Data Science

Artificial Intelligence

Tinglong Dai, PhD

Bernard T. Ferrari Professor

Johns Hopkins Carey Business School

Agenda

1 — **Week 1: Foundations**

2 — **Week 2: Tools & Techniques**

Machine learning, deep learning, Keras, Python, and no-code tools.

3 — **Week 3: CNNs & Business Models**

Convolutional neural networks, using AI to create human value Assignment #1 due.

4 — **Week 4: MLOps & Evaluation**

MLOps, transformers, evaluation of AI. Lab proposal due.

5 — **Week 5: LLMs & GenAI**

Large language models, generative AI, using AI to create business value. Assignment #2 due.

6 — **Week 6: Human-AI Interaction**

Human-AI interaction, reinforcement learning, system design.

7 — **Week 7: Responsible AI**

Responsible AI, future of AI. Lab final deliverable due.

8 — **Week 8: Individual Final Exam; Peer Reviews of AI Lab**

Python Basics: First TA Tutorial

🗓️ **Friday, 10/31, 12:00–1:00 PM**

📍 **Join via Zoom:** https://bit.ly/jhuaita25

Attendance is optional. Materials will be posted on Canvas

No-Code AI Development Tools

Sean Cusack, a beekeeper, was curious if anything other than bees was entering his hives. So he made a tiny photo booth that snapped pictures whenever something moved around it. However, sorting through thousands of insect portraits proved time-consuming... So he used Lobe to create an AI solution.

"It was just really simple," Mr. Cusack said, adding that the underlying data science was "over my head.." The Lobe platform allowed him to drag and drop sample photos and click a few buttons to make a system that could recognize his beloved bees and spot unwelcome visitors.

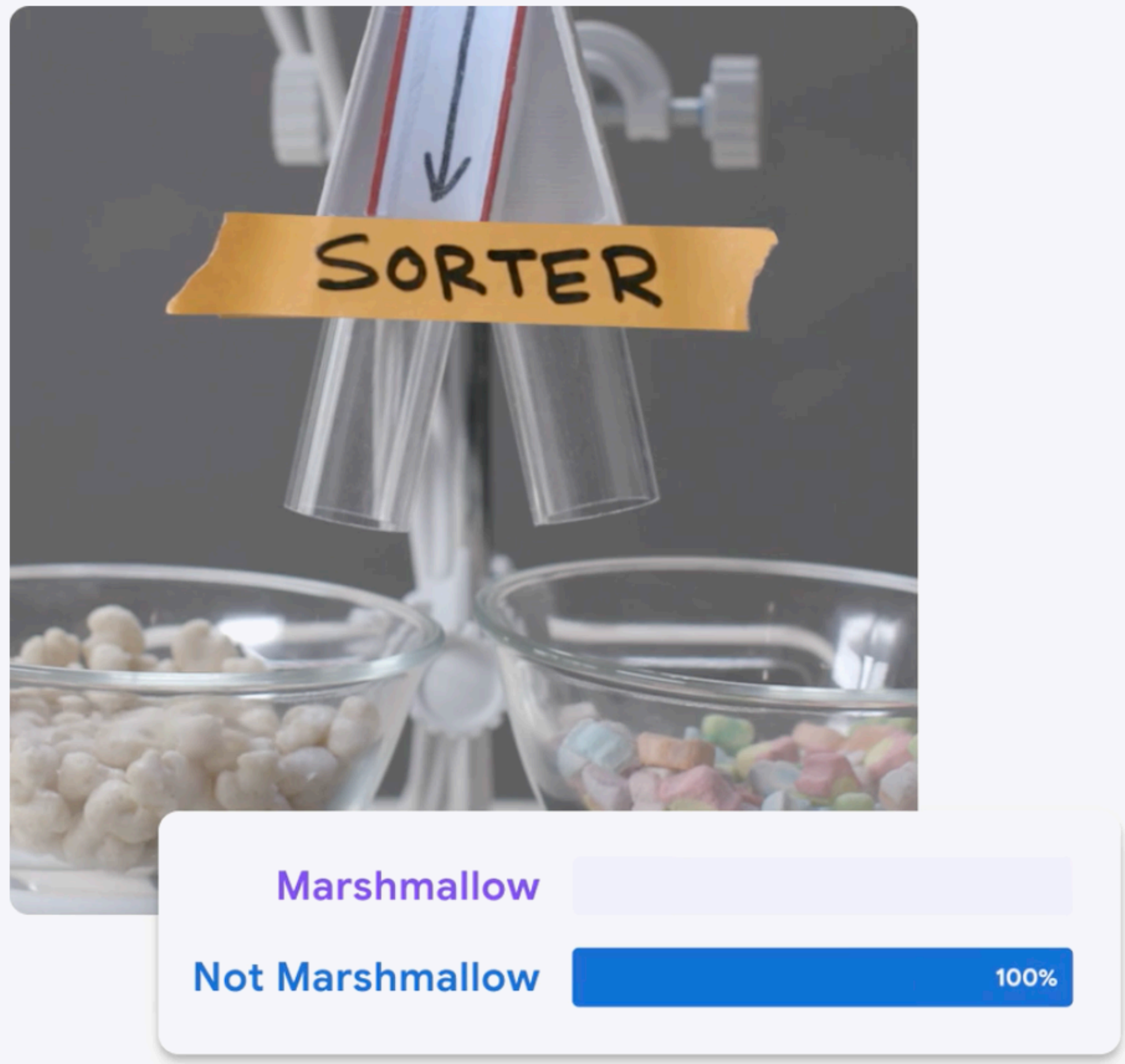
Smith, C. S. (2022) "'No-code' brings the power of A.I. to the masses," *New York Times* (Mar. 15). https://nyti.ms/3p78s2g

