDA

- What exactly is Machine Learning?
- Types of Machine Learning Models
  - Linear Regression
  - Logistic Regression
  - Support Vector Machines
- Applications of these Algorithms



MACHINE LEARNING

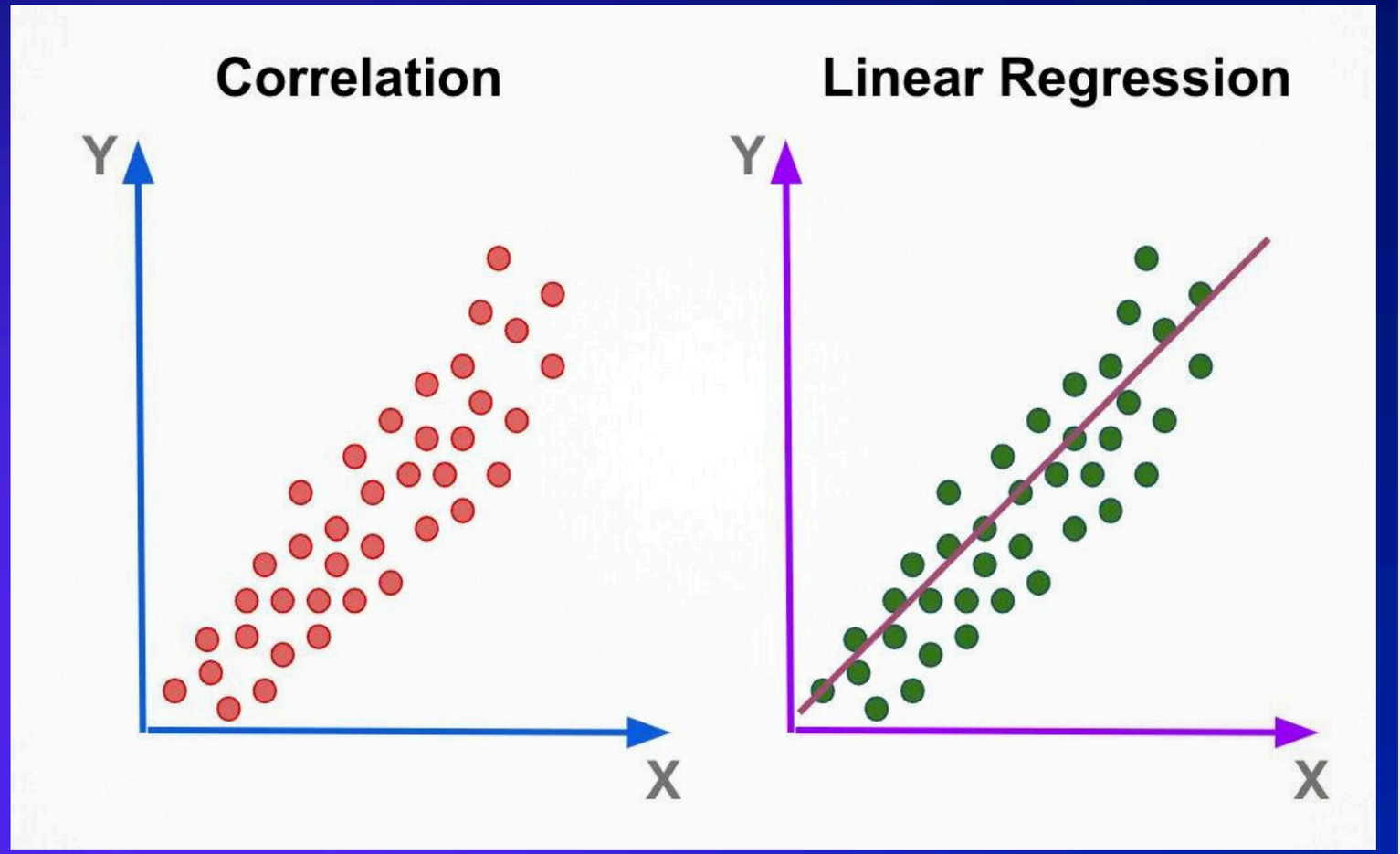
What is it?

It’s a field of study in AI that trains computers, using data and algorithms to draw patterns from data.

Machine Learning Process

LINEAR REGRESSION

Linear Regression is used to predict a variable based on the value of another variable.



LINEAR REGRESSION APPLICATIONS

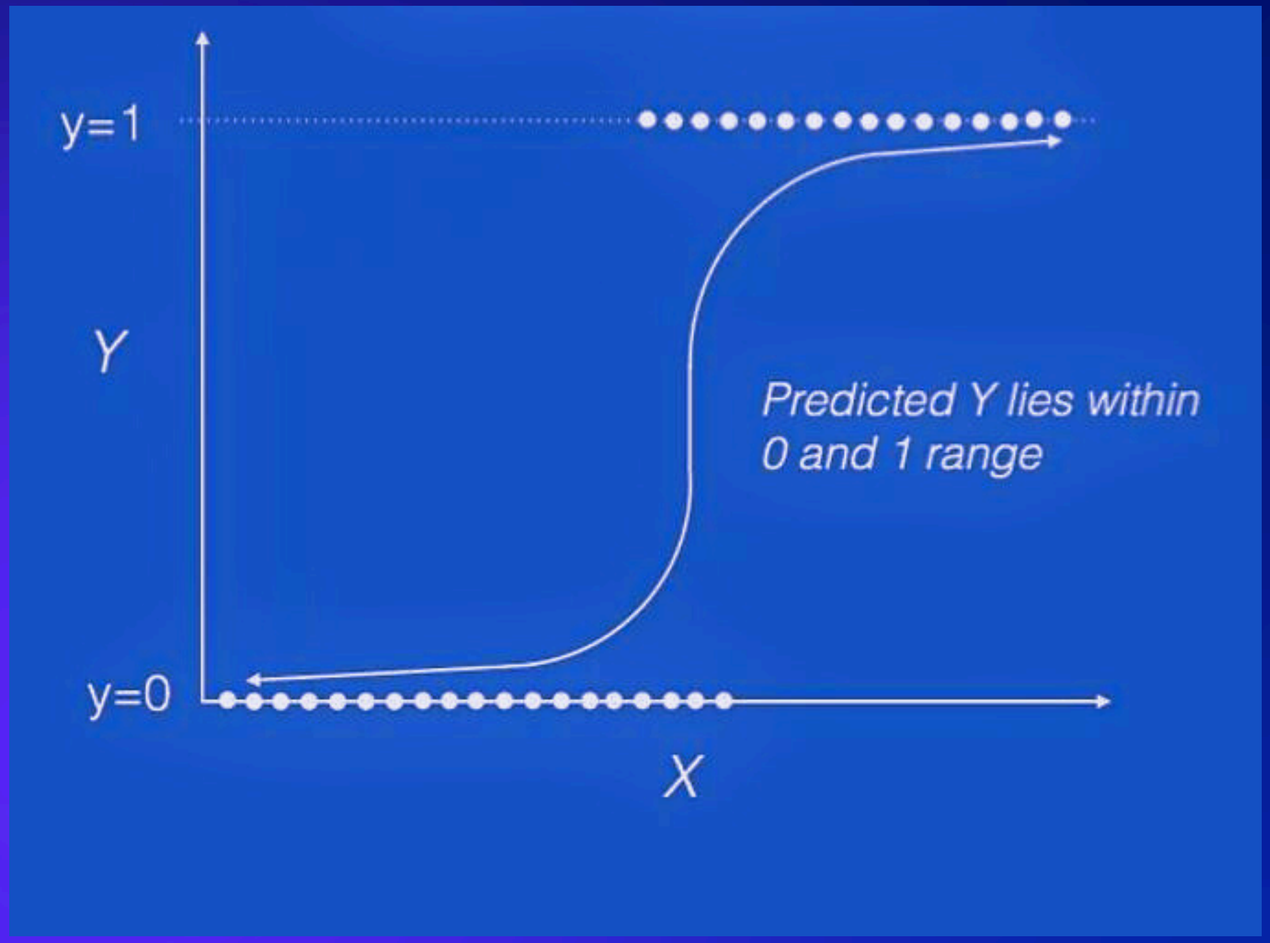
Advertising vs Revenue

Drug dosage vs BP

Fertilizer vs Crop Yield

LOGISTIC REGRESSION

Logistic Regression is used to generate a “yes” or “no” type response based on independent variables.



LOGISTIC REGRESSION APPLICATIONS

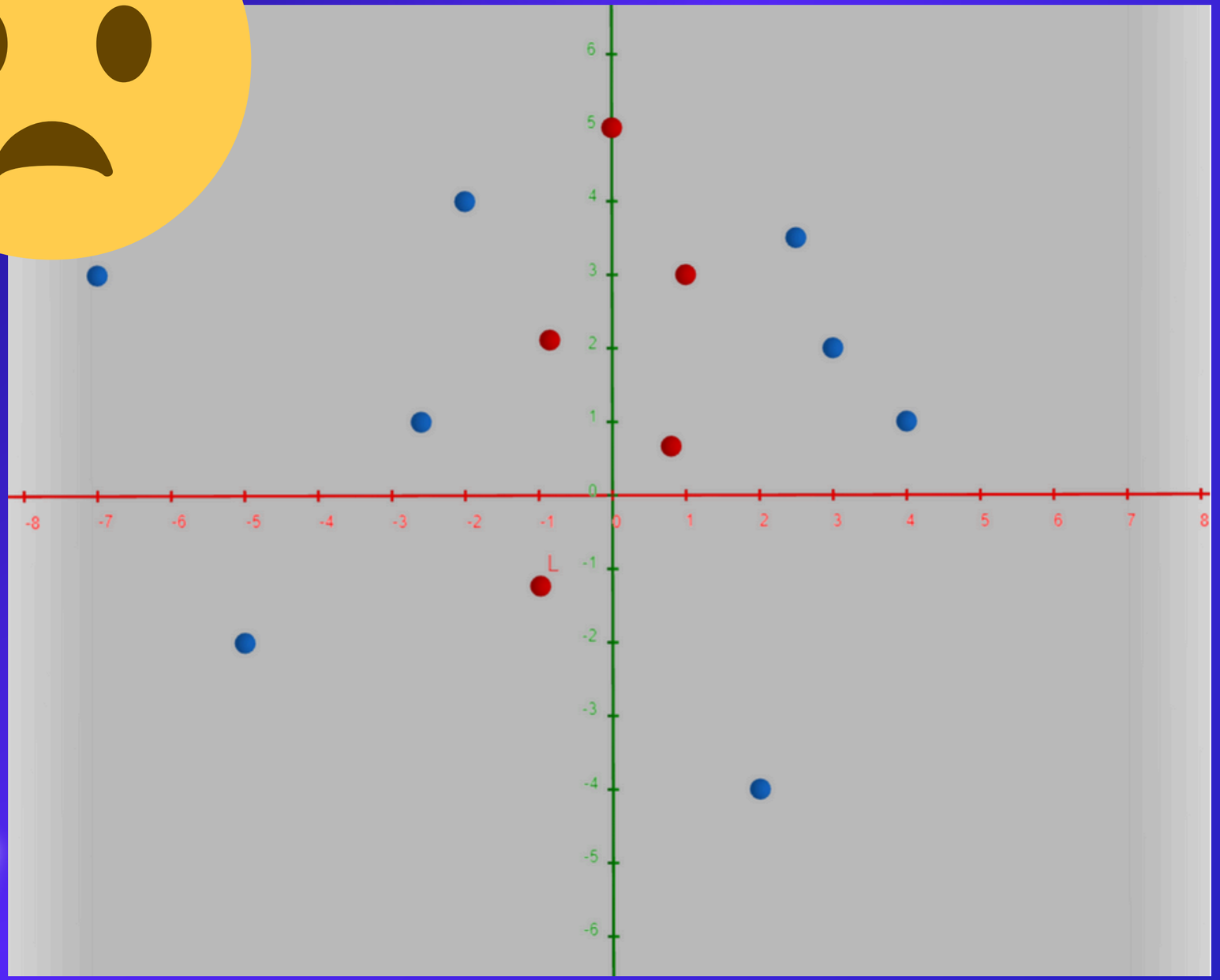
Customer Usage Probability

Spam Likelihood

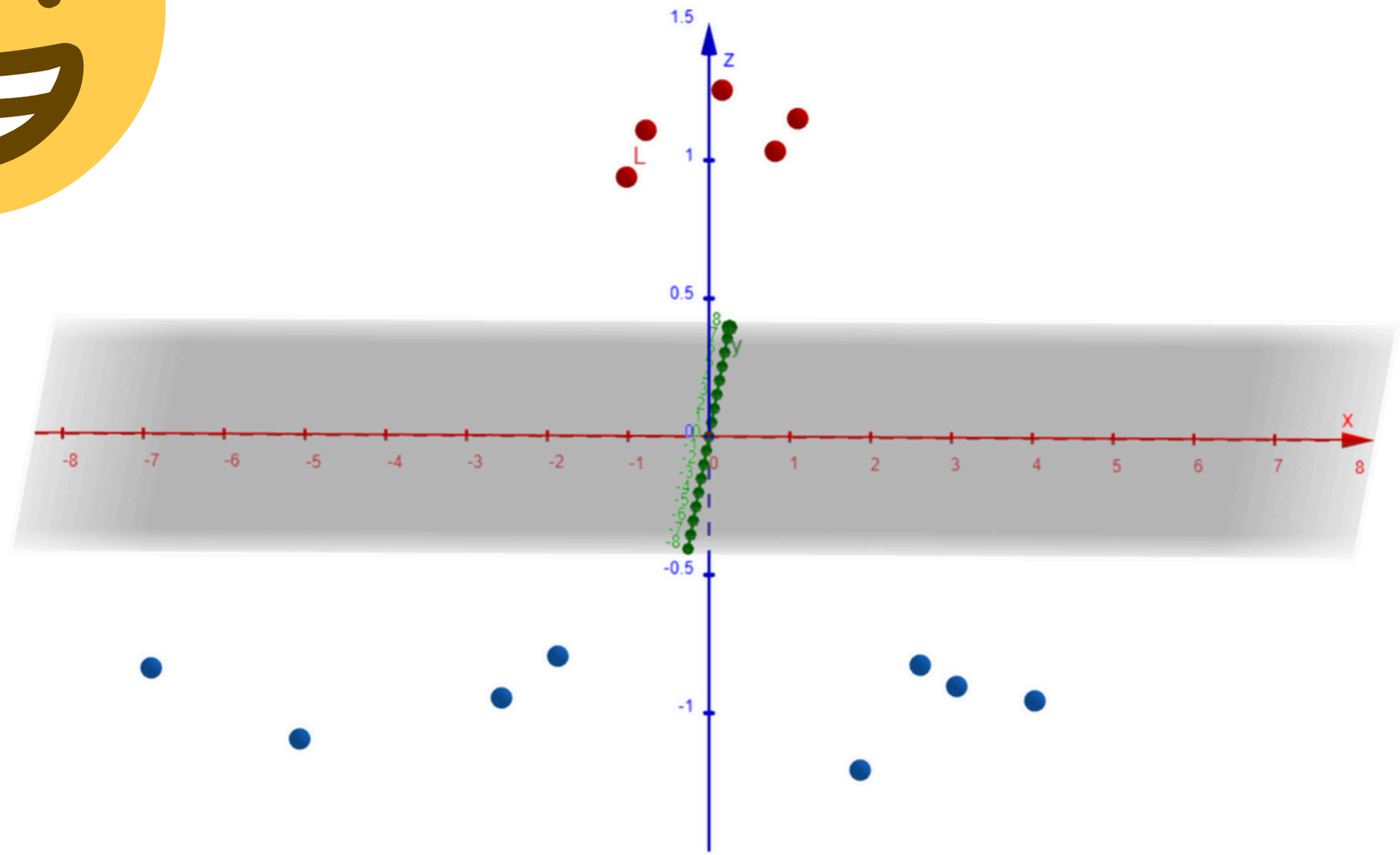
Disease Risk Chance

SVM

SVM



SVM



SVM USE CASES

CLASSIFICATION

FAST

GOOD WITH HIGH DIMENSIONALITY

Fraud Detection

Google Image Classification

Recommender Systems

Text Classification

Sentiment Analysis

Handwriting Recognition

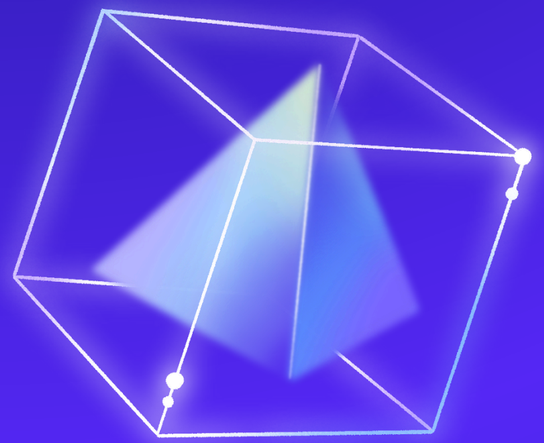
Flower Classification

Fortnite Win Prediction

Gmail Classification

Among Us Imposter Detection

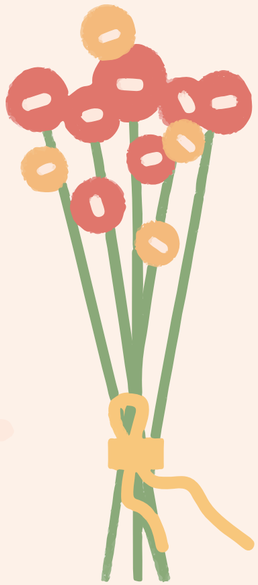
...



THANK YOU!

NAISS MLB

Session #2
Whitebox Models (continued)



AGENDA

Decision Trees

Applications of Decision Trees

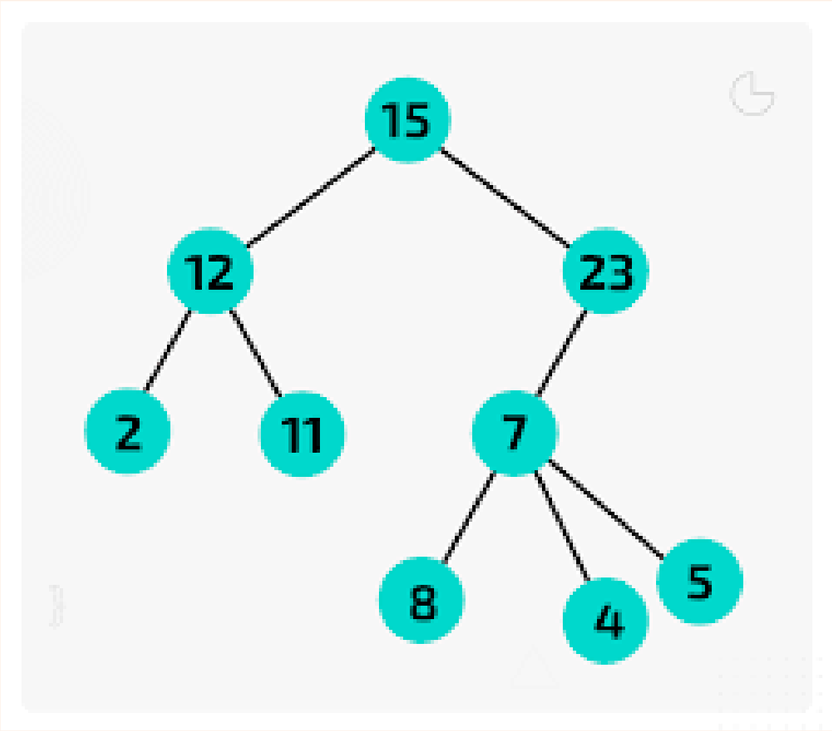
K-means Clustering

Applications of K-means clustering

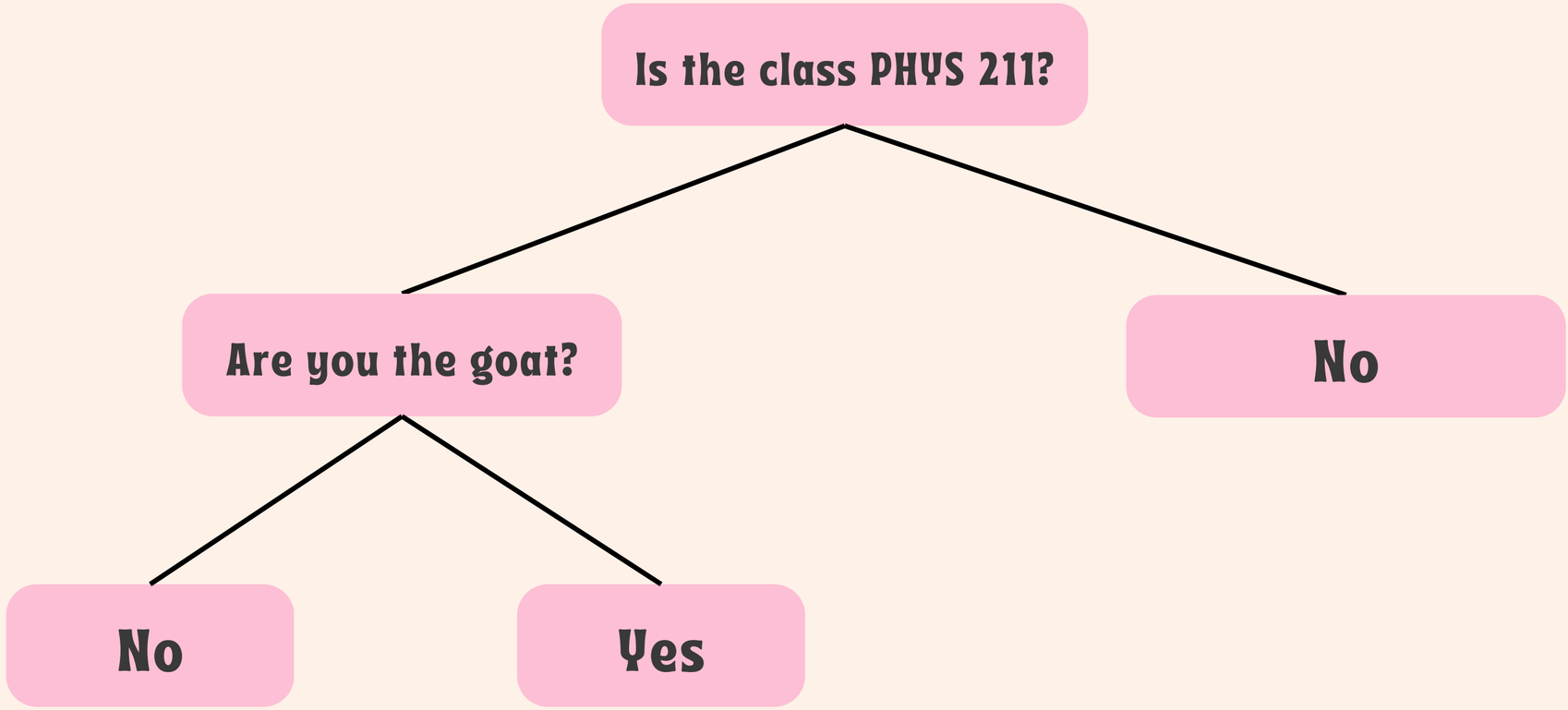
Decision Trees

What are decision trees?