Teachable Machine

Train a computer to recognize your own images, sounds, & poses.

A fast, easy way to create machine learning models for your sites, apps, and more – no expertise or coding required.

Get Started



https://teachablemachine.withgoogle.com

How to Build AI That Actually Works for Your Business

Mims, C. (2022) *Wall Street Journal* (July 23)

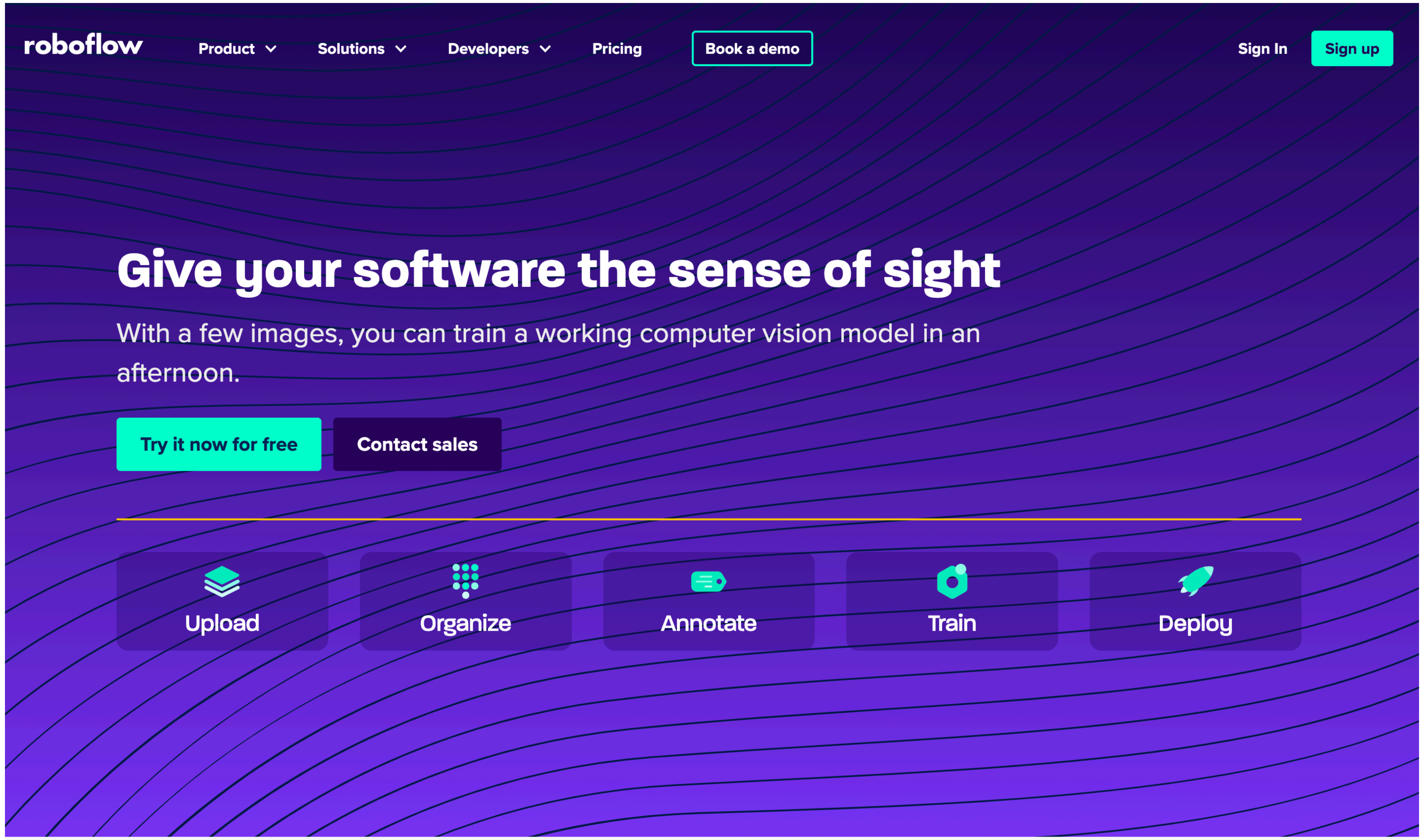
- Stories from the article — Can anyone explain these stories?