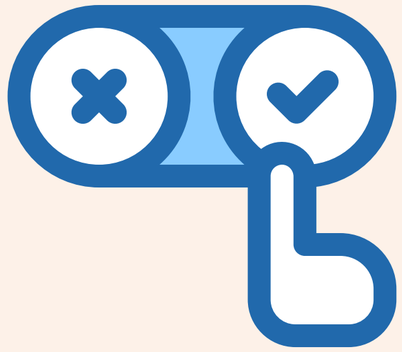
Decision trees is a machine learning algorithm that uses a tree structure



Should you study for your midterm?



Applications of decision trees



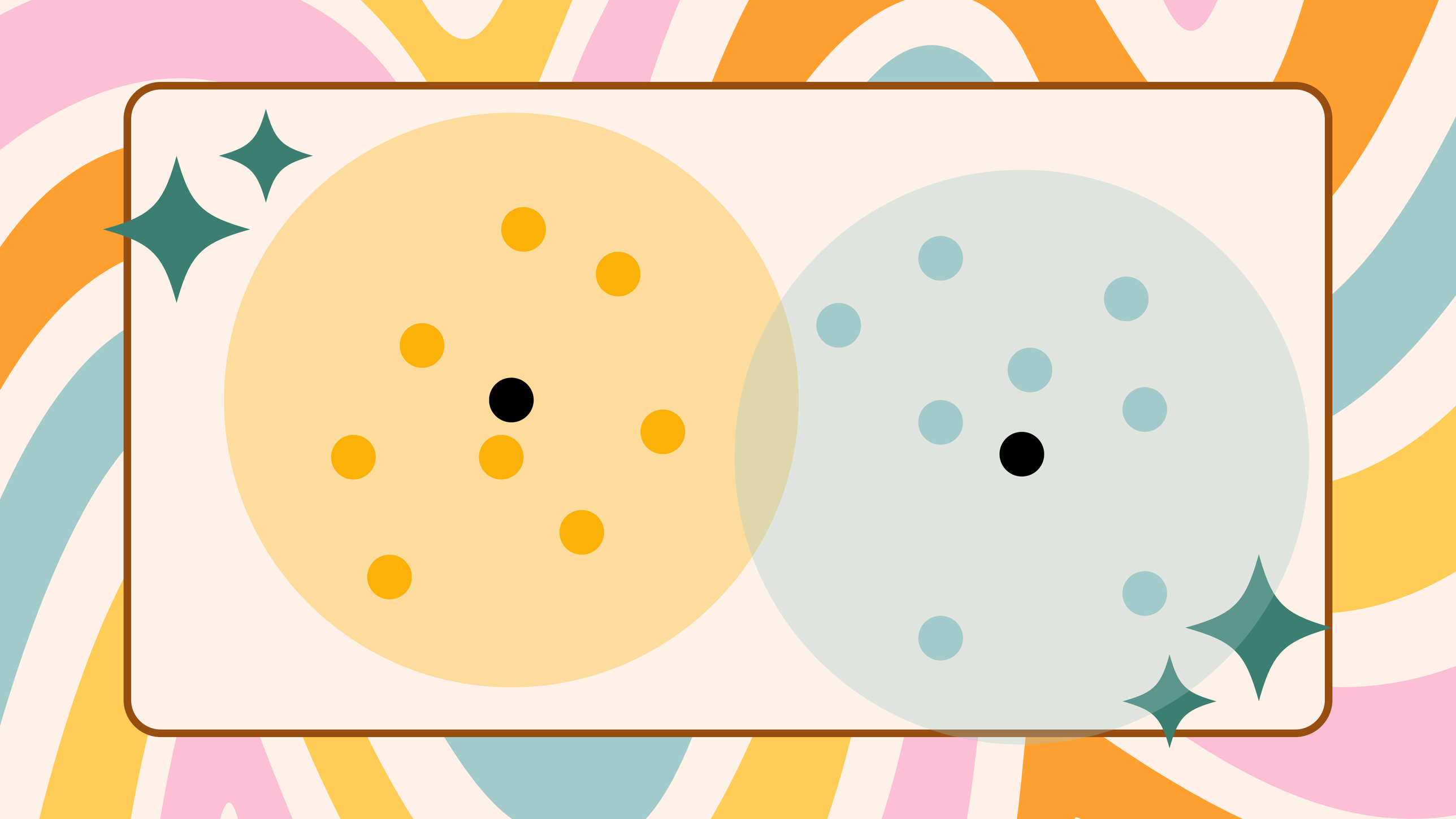
Decision Making

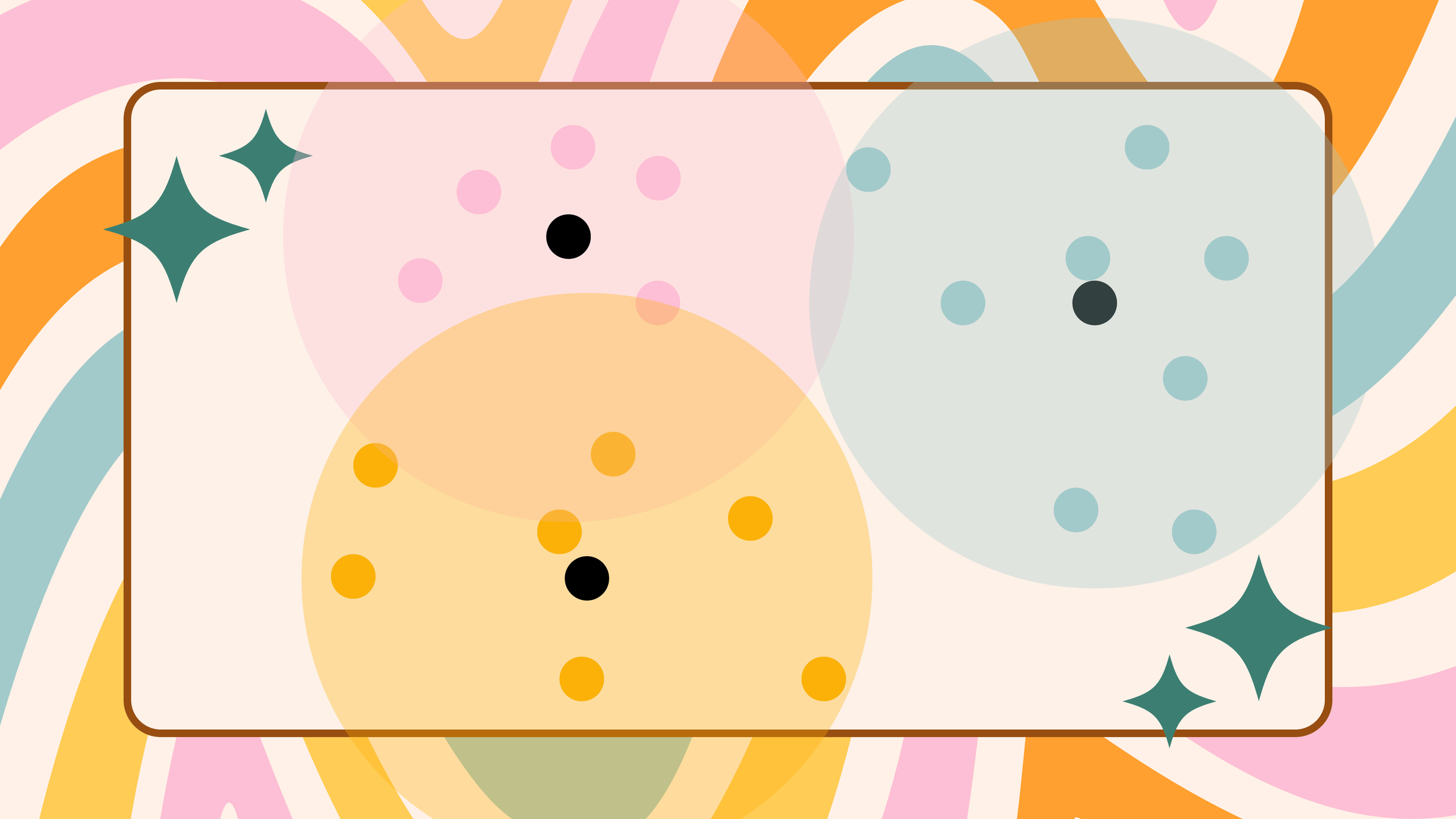
Classifying Diseases

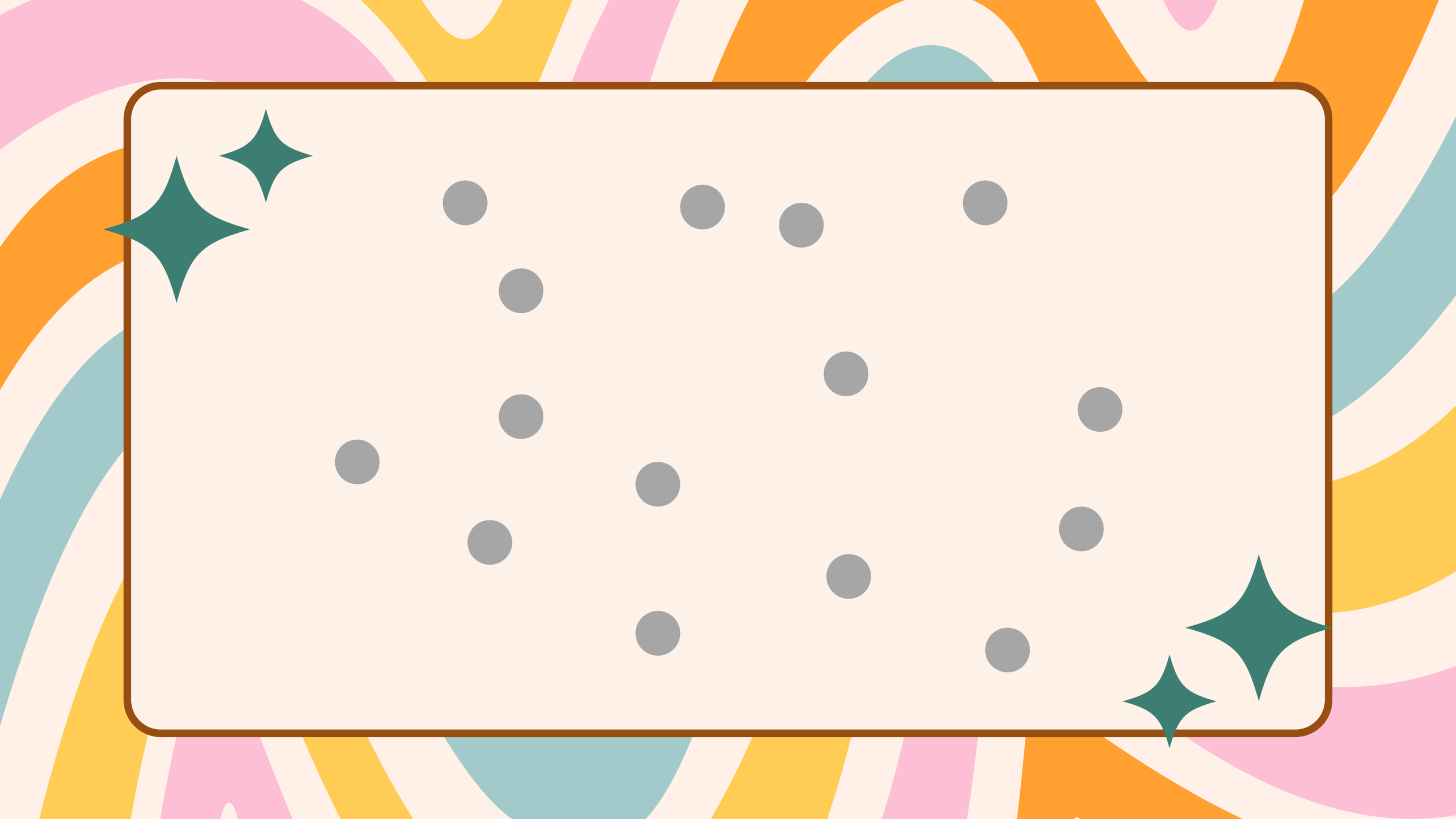
Quality Control

K-means Clustering

Eric Farrall

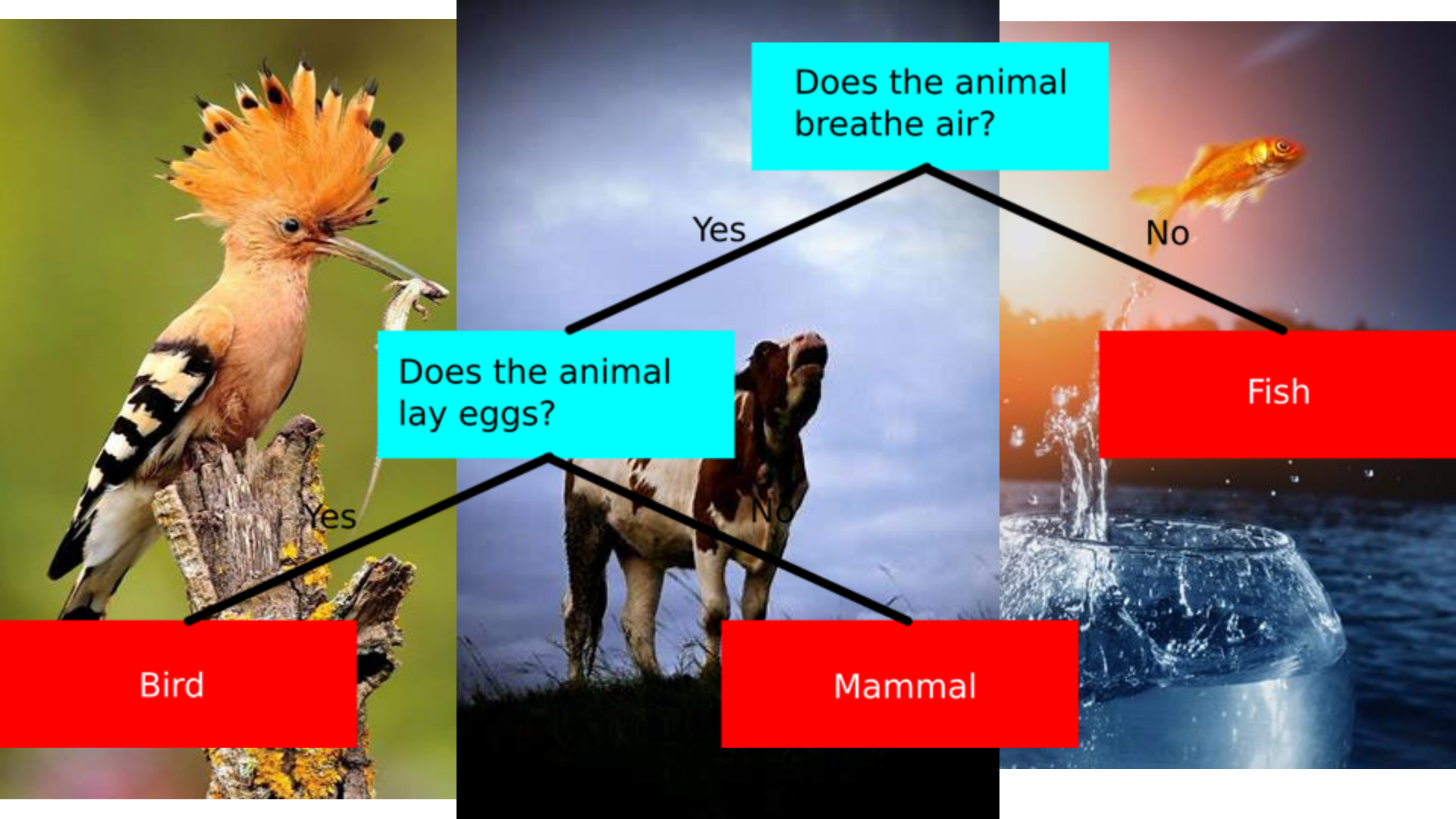






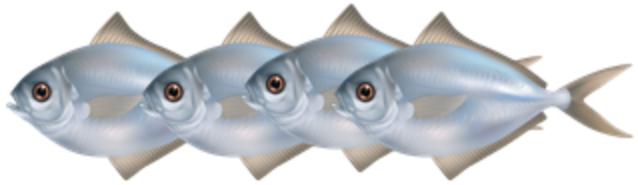
Thank you!


Random Forest
CRAZY FRANKENSTEIN.COM


Does the animal breathe air?
Yes
No
Does the animal lay eggs?
Yes
No
Bird
Mammal
Fish

Messy 🤮

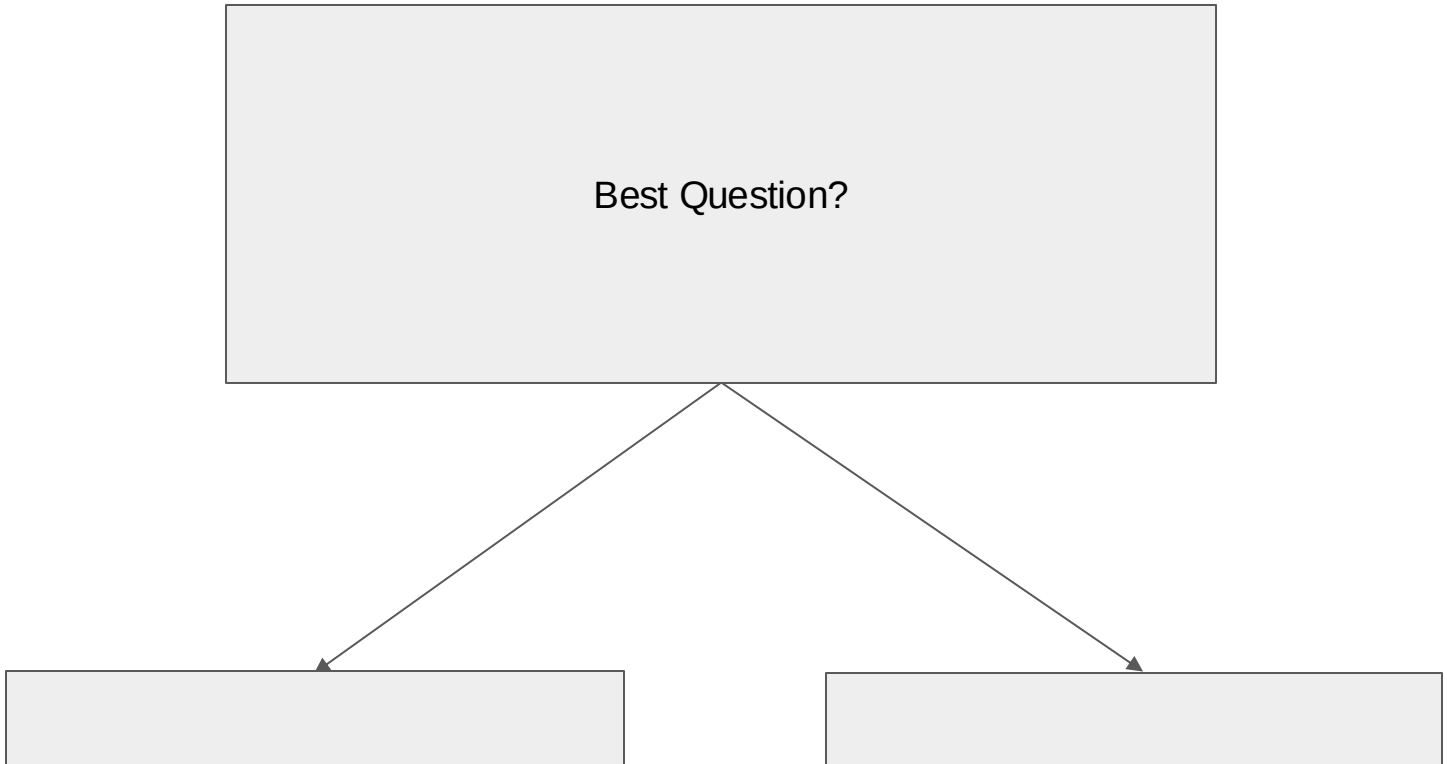
Good 👍

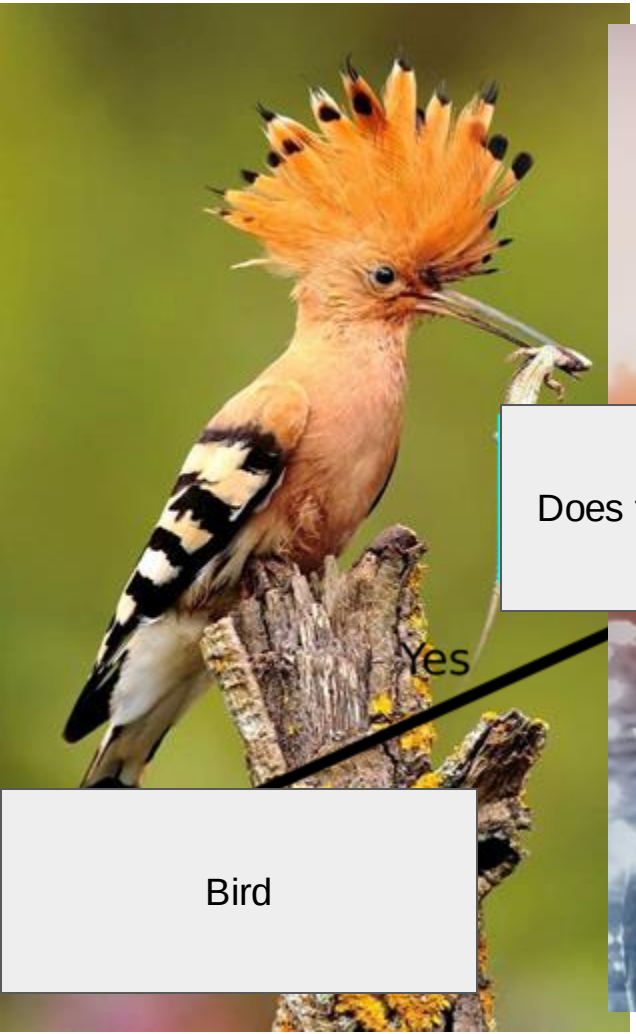


Good 👍

Good 👍

2 features:
Does it Breath; Does it lay Eggs





Does the animal lay eggs?

yes

No

Does the animal breath?

Cow

Yes

no

Bird

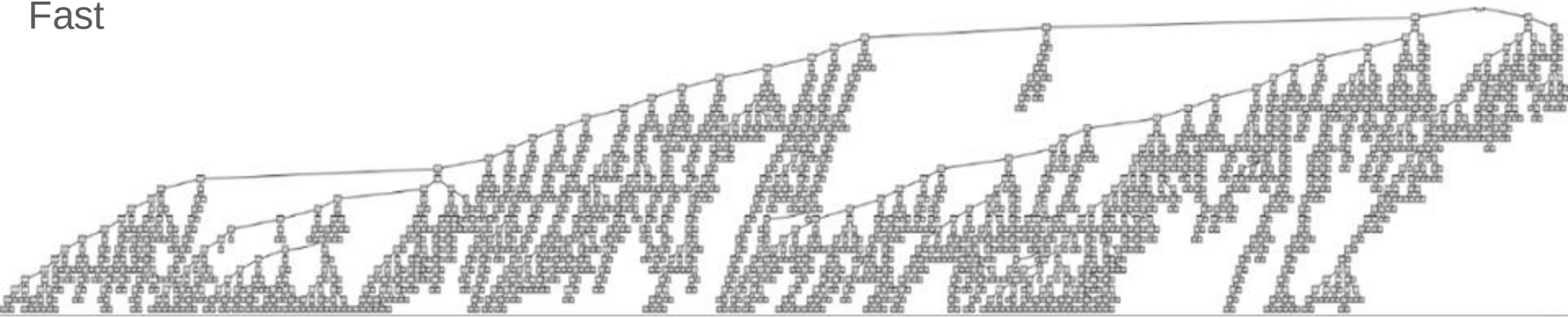
Fish

What animal?

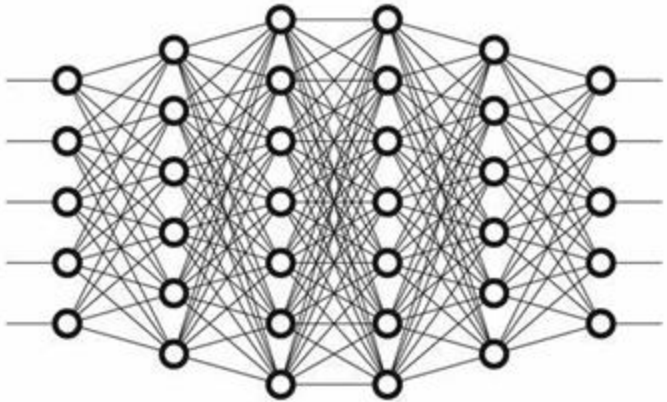
7 votes for fish
4 votes for cow
1 vote for bird

fish

Fast



Slow (to train)



Why

Train once (Top to bottom

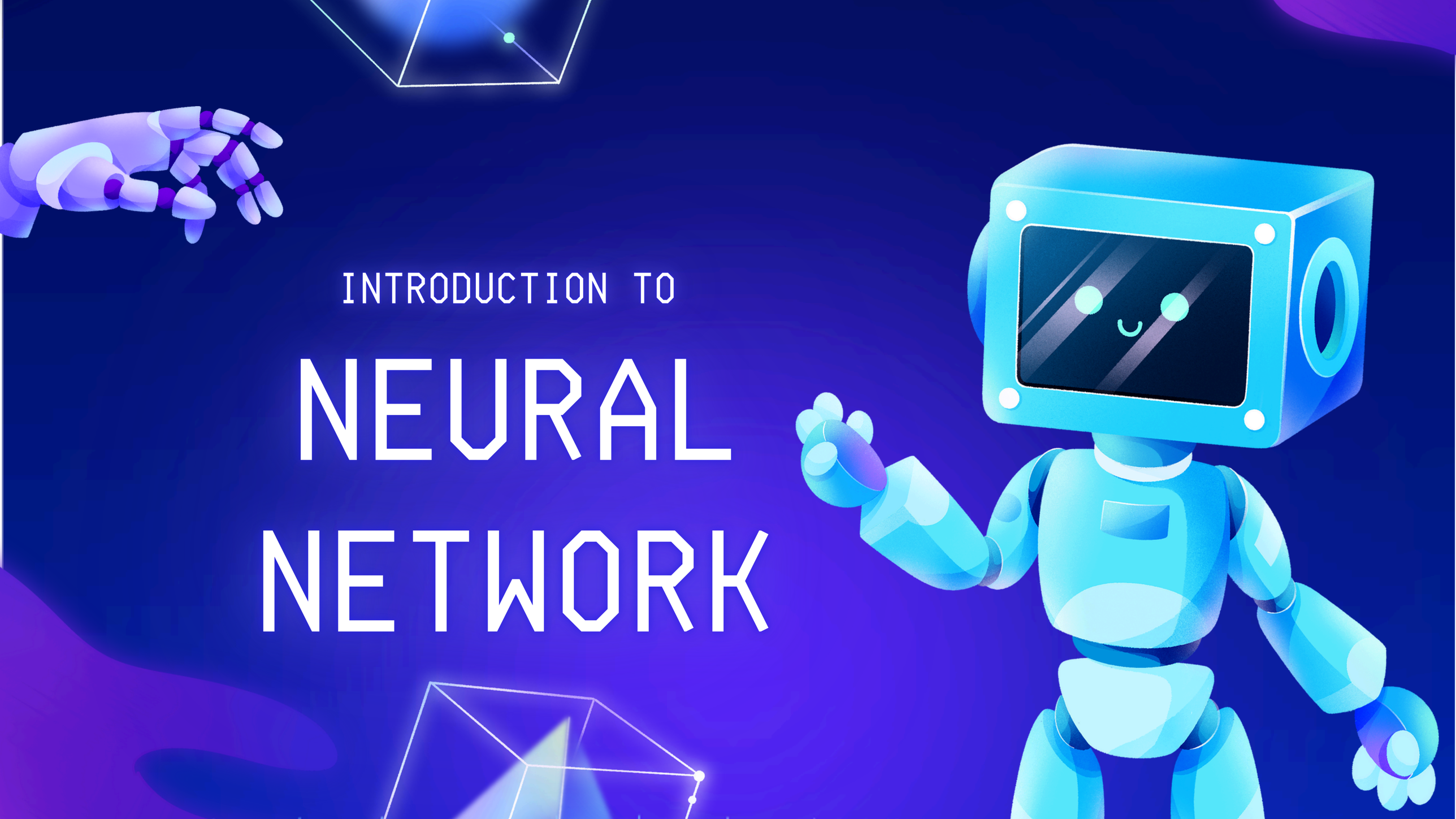
Data doesn't slow it (much)

Pros
Fast
Readable (kinda)
Good for structured data

Bads
Unstructured data (what is a pixel?)
Doesn’t run much faster than others

INTRODUCTION TO

NEURAL NETWORK



REGISTER

GREATEST HACKATHON ON EARTH

HACKPSU

Penn State's Largest 24-Hour Hackathon

FALL 2024

OCTOBER 12-13 AT THE BUSINESS BUILDING

ZERO coding experience? NO PROBLEM!
Come to the event to learn new skills and technologies!
YOU can win prizes at HACKPSU and have
amazing networking opportunities
along with free merchindise!

REGISTER NOW at HackPSU.org!

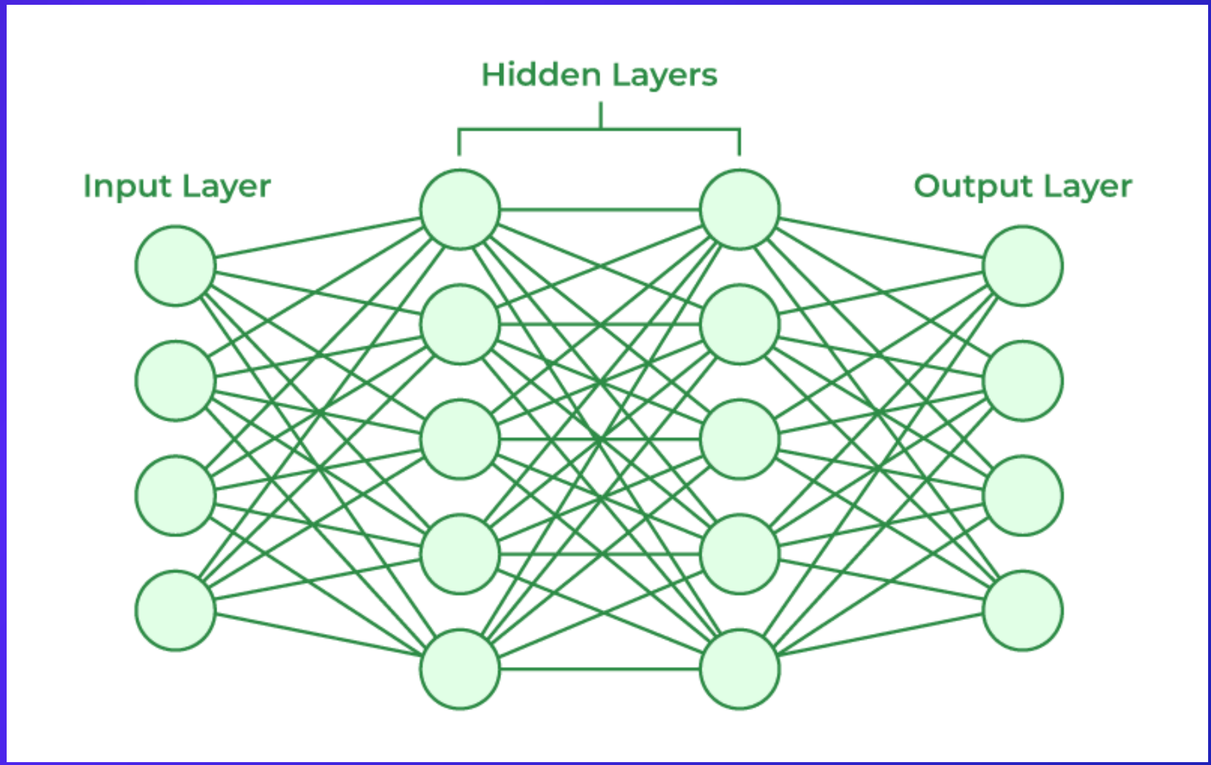
INSTAGRAM

Agenda

- What is **Neural Network**
- Understanding **Weights** and **Biases**
- The **Feedforward Process**
- **MNIST** Dataset Overview
- Activation Function

What is a Neural Network?

- Inspired by the human brain
- Processes information in layers
- image classification, language translation, and more



Weights and Biases

Weight: Weights are numbers in the network that show how **important** each **connection** between neurons is..

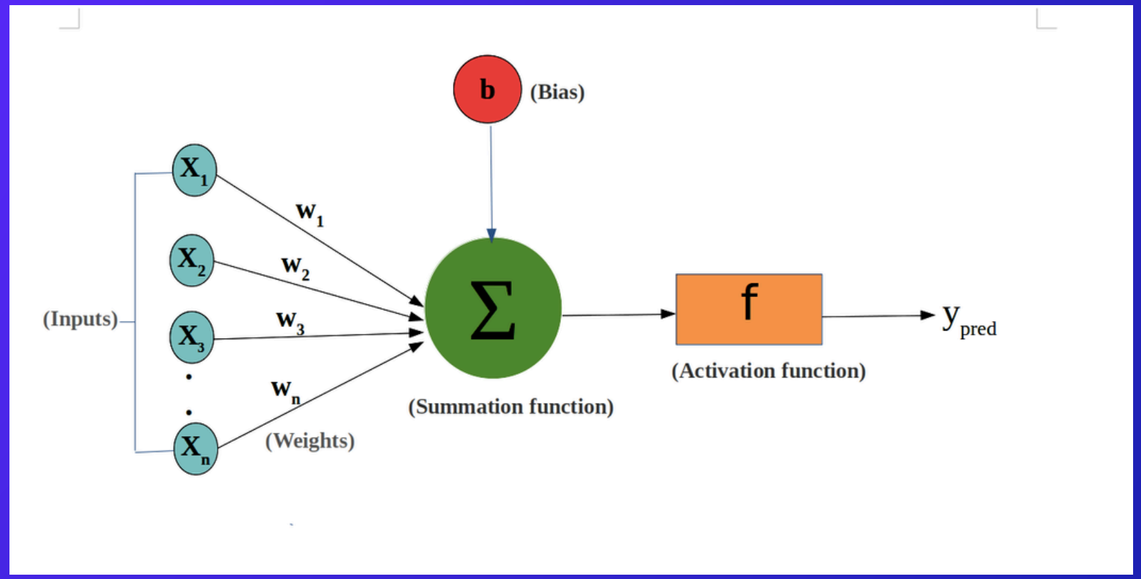
Bias: Bias is a number added to the result that helps the network make better decisions, even if all the inputs are zero.

Housing Price prediction Dataset

Weight : the size of the house

Bias : the location of the house

The number of the weight and bias are learned from **Backpropagation**



Questions ?

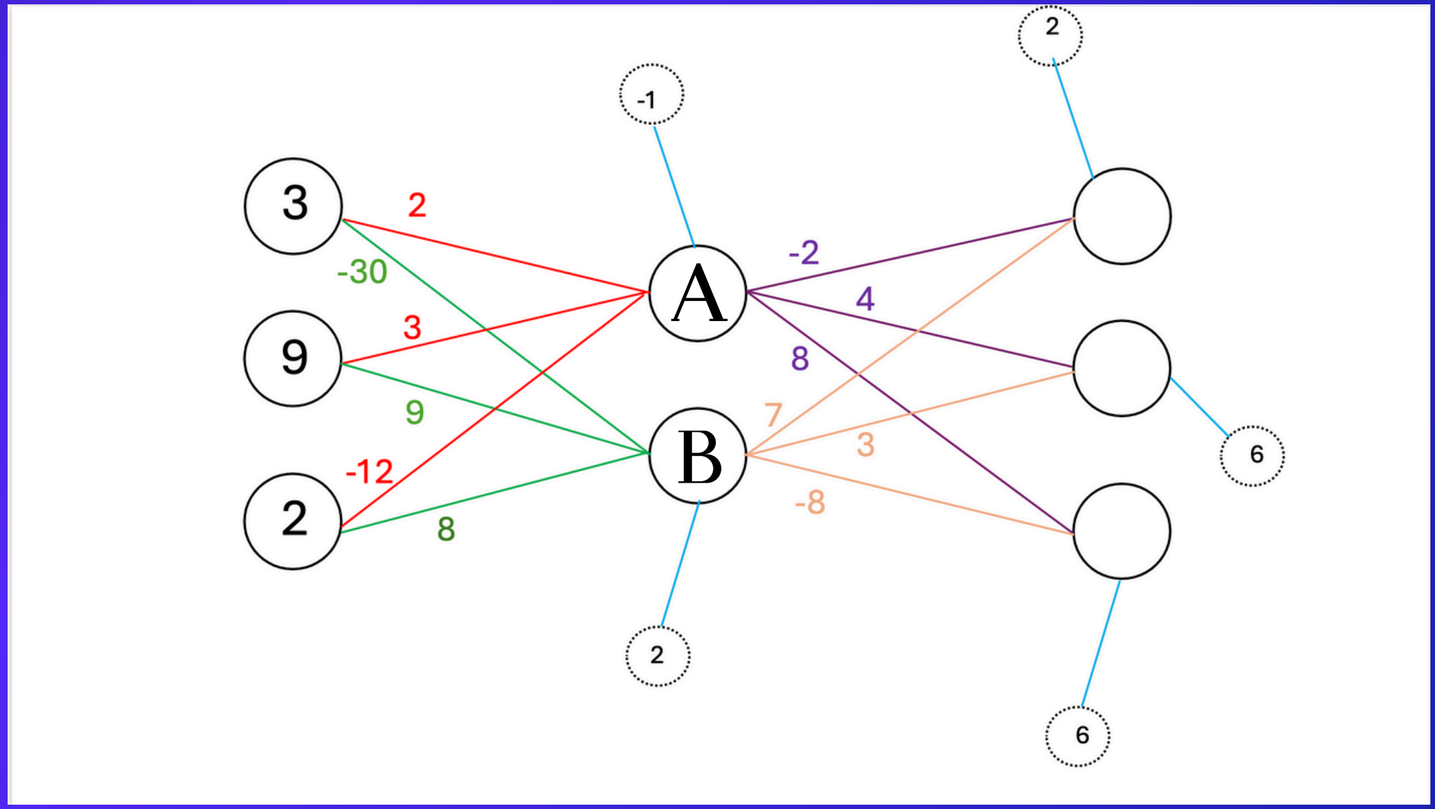
- What is neural network
- Understanding Weights and Biases
- The Feedforward Process
- MNIST Dataset Overview
- Activation Function

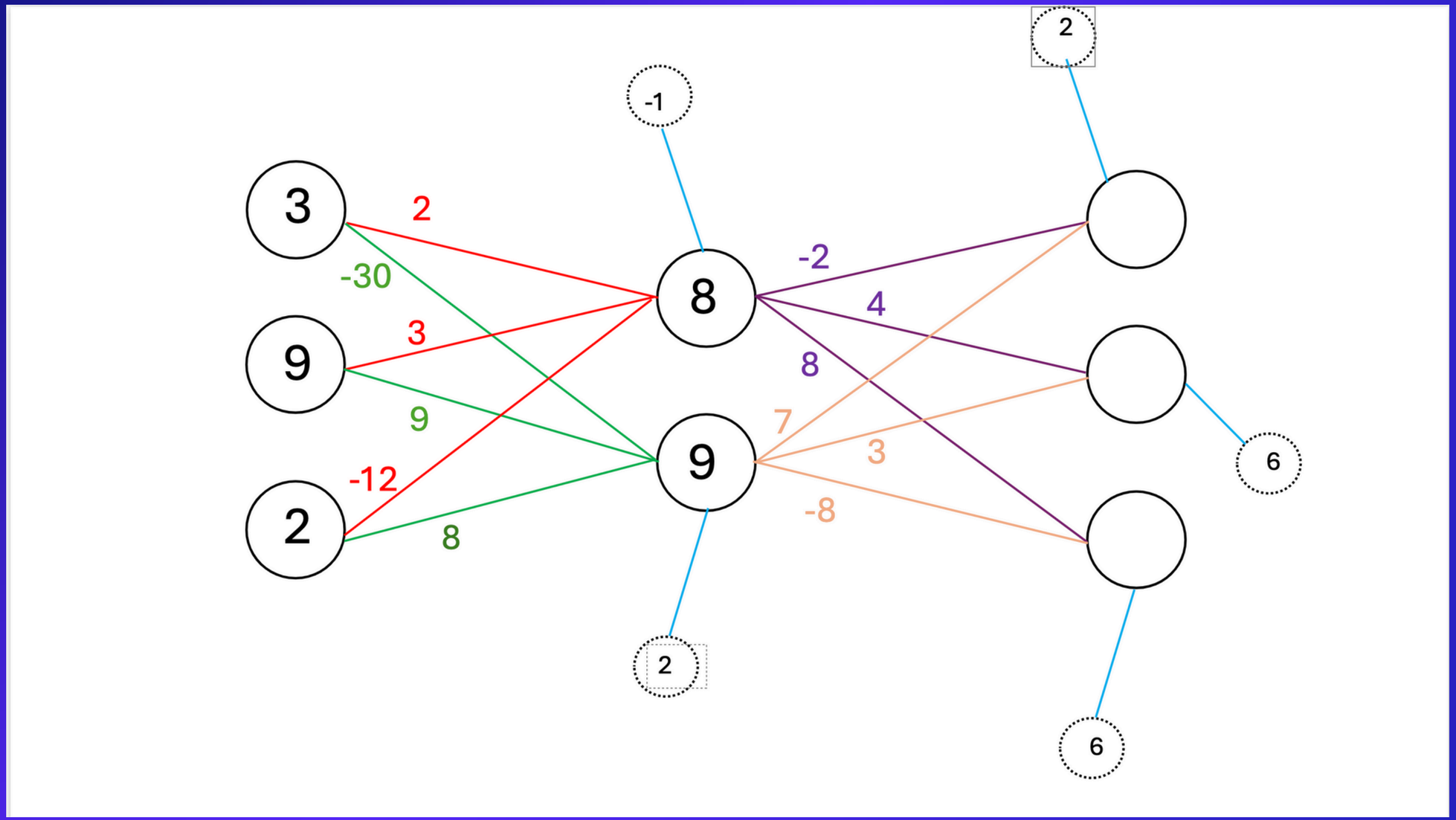
Weights and Bias Example

- **Formula: Z = w1 * x1 + w2 * x2 + ... + wn * xn + b**

  - **z is the output or result**
  - **w is the weight**
  - **x is the input value**
  - **b is the bias**

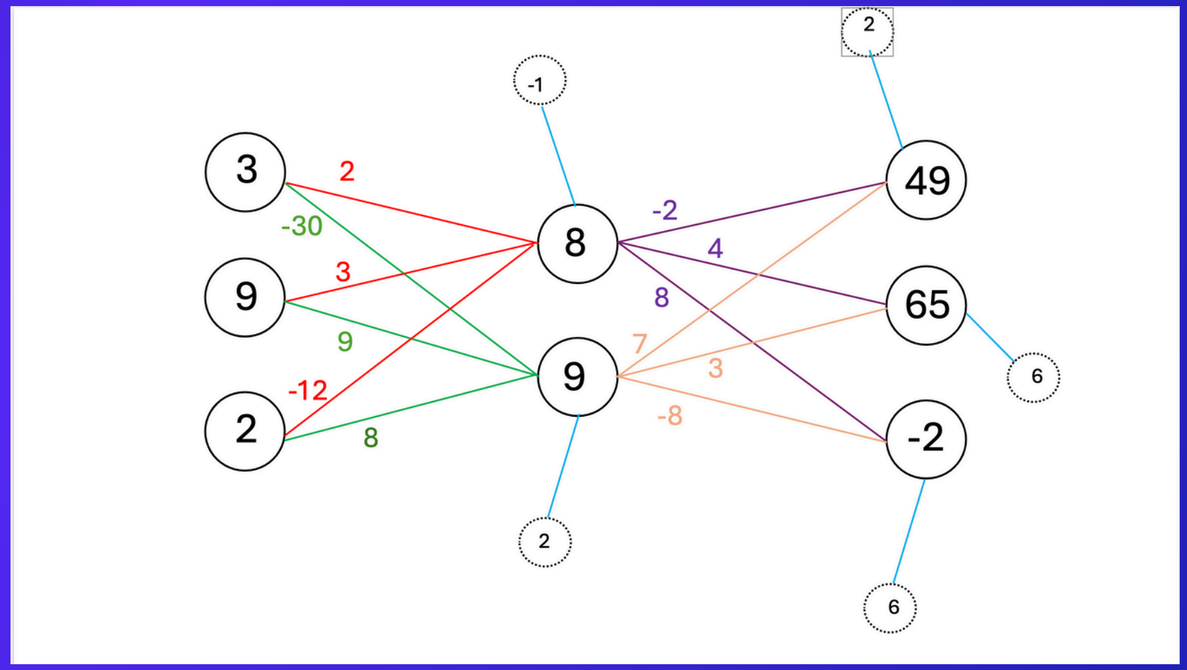
What is the activation number for A and B?