  - **PreciTaste** by Ingo Stork: Using real-time sensor data and AI algorithms to predict how much food people will order at any given moment, and prepare just enough food

  - **Phuc Labs** by Phuc Vinh Truong: Using AI to make recycling electronic waste more efficient

  - **Gong** by Amit Bendov: Capturing and analyzing multi-channel communications used by a sales team. It then makes suggestions to help salespeople close more deals.

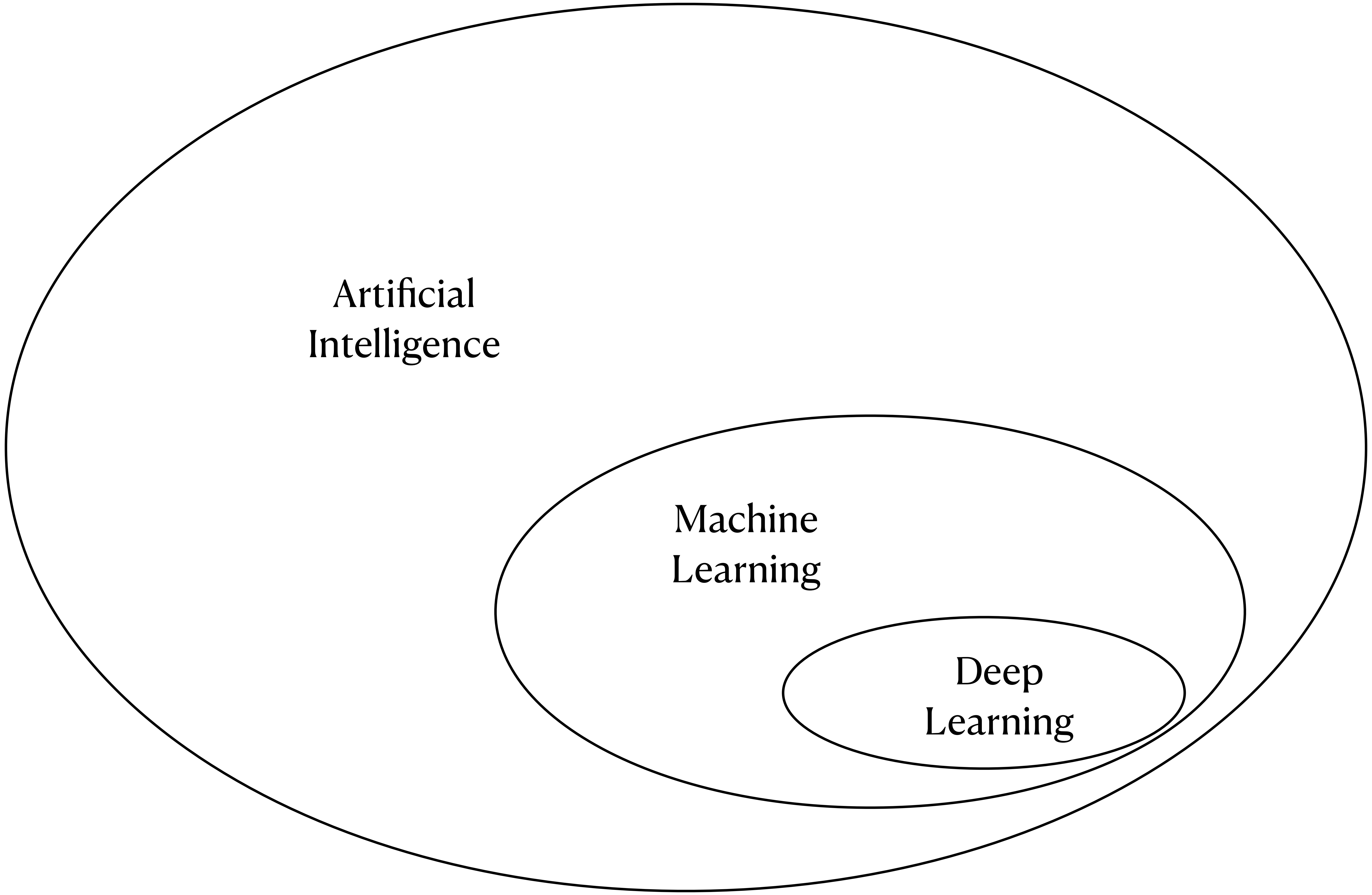
Roboflow: Commercial-Grade, No-Code AI Development