2
-1
3
2
-30
8
-2
4
9
3
8
9
7
3
-12
2
-8
8
2
6
6

The Feedforward Process

- Data moves through the network layer by layer.
- Weighted sum + bias is passed through each neuron.
- Final output: Prediction or classification.

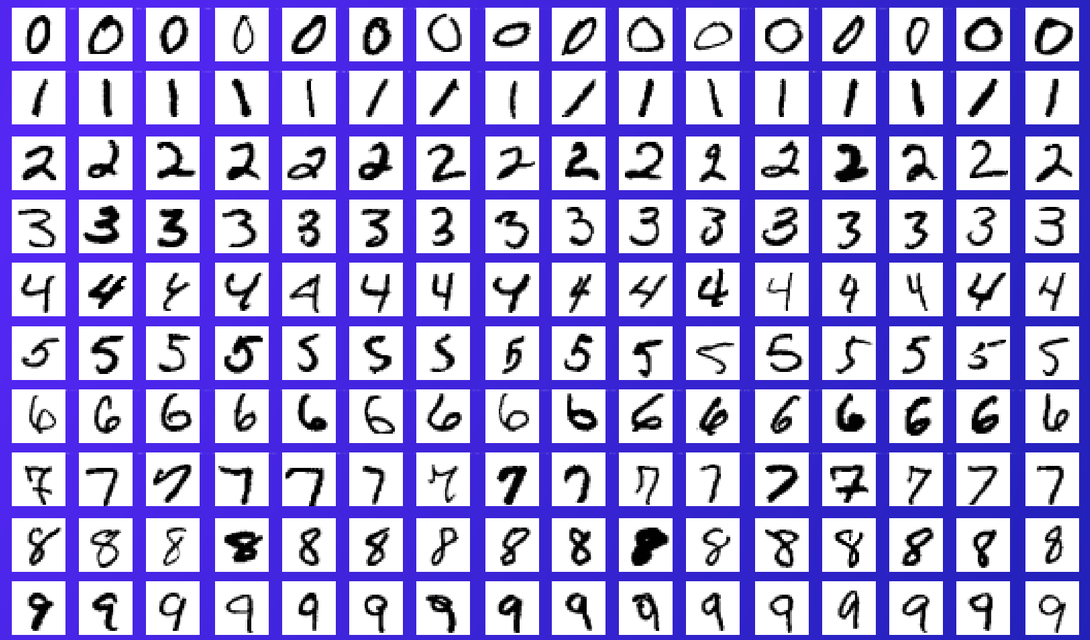


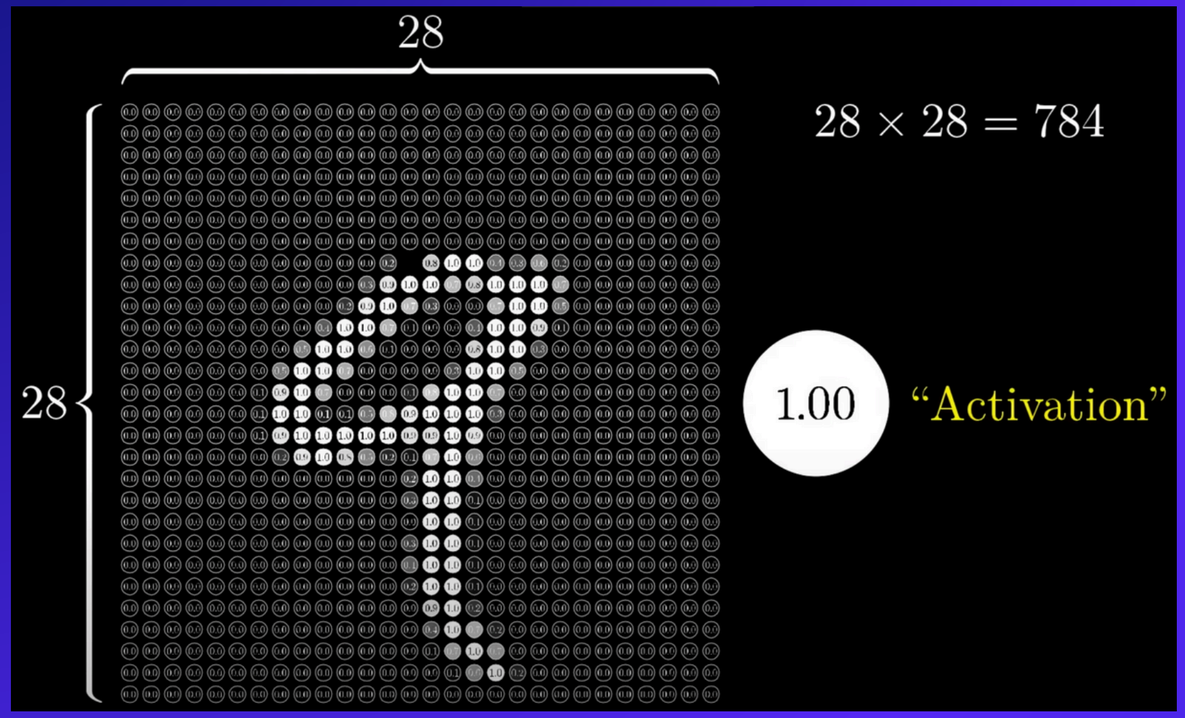
Questions ?

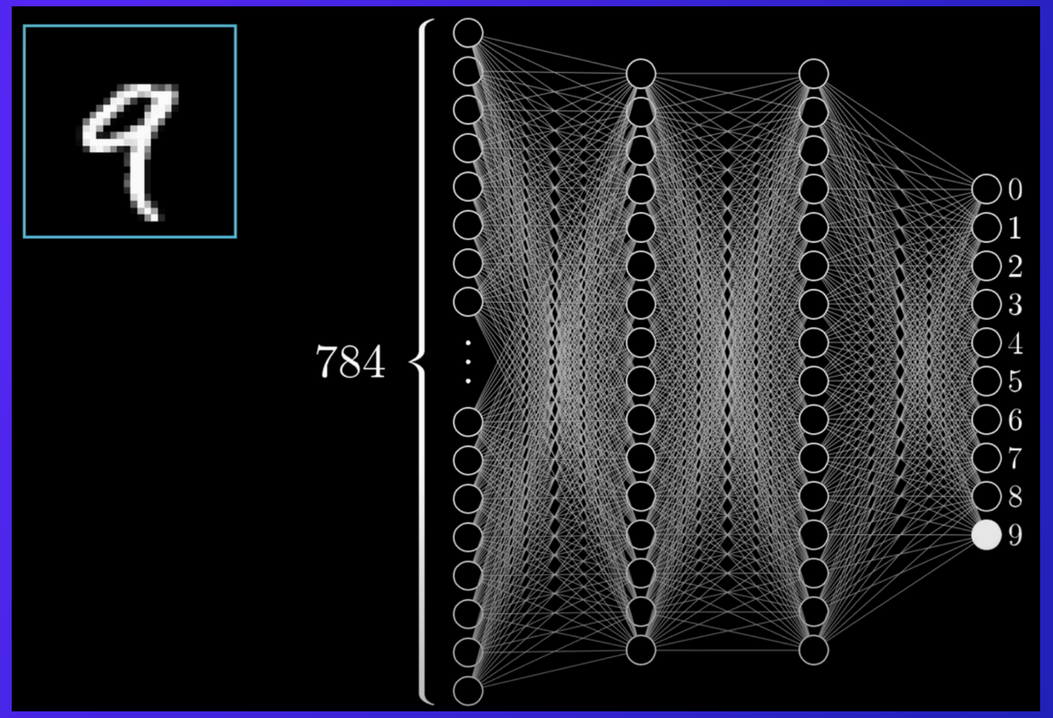
- What is neural network
- Understanding Weights and Biases
- The Feedforward Process
- MNIST Dataset Overview
- Activation Function

MNIST Dataset

- Collection of 70,000 handwritten digits (0-9).
- Each image is 28x28 pixels, grayscale.
- Goal: Train the network to recognize digits.







"Source: 3Blue1Brown (2017). But what is a Neural Network? [YouTube Video]. Retrieved from https://youtu.be/aircAruvnKk"

TensorFlow vs PyTorch

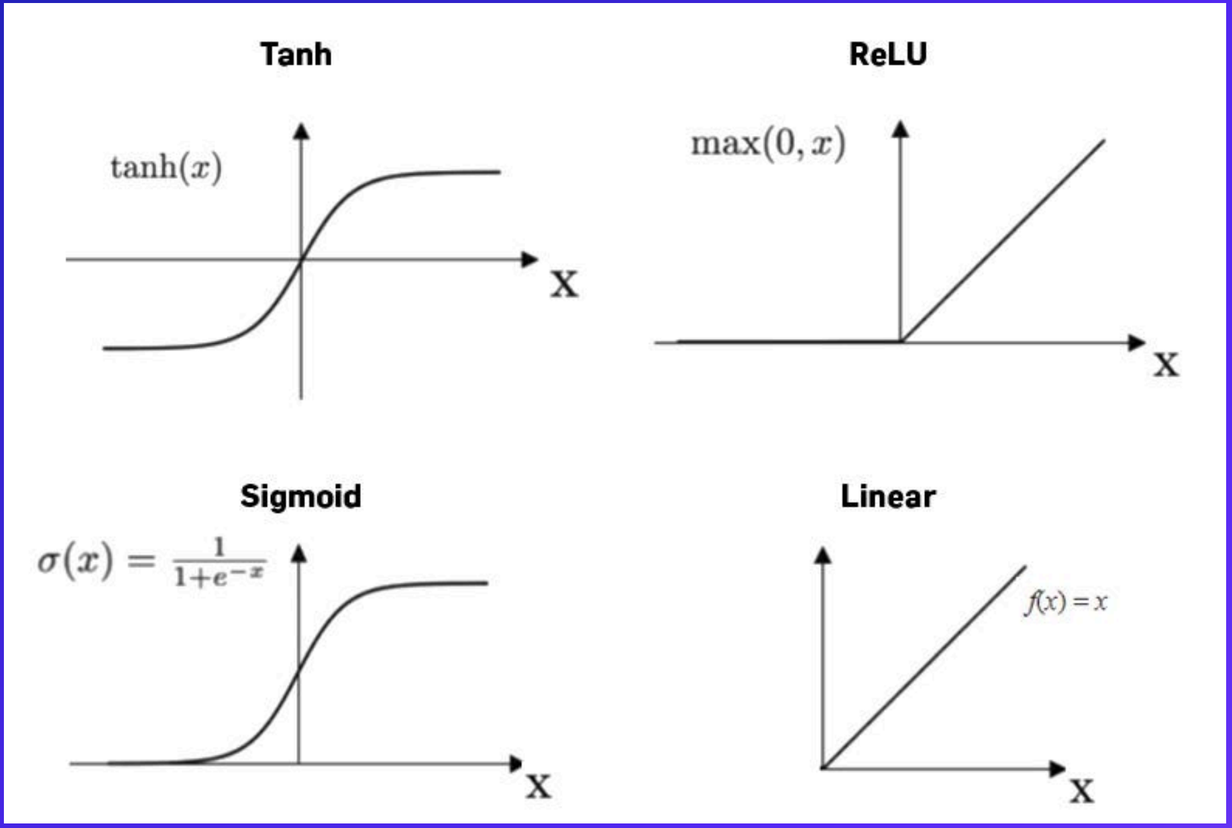
```tensorflow
#TensorFlow
#Declaring the type of model
model = tf.keras.models.Sequential()

#Adding Layers to the model
model.add( layers.Input(np.shape(X_train)[1:]) )
model.add( layers.Flatten() )
model.add( layers.Dense(16, activation = 'relu') )
model.add( layers.Dense(16, activation = 'relu'))

#Final Layer, where the prediction will be made
model.add( layers.Dense(10, activation = 'softmax') )
```

```pytorch
#PyTorch
class Classifier(nn.Module):
  def __init__(self, batch_size):
    super().__init__()
    self.dense = nn.Sequential(
        nn.Flatten(),
        nn.Linear(2048, 256),
        nn.Sigmoid()
    )

    self.dense = nn.Sequential(
        nn.Flatten(),
        nn.Linear(256, 10),
        nn.Sigmoid()
    )

    self.dense = nn.Sequential(
        nn.Flatten(),
        nn.Linear(256, 1),
        nn.Sigmoid()
    )

    self.batch_size = batch_size
  def forward(self, x):
    x = self.dense(x)
    x = self.dense(x)
    x = self.dense(x)
    x = self.dense(x)
    return x.squeeze()
```

Questions ?

- What is neural network
- Understanding Weights and Biases
- The Feedforward Process
- MNIST Dataset Overview
- Activation Function

Activation function

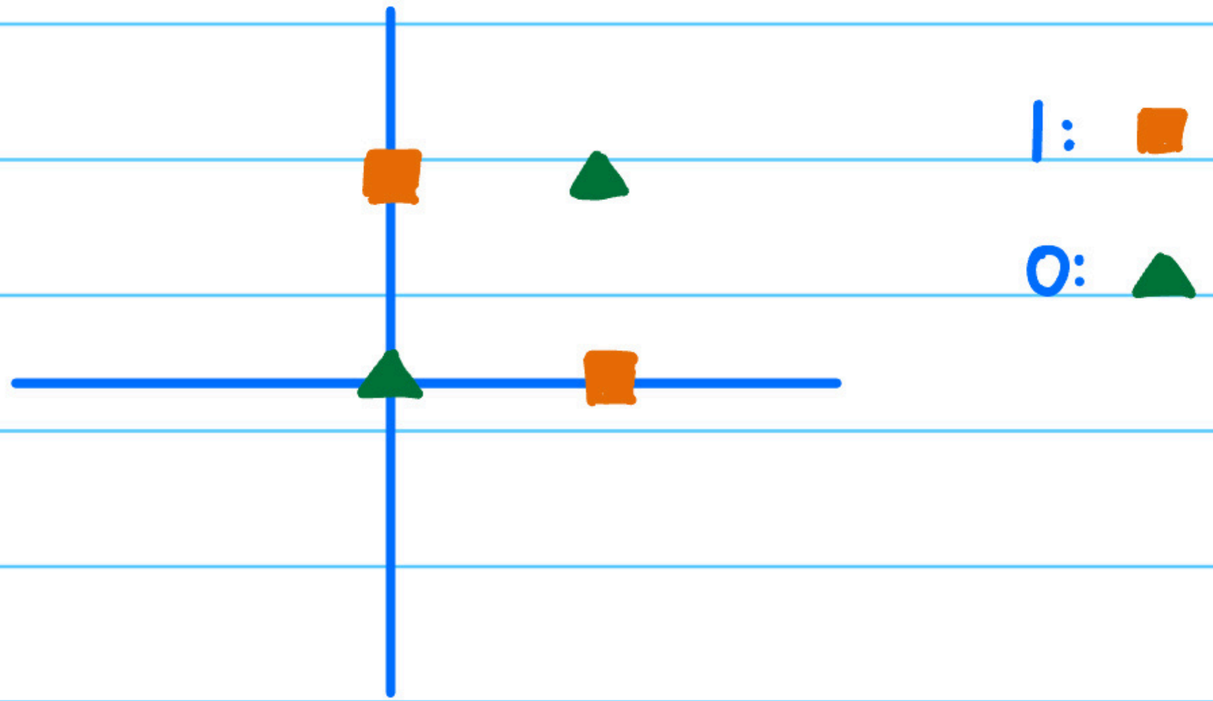


In machine learning, the **activation function** introduces **non-linearity** into the output of a neuron in a neural network

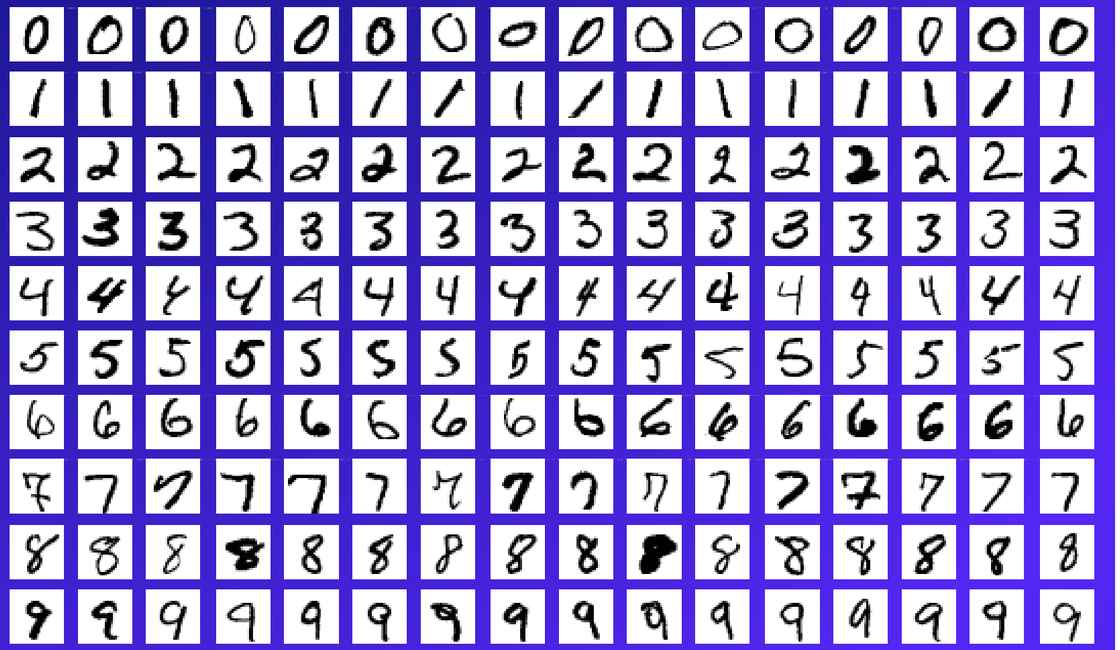
THANK YOU!

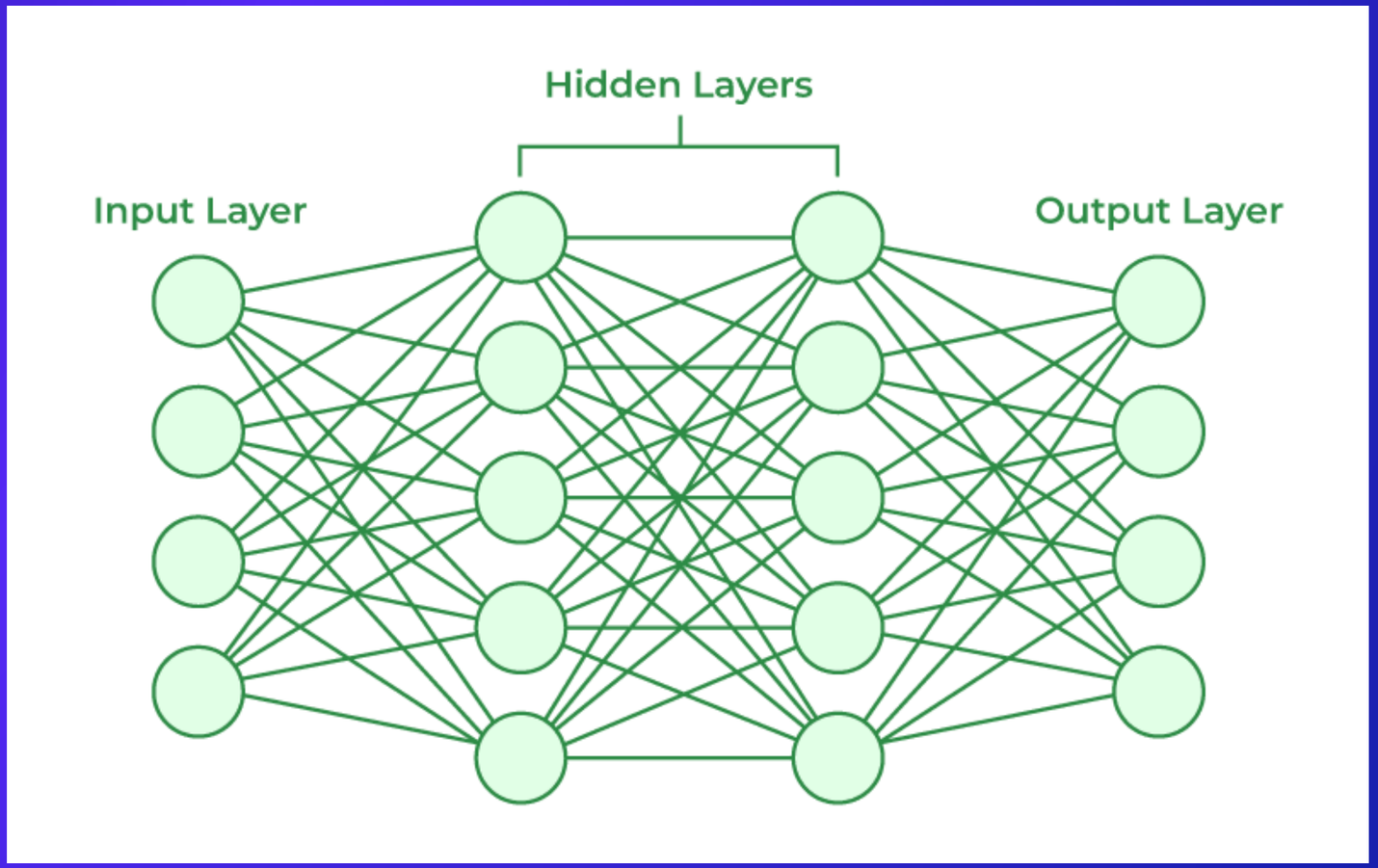
Question: can we apply linear activation on XOR logic?

Input		Output
X_1	X_2	Z
0	1	1
0	0	0
1	1	0
1	0	1



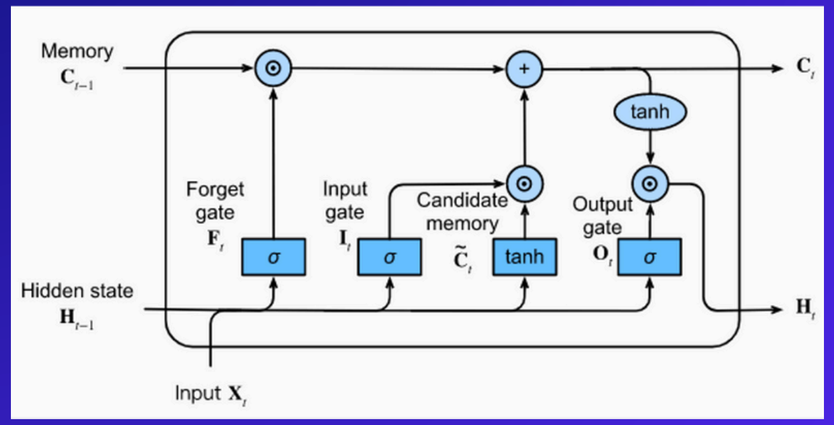
Overview of Neural Network



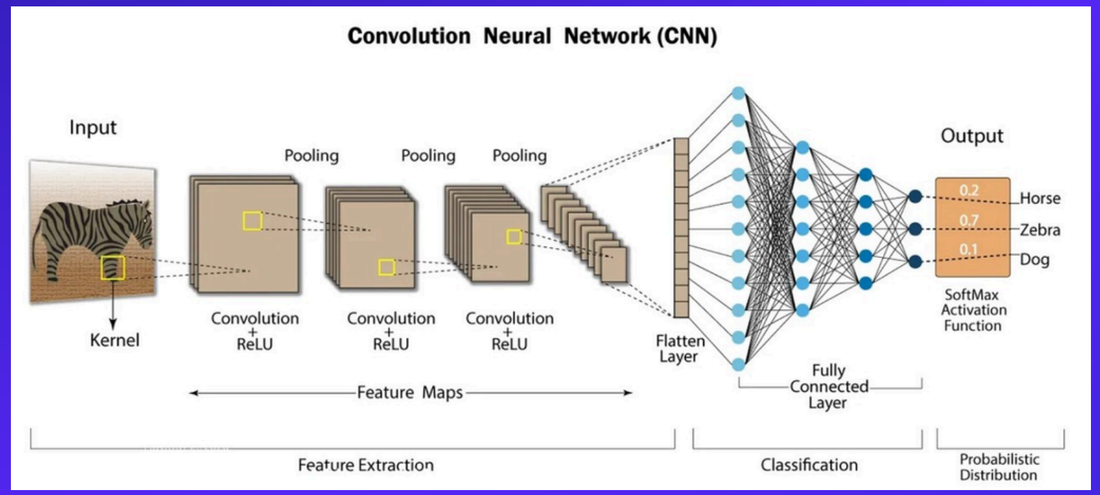


Different types of Neural Network

LSTM - Long Short Term Memory



CNN - Convolutional Neural Network



Transformer Neural Network

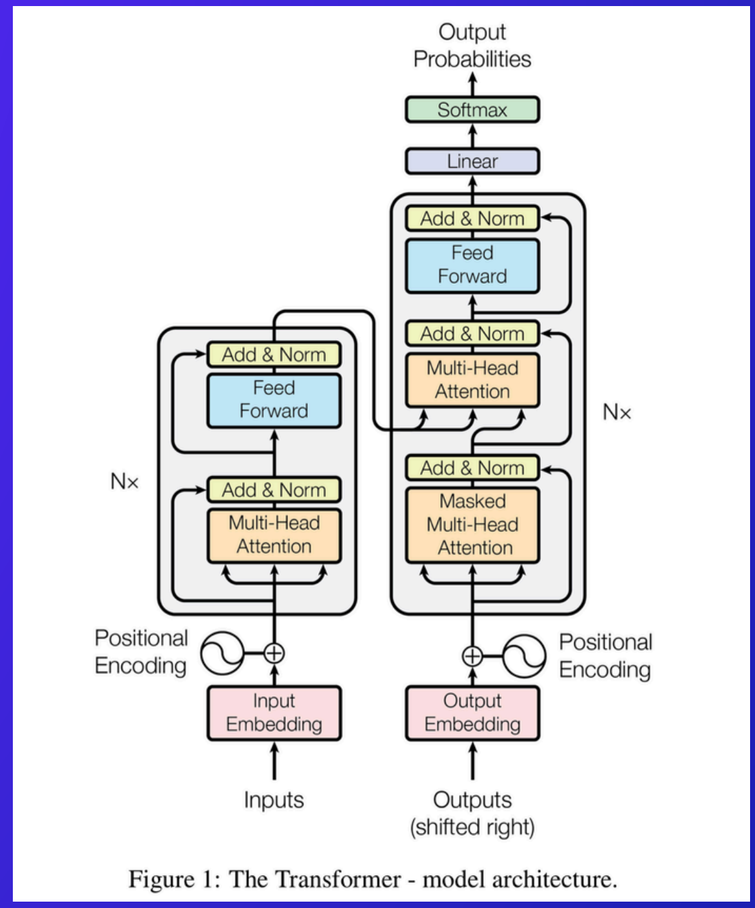
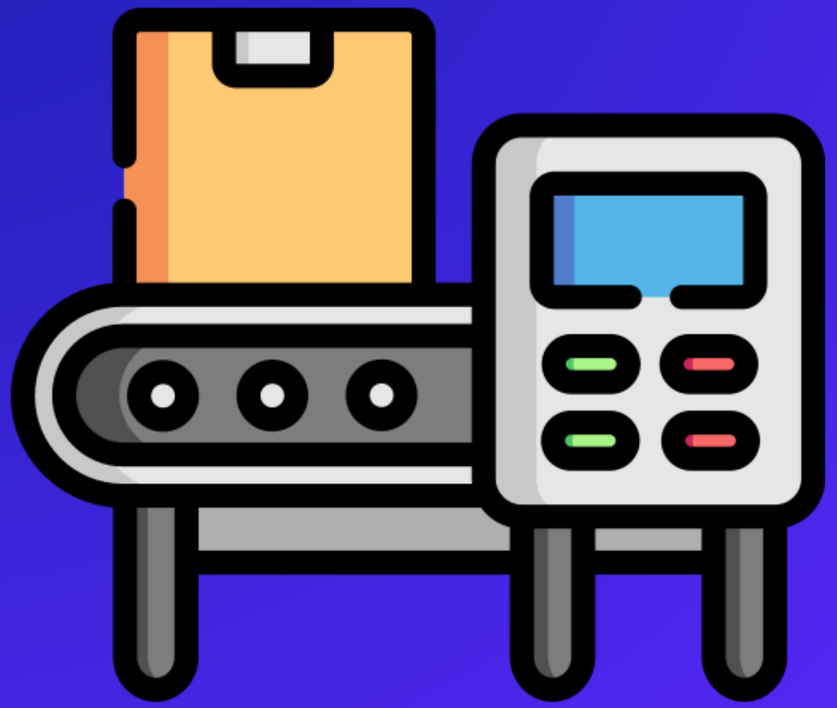
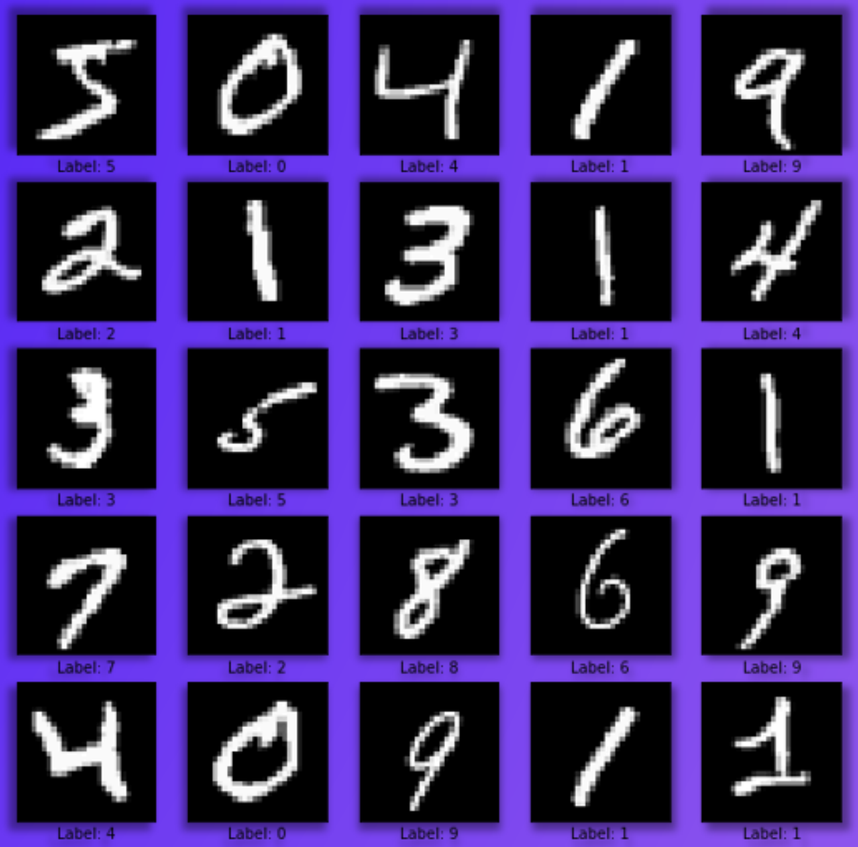


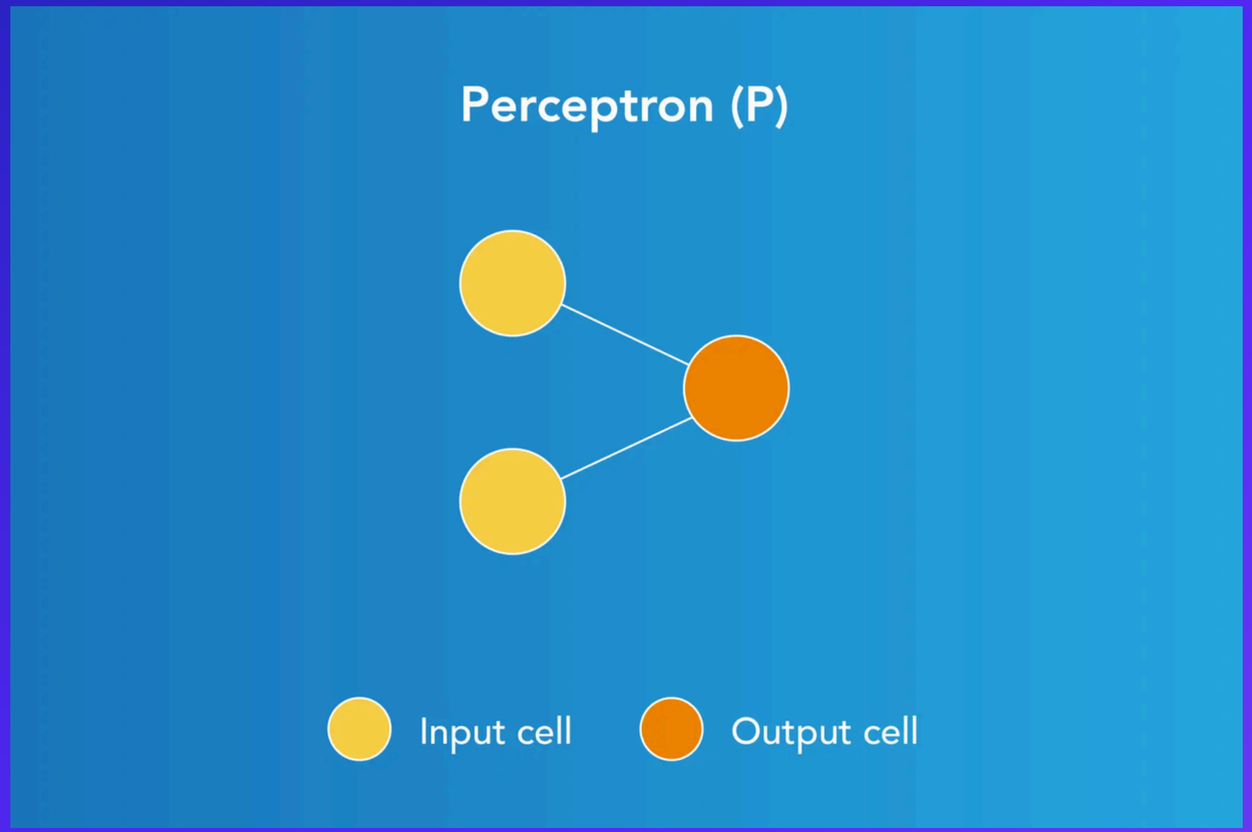
Figure 1: The Transformer - model architecture.

How does a machine learns these digit?





PERCEPTRON - THE BEGINNING OF NEURAL NETWORK

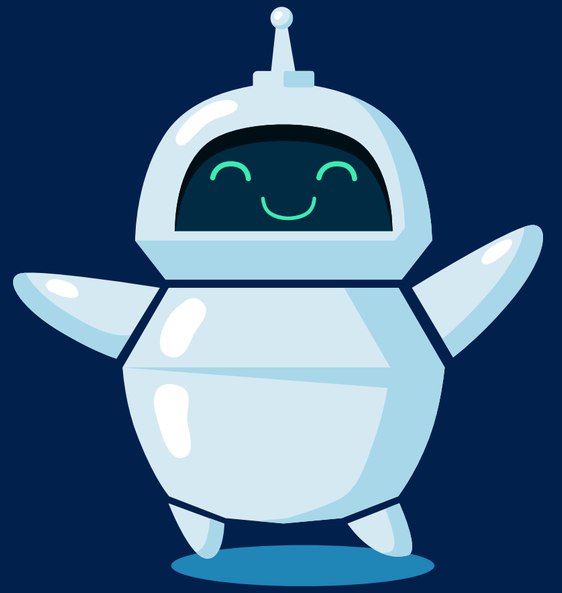


Can only slove **linear seprateable** problem

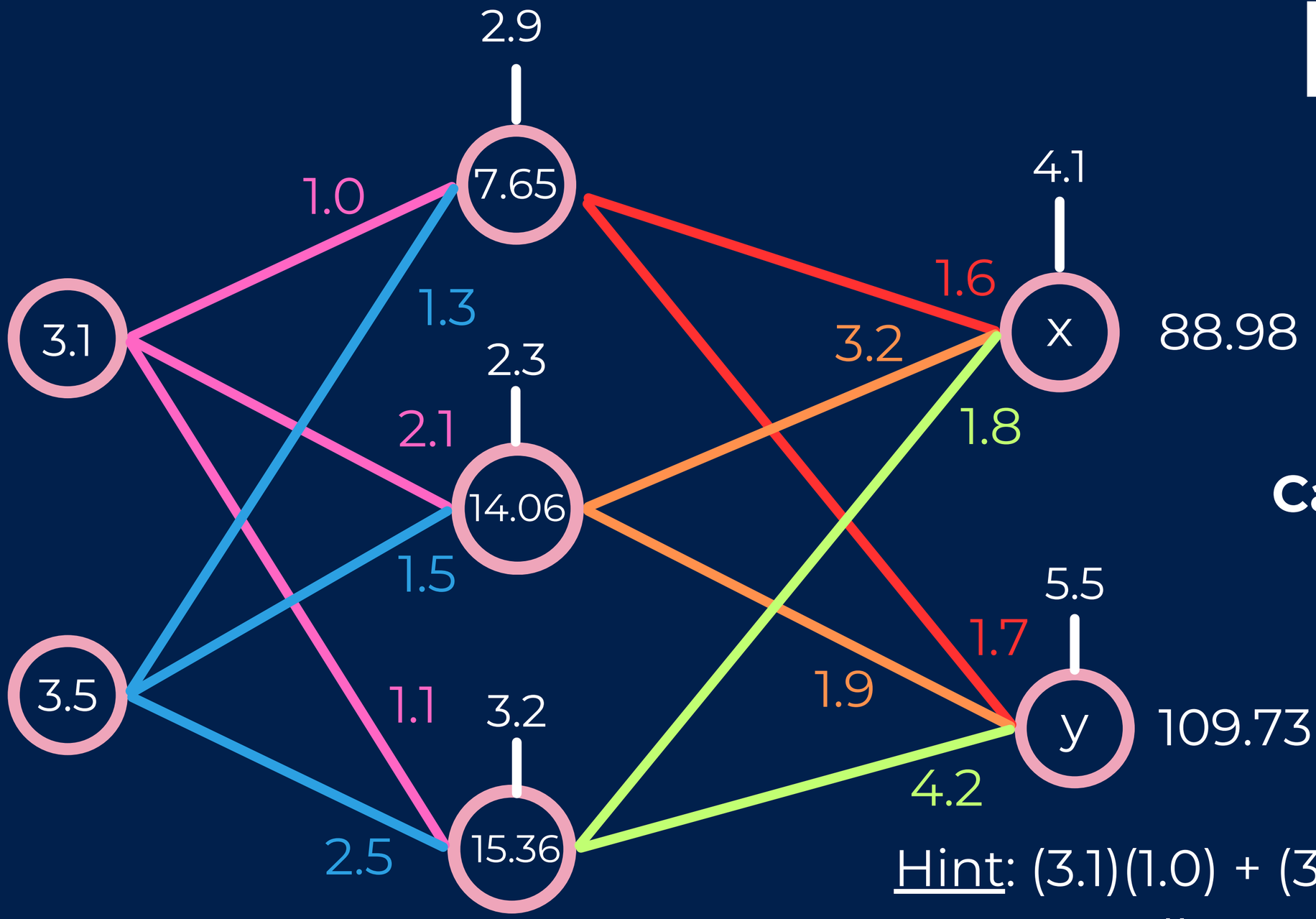
Ex: AND Logic

NAISS MLB Session 5

How Neural Networks Learn



Recap



Can you find x and y?

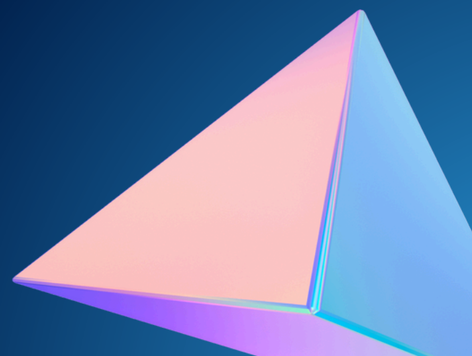
Hint: (3.1)(1.0) + (3.5)(1.3) + 2.9 = 7.65

Bonus: Tell me the formula used

The Cost Function

A way for a machine learning algorithm to know how “wrong” it is. Below is a simplified version of a common one used in ML.

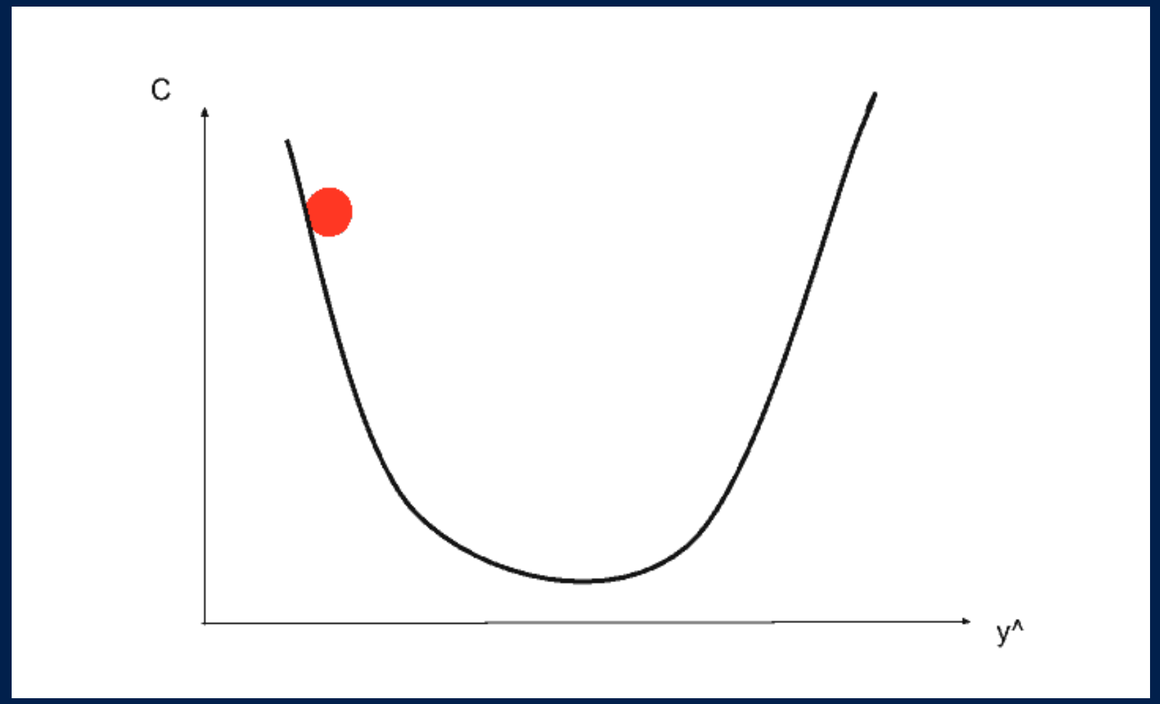
$$J(\dots) = \sum (predicted - actual)^2$$



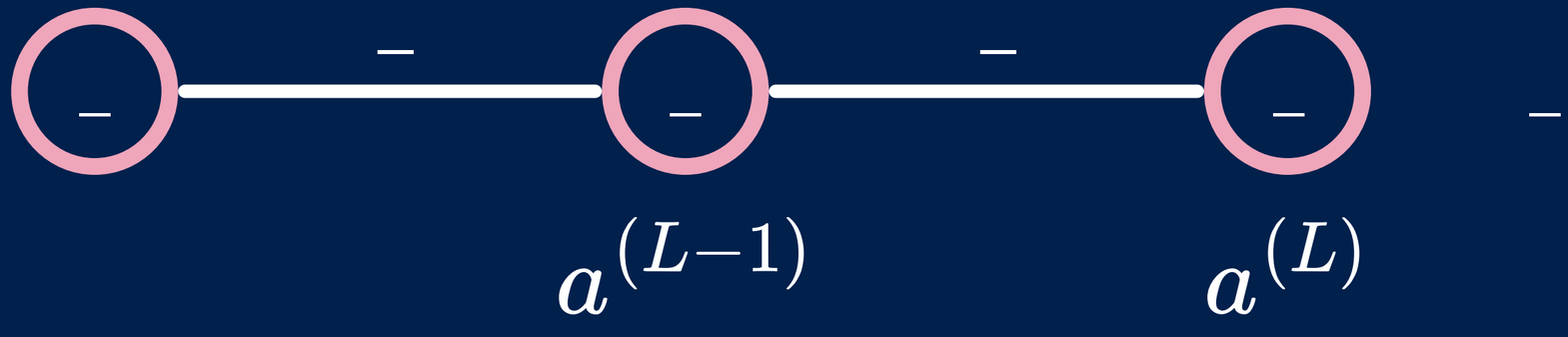
Gradient Descent

Since a high output of cost is not wanted, gradient descent is an algorithm used to **minimize** the cost function.