https://roboflow.com/

https://roboflow.com/

Machine Learning Basics

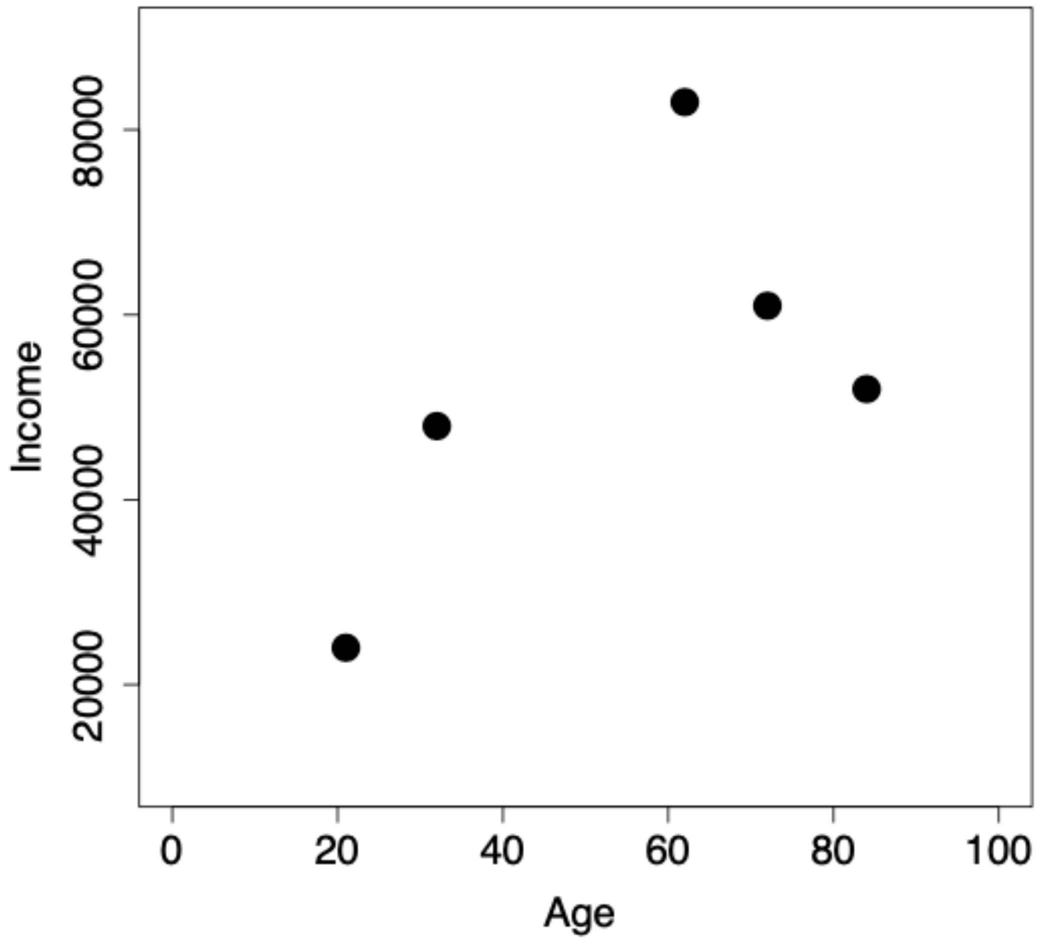