The lowest error we can get means that we get the weights and biases we want.

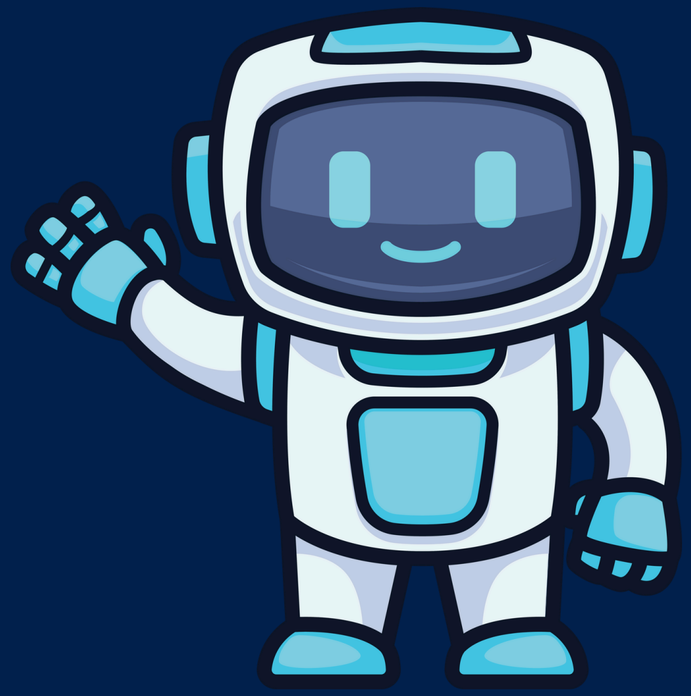


But how do we update?



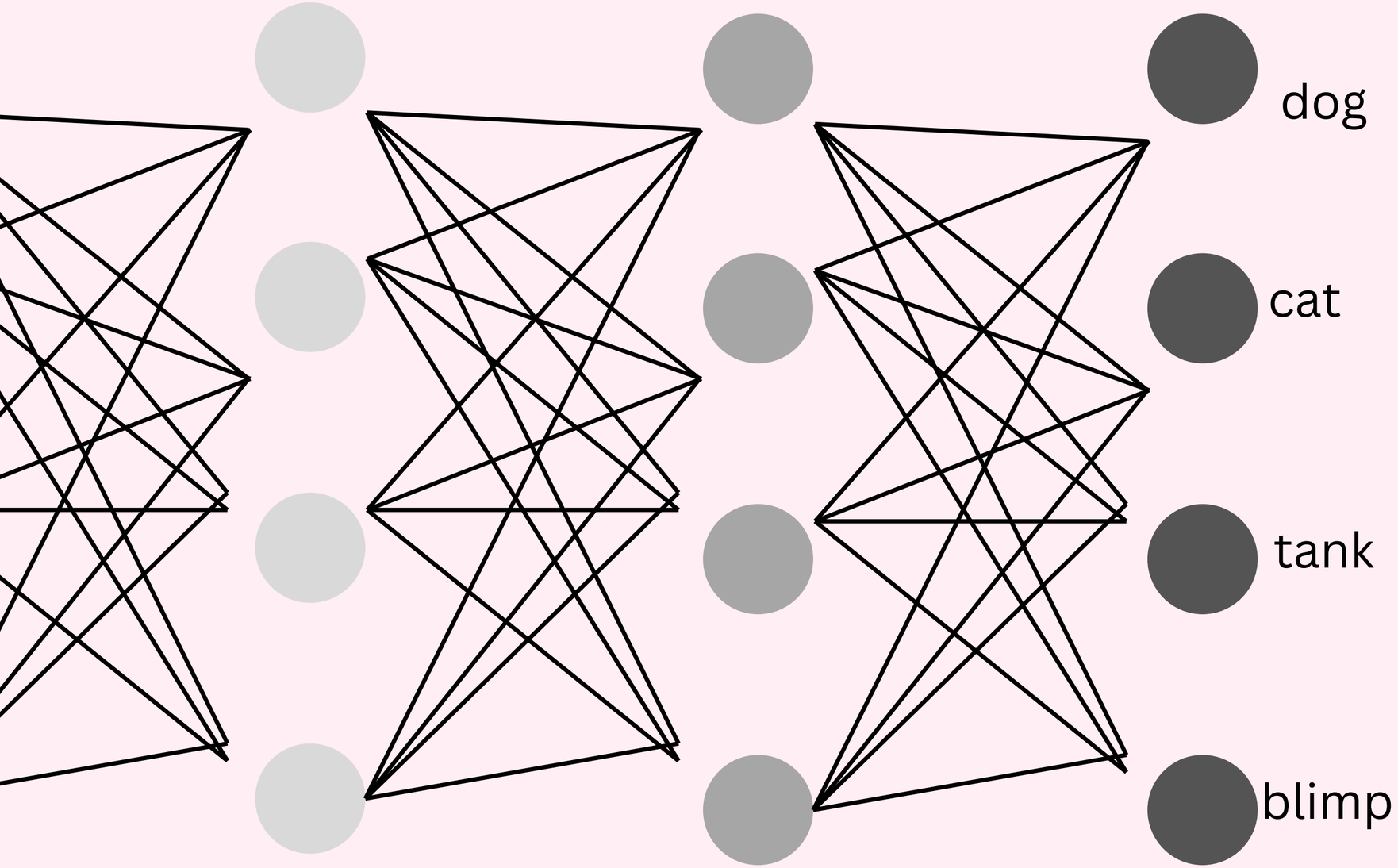
Let's say our predicted value here is: , but we want a target value of: . How do we update the weights and biases?

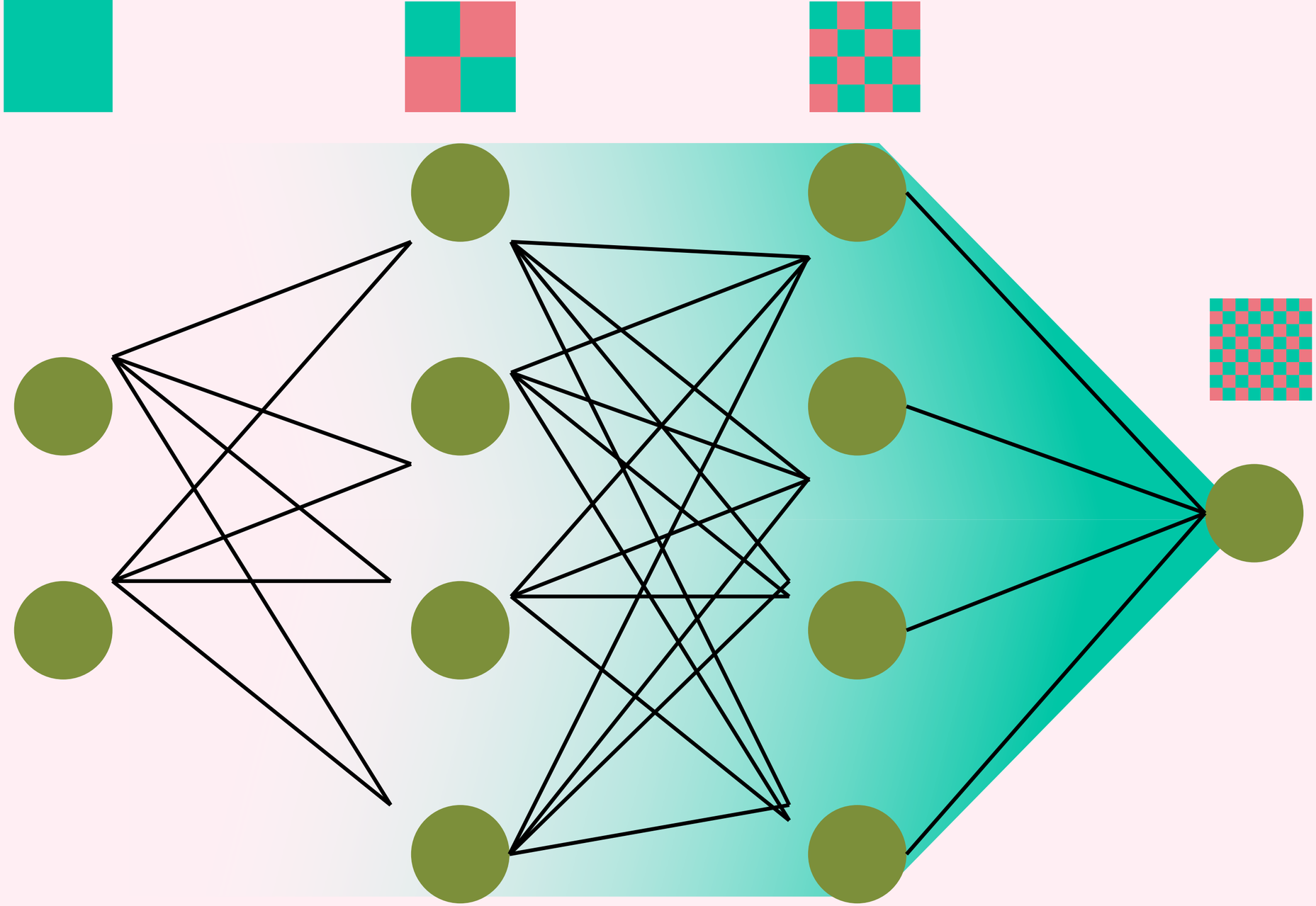
Thank You!



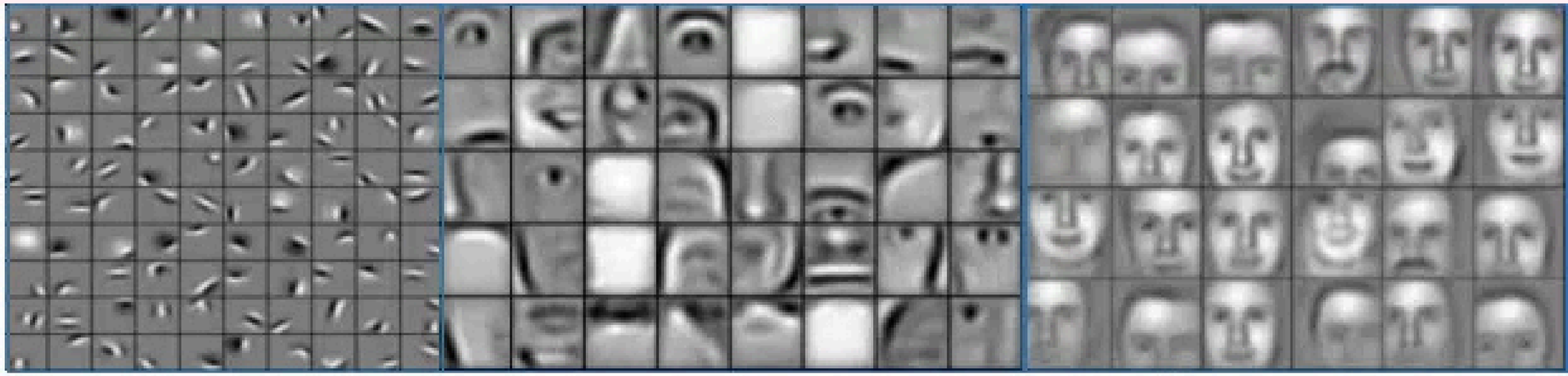
Fine-tuning

What is it?

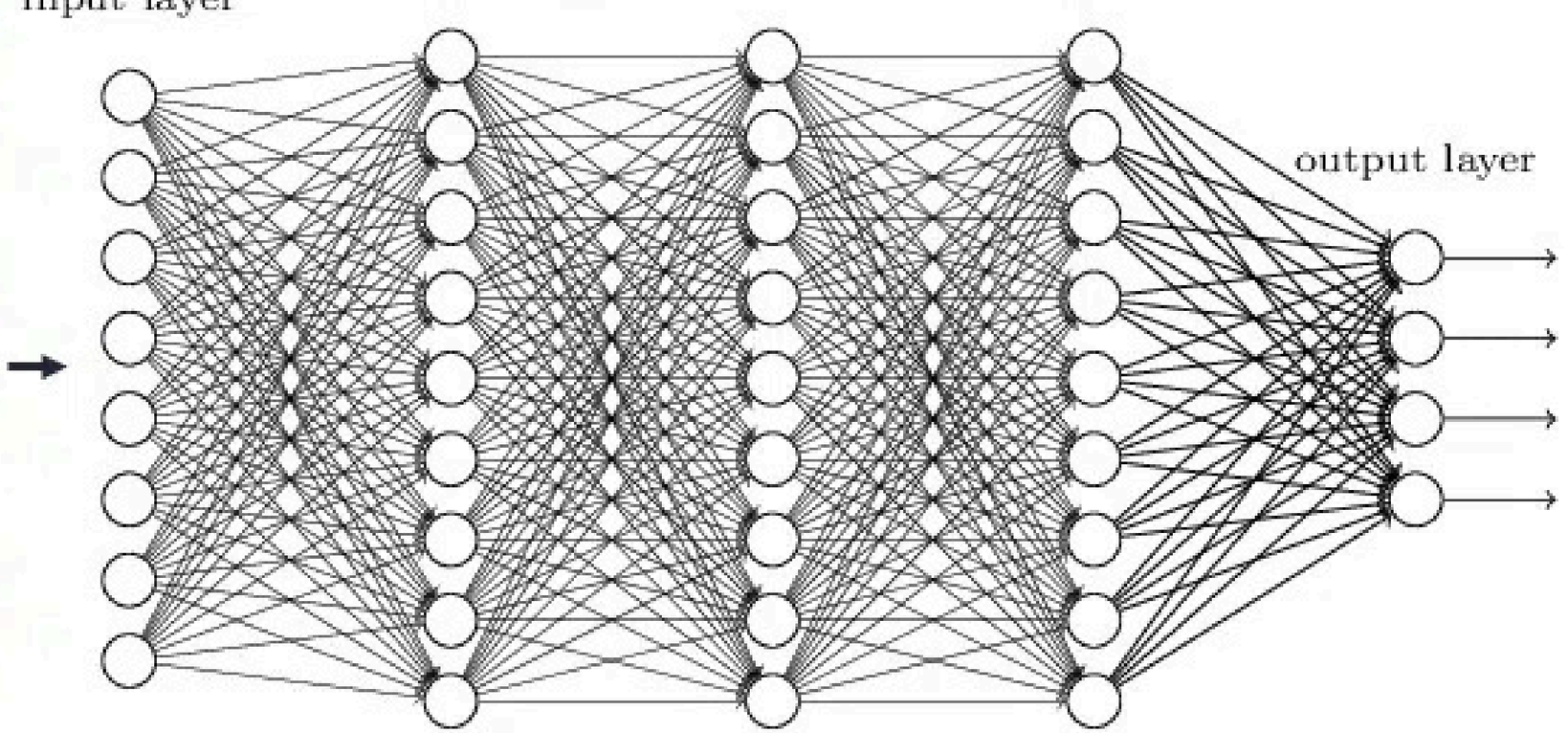

dog
cat
tank
blimp



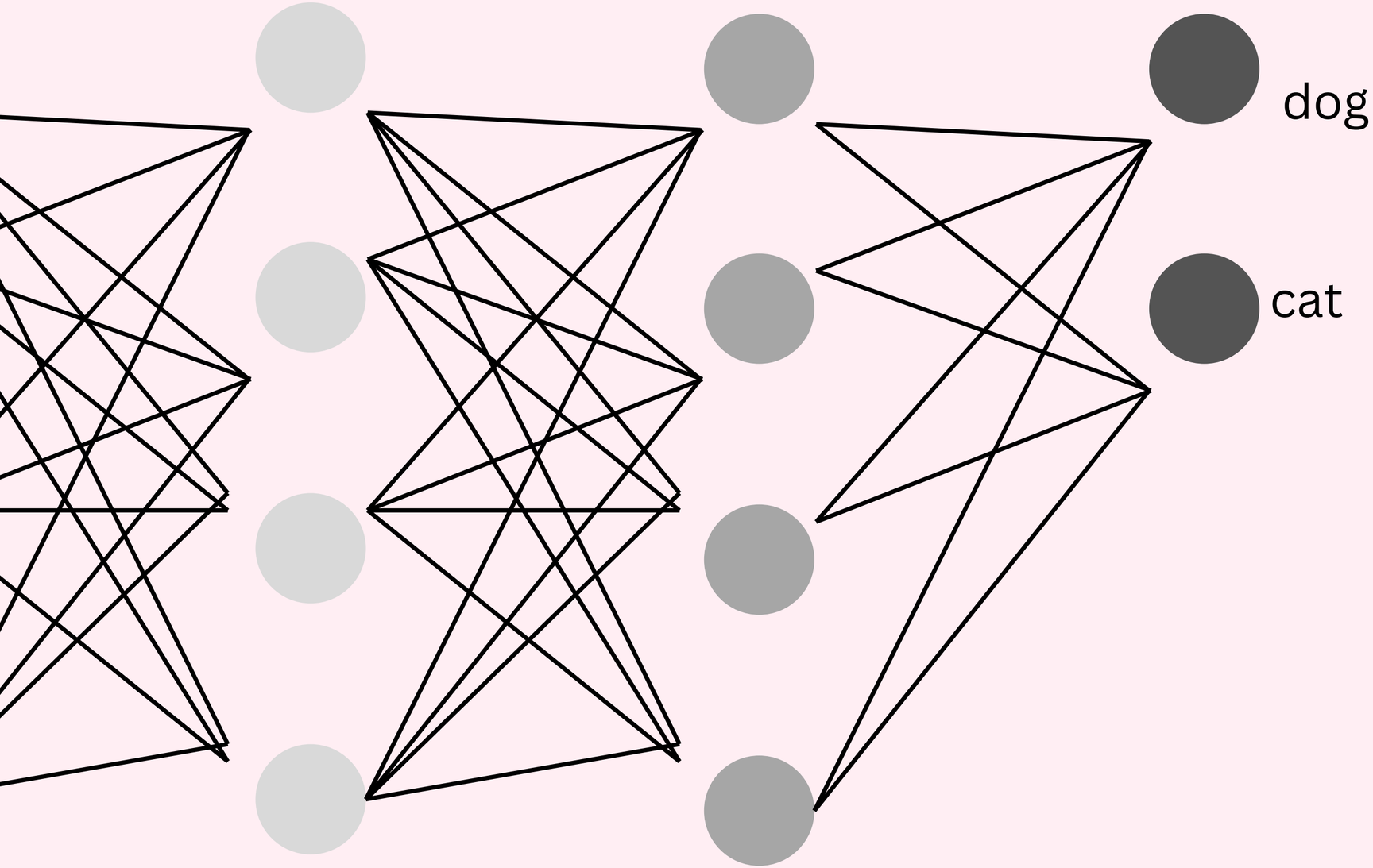
Deep neural networks learn hierarchical feature representations

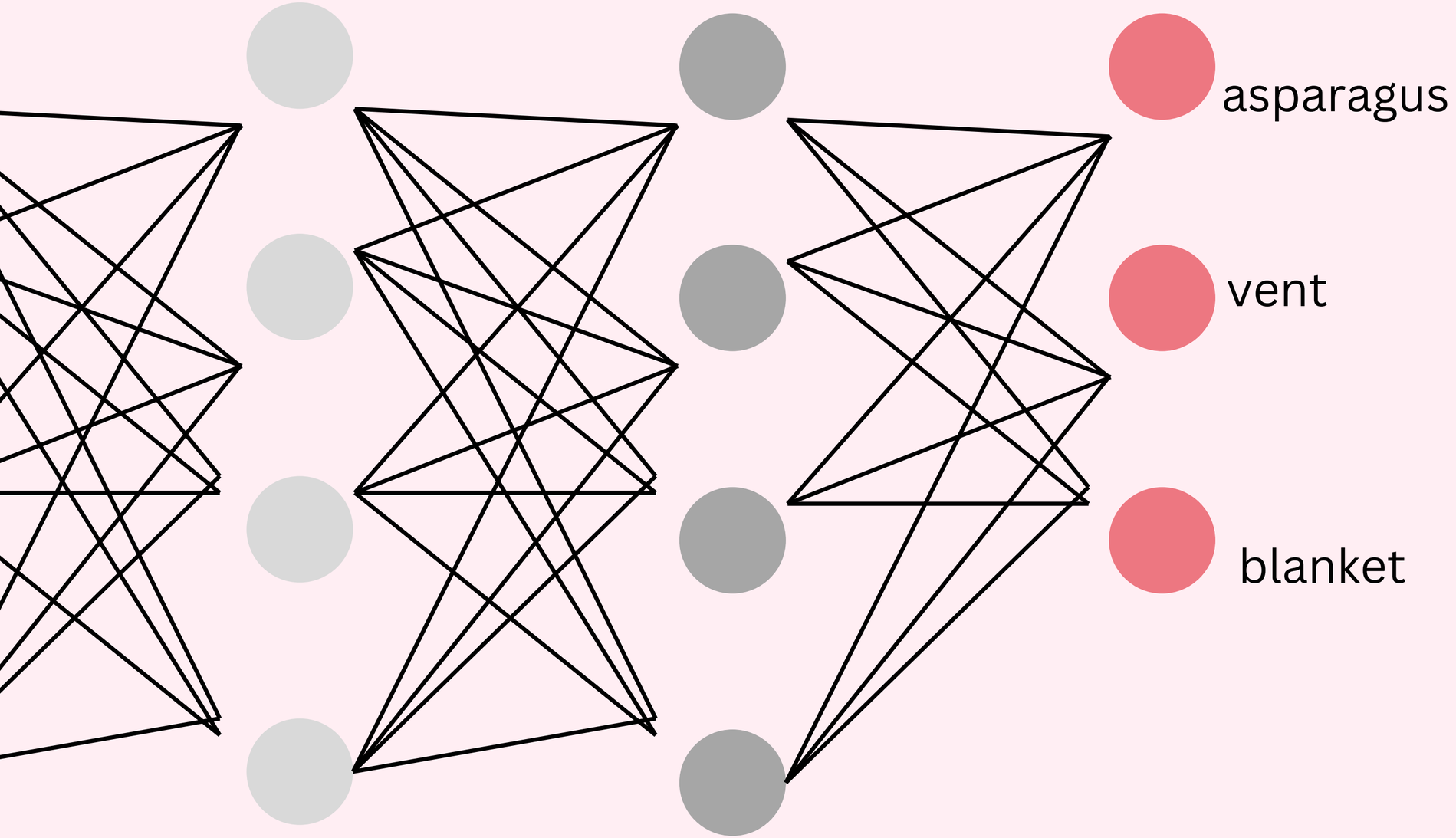


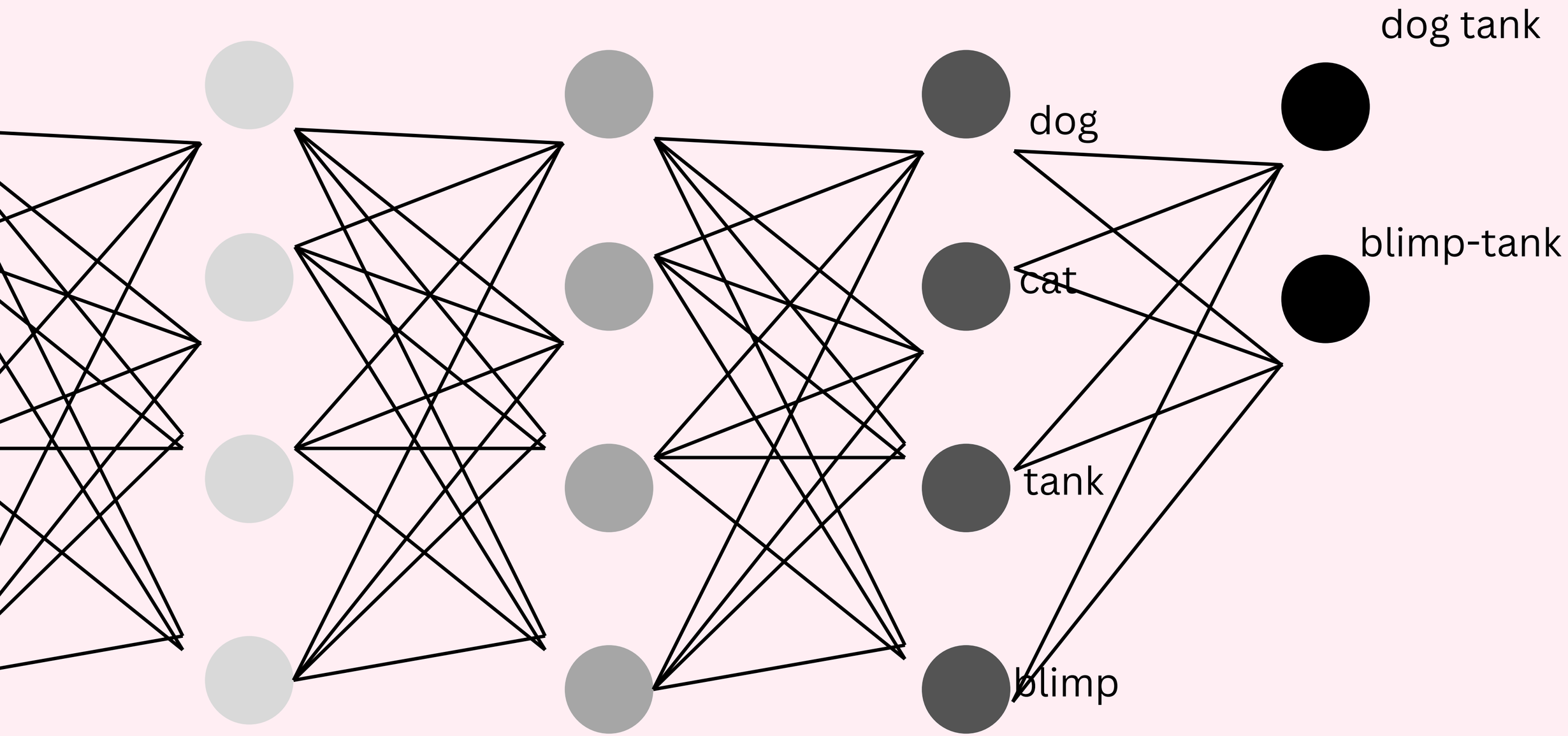
hidden layer 1 hidden layer 2 hidden layer 3



Top Layer Fine-tuning


dog
cat


asparagus
vent
blanket


dog tank
dog
blimp-tank
cat
tank
blimp

Pros / Cons

Ease

Dataload

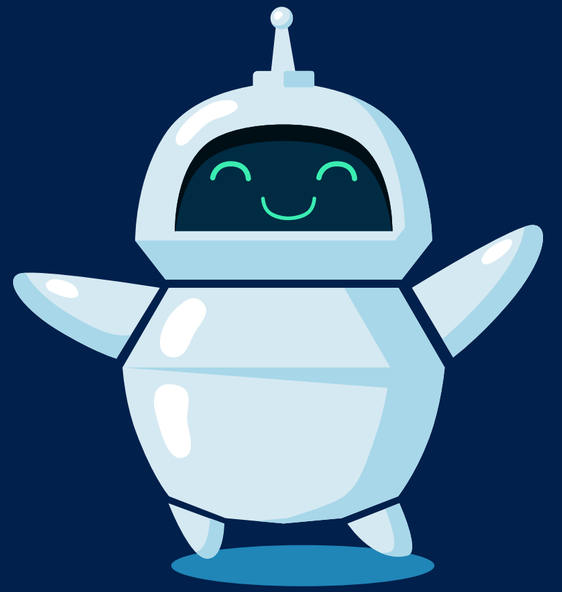
Speed

IP

Dependent on existing representations

NAISS MLB Session 7

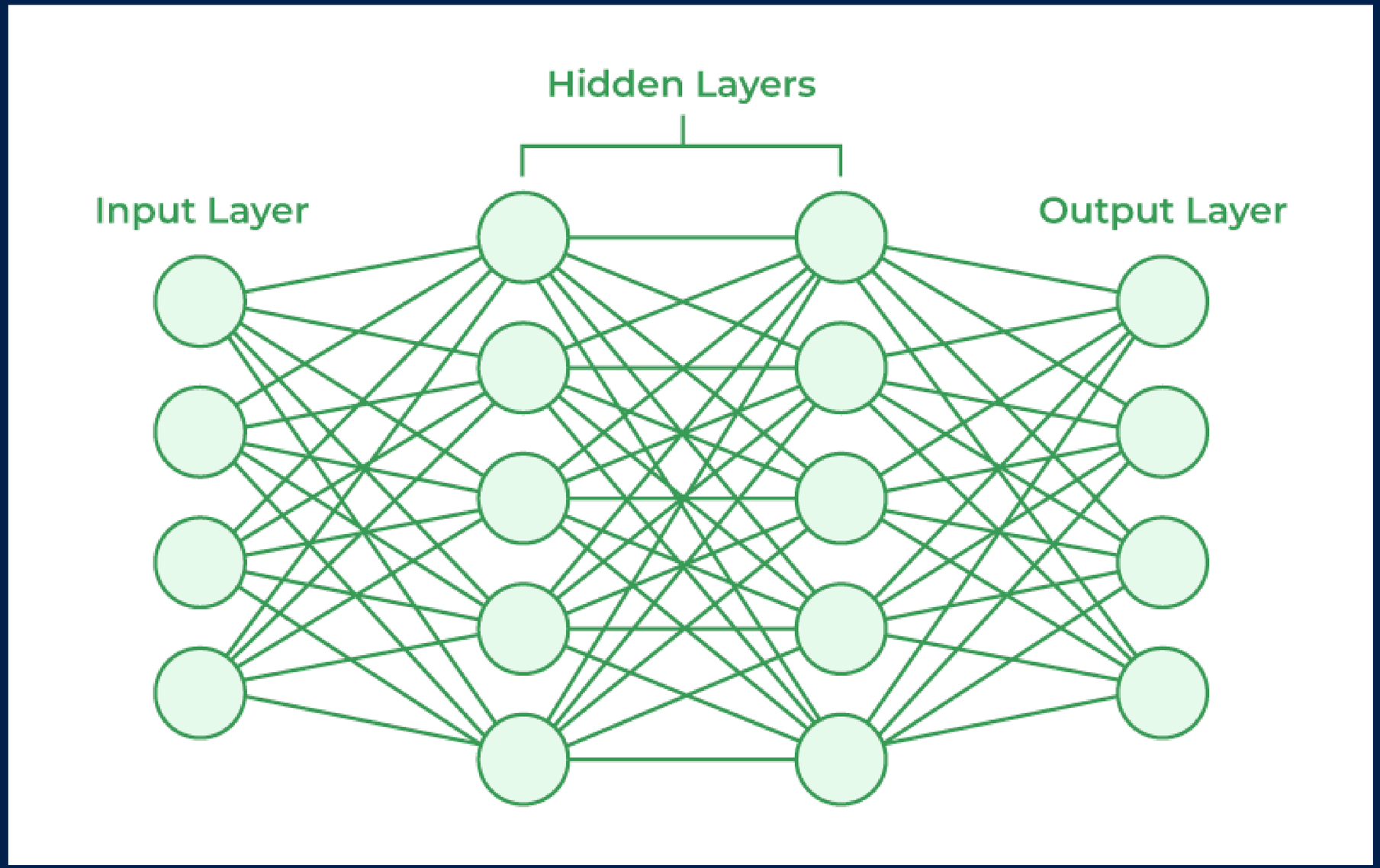
Fine Tuning an LLM



Quick Announcements

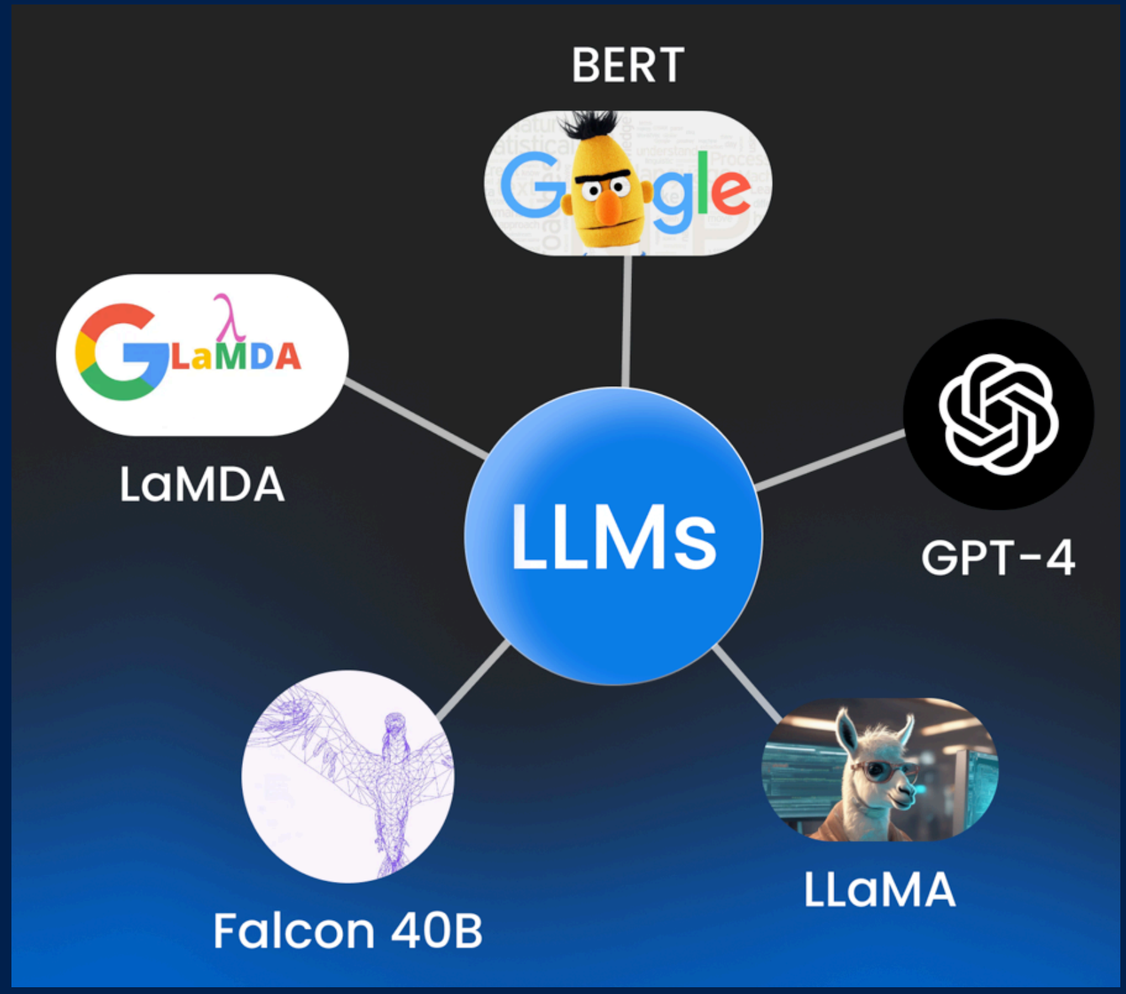
- New form for fine tuning project in discord, old one is fine to submit
- NN projects will be finished evaluated by late this week and early next week

Fine Tuning Recap



What is an LLM

Simply put, an LLMs are built on a specialized type of Neural Network architecture called a Transformer, used mainly to work with language.

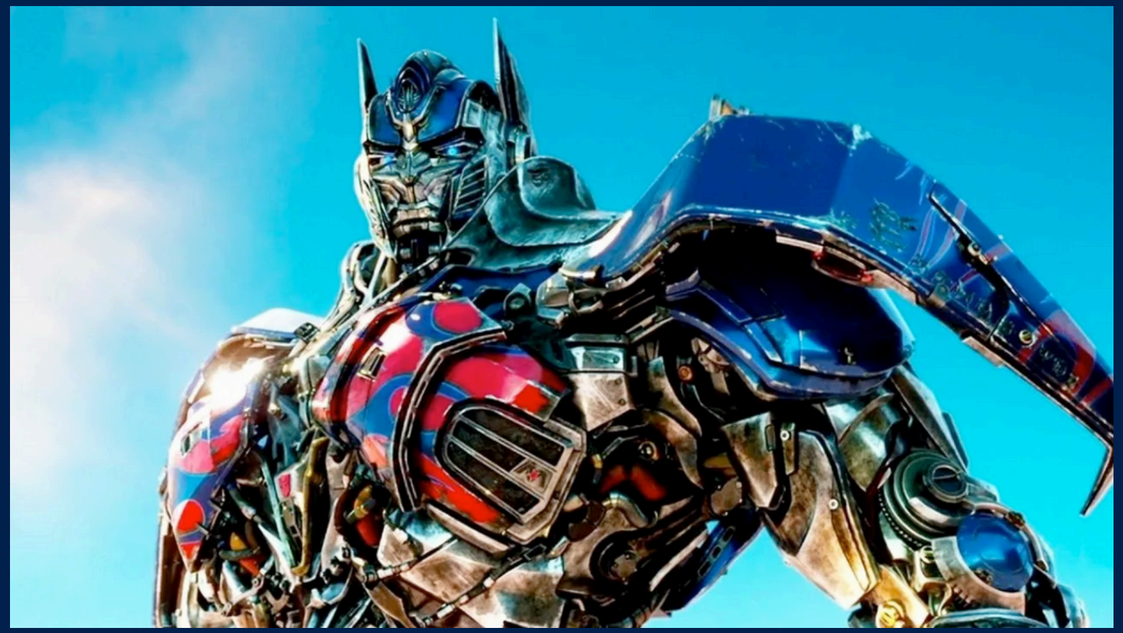


More about BERT

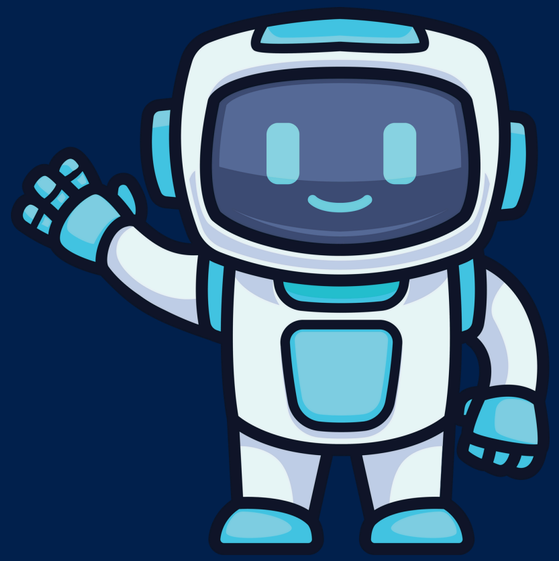
BERT is a LLM developed by Google with a focus on language **understanding** rather than text generation.

Transformers

1. Positional Encoding
    a. “I love my dog"
2. Attention
    a. “The agreement on the European Economic Area was signed in August 1992.”
3. Self Attention (The key)
    a. “Server, can I have the check?”
    b. “Looks like I just crashed the server”

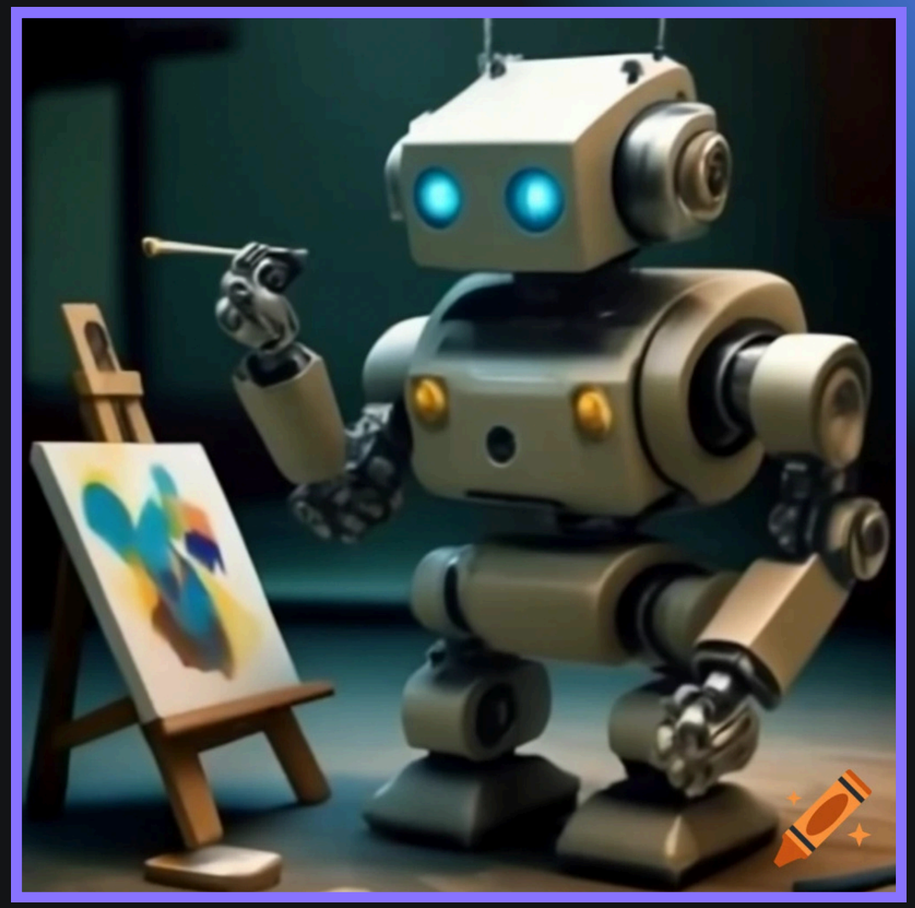


Thank You!



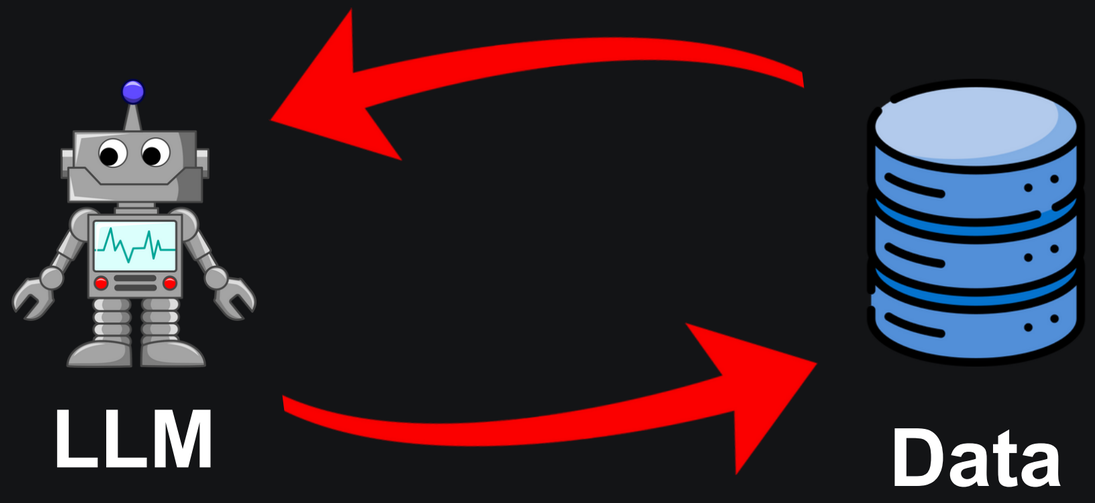
INCREASING LLM PERFORMANCE WITH

RAG



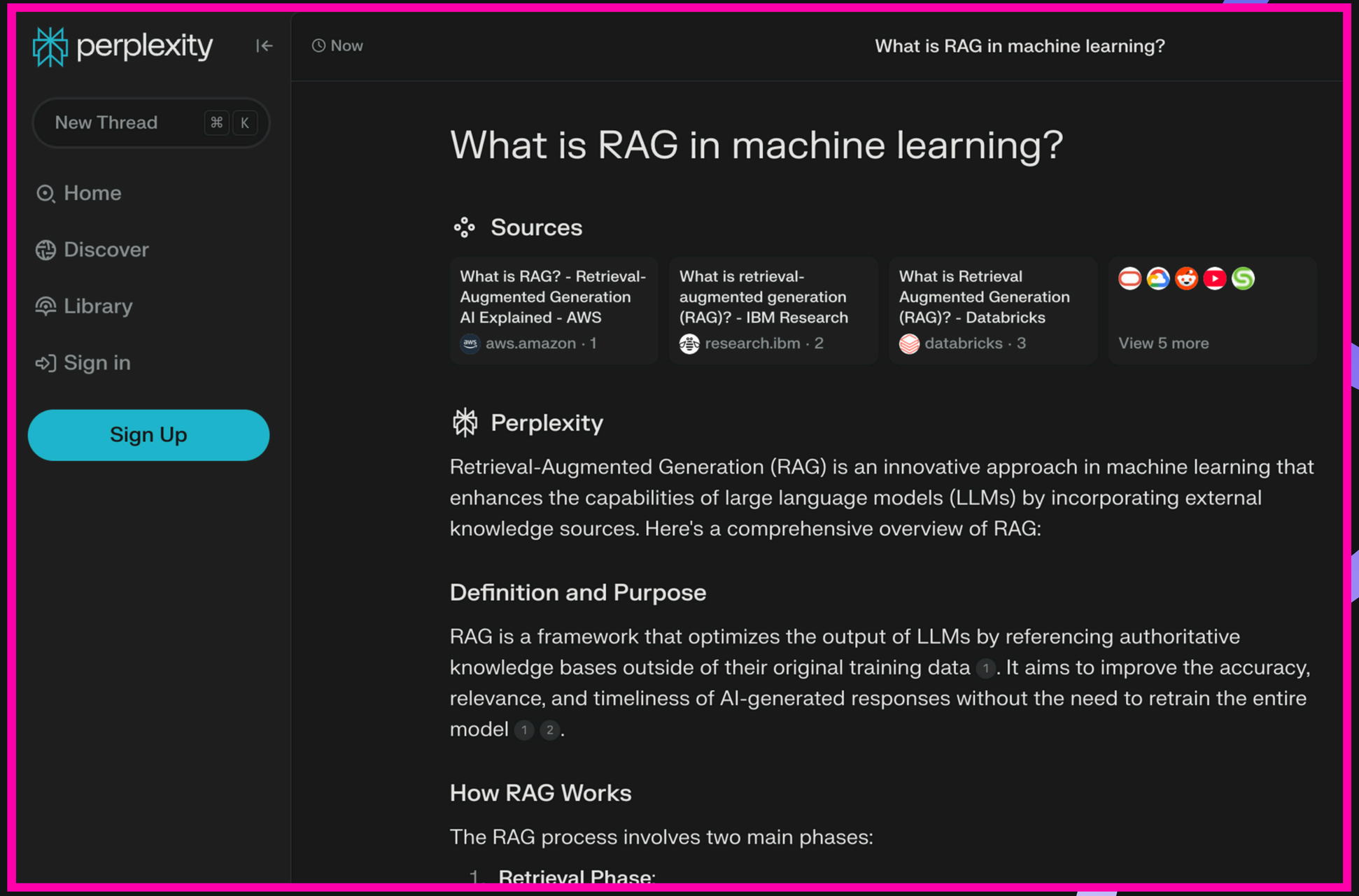
Retrieval Augmented Generation

- Use reference material to inform LLM responses
- Adaptable to growing/changing data
- Don’t increase generation time by too much



Why RAG?

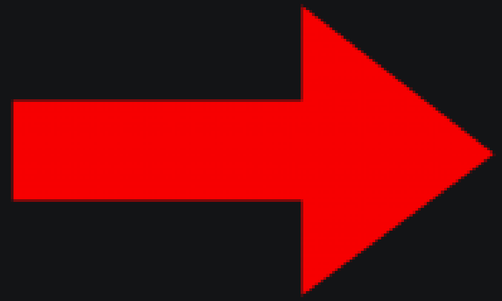
- Specialized Knowledge
- Expert on your specific data
- Cite sources


perplexity
New Thread ⌘ K
Home
Discover
Library
Sign in
Sign Up
Now
What is RAG in machine learning?
What is RAG in machine learning?
Sources
What is RAG? - Retrieval-Augmented Generation AI Explained - AWS
aws.amazon · 1
What is retrieval-augmented generation (RAG)? - IBM Research
research.ibm · 2
What is Retrieval Augmented Generation (RAG)? - Databricks
databricks · 3
View 5 more
Perplexity
Retrieval-Augmented Generation (RAG) is an innovative approach in machine learning that enhances the capabilities of large language models (LLMs) by incorporating external knowledge sources. Here's a comprehensive overview of RAG:
Definition and Purpose
RAG is a framework that optimizes the output of LLMs by referencing authoritative knowledge bases outside of their original training data 1. It aims to improve the accuracy, relevance, and timeliness of AI-generated responses without the need to retrain the entire model 1 2.
How RAG Works
The RAG process involves two main phases:
1. Retrieval Phase:

High view of how it works

High view of how it works

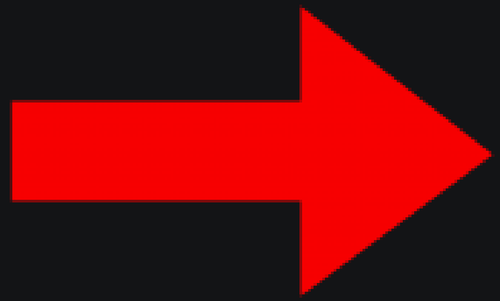
“Who is the greatest basketball player of all time?”



[51, 25.2, 32.5, 76.6]

High view of how it works

"Who is the greatest basketball player of all time?"



[51, 25.2, 32.5, 76.6]

"Jordan is an alright player."

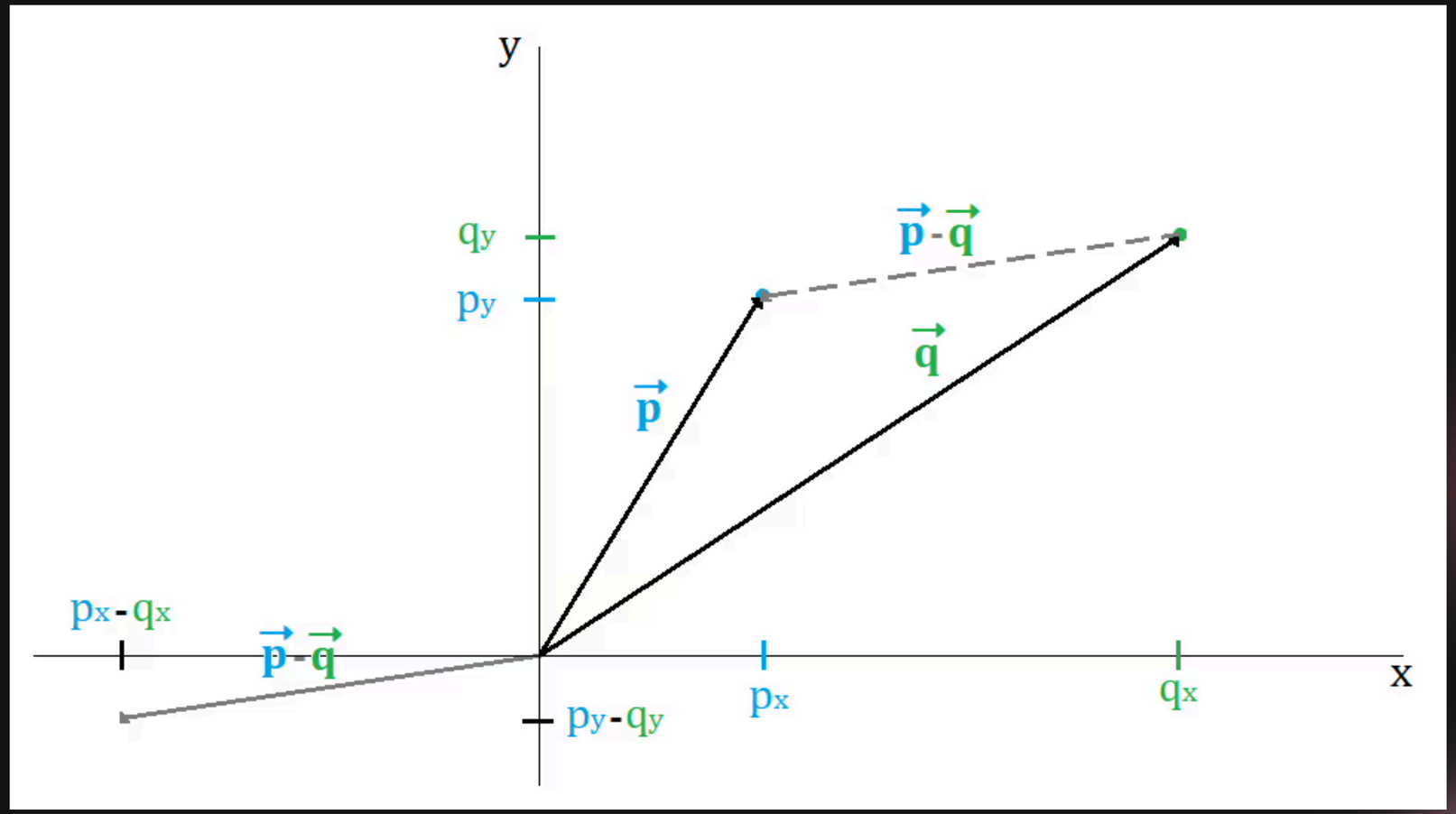
[-21, 14.4, -31, 9.3, 1.2]

"Lebron is the great player of all time."

[9.4, 10, 23, 1.4, 8.2]

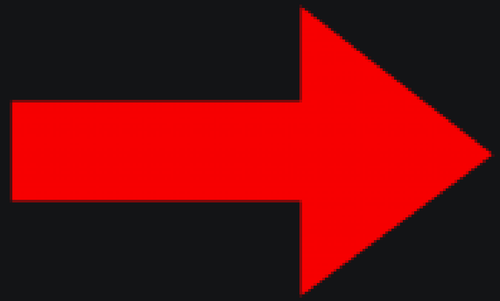
⋮

High view of how it works



High view of how it works

“Who is the greatest basketball player of all time?”



[51, 25.2, 32.5, 76.6]

“Lebron is the great player of all time.”

[9.4, 10, 23, 1.4, 8.2]

High view of how it works

“Who is the greatest basketball player of all time?”

“Lebron is the great player of all time.”

High view of how it works

“Lebron is the greatest basketball player of all time, of course.”

The difference between fine tuning an LLM and RAG

RAG

- Extremely adaptable
- Used when you need more accurate information from LLM

Fine tuning

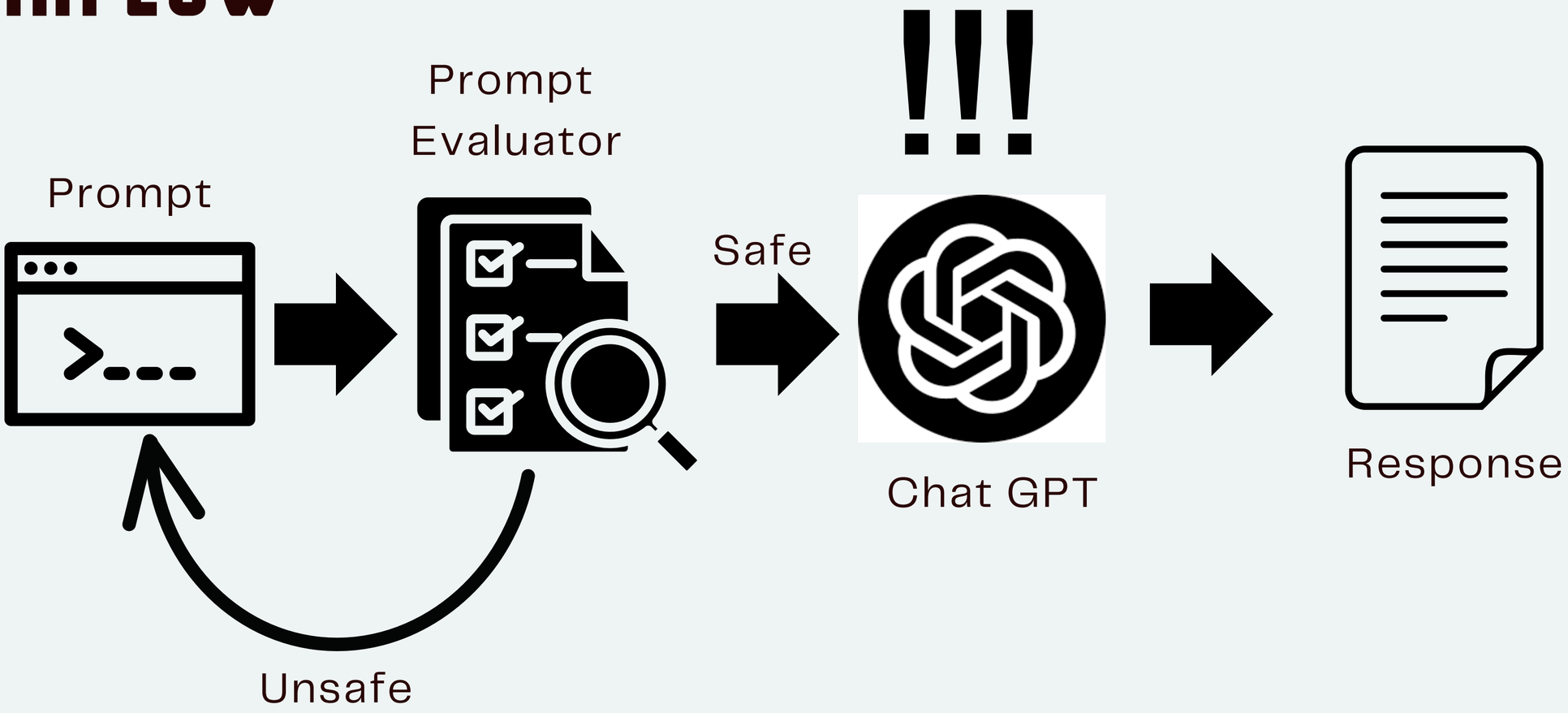
- Not adaptable + expensive
- Used when you want the LLM to “act” in a specific manner

RAG Demo

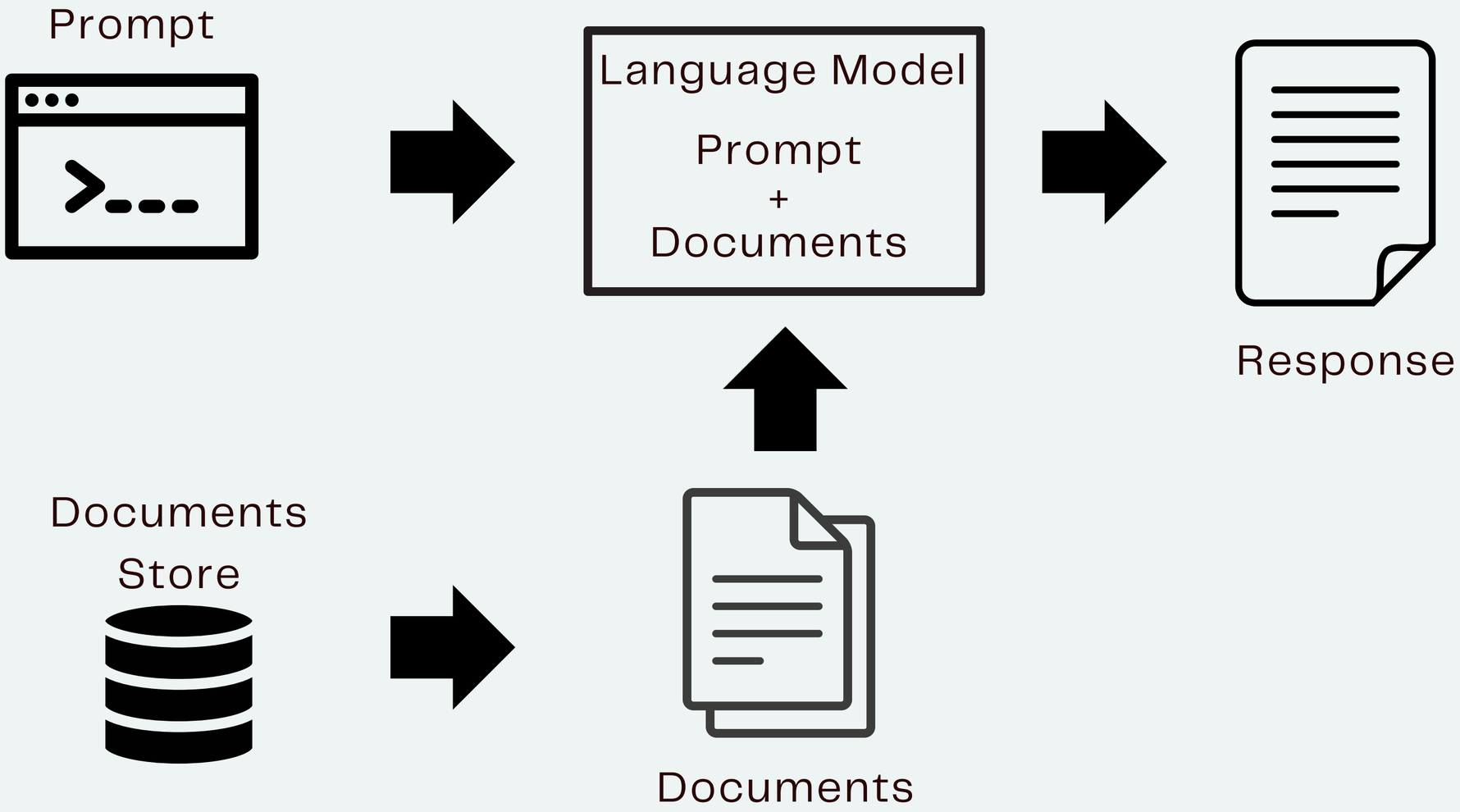
Attendance

RETRIVAL AUGMENTED GENERATION & RAPID DEVELOPMENT

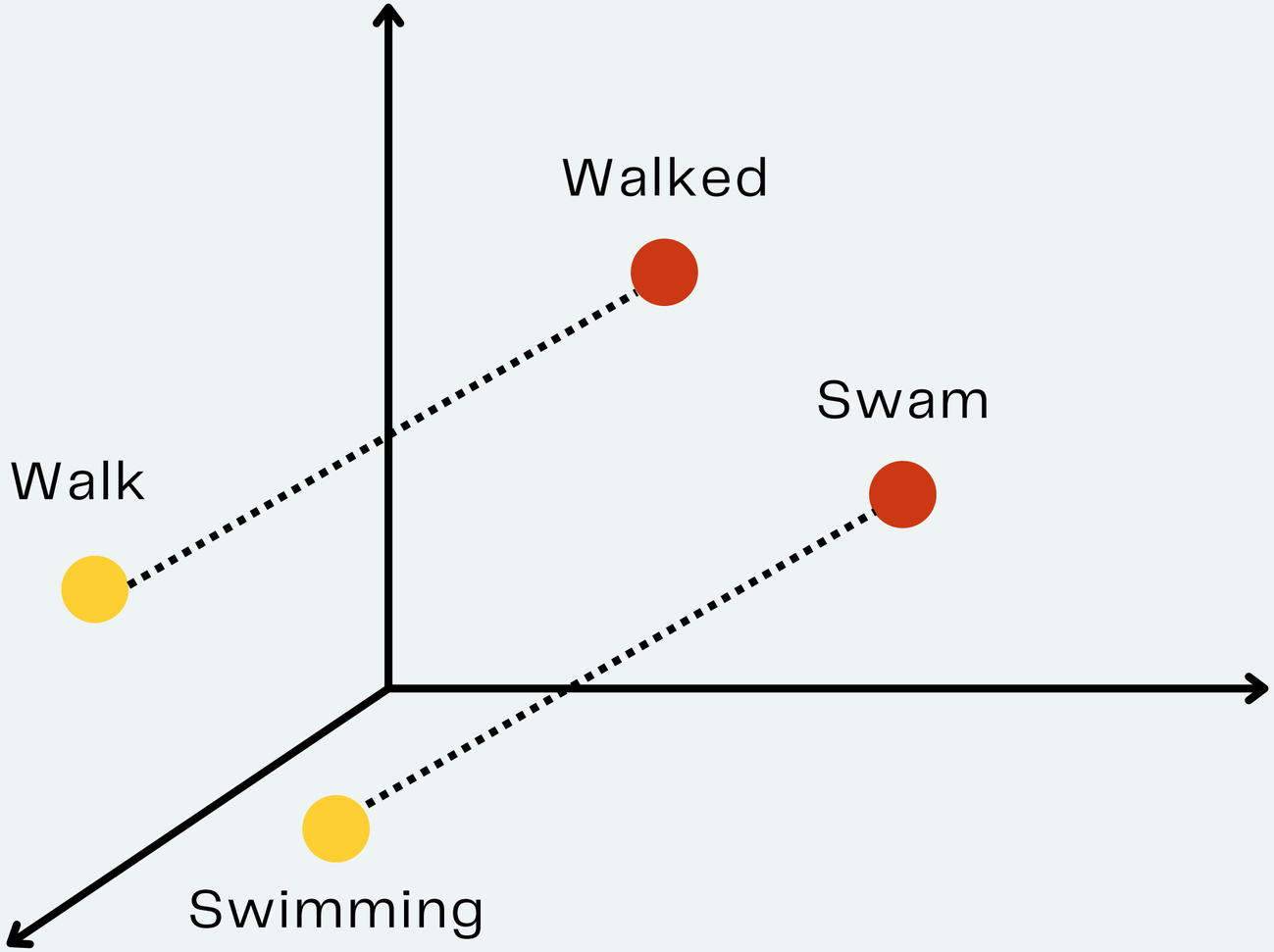
REGULAR CHAT GPT WORKFLOW



WHAT IS R.A.G.?


Prompt
Language Model
Prompt
+
Documents
Response
Documents
Store
Documents

WORD TO VECTOR



WORD TO VECTOR

	Living Being	Feline	Human	Gender	Royalty	Verb	Plural
Cat	0.6	0.9	0.1	0.4	-0.7	-0.3	-0.2
Kitten	05	0.8	-0.1	0.2	-0.6	-0.5	-0.1
Dog	0.7	-0.1	0.4	0.3	-0.4.	-0.1	-0.3
Houses	-0.8	-0.4	-0.5	0.1	-0.9	0.3	0.8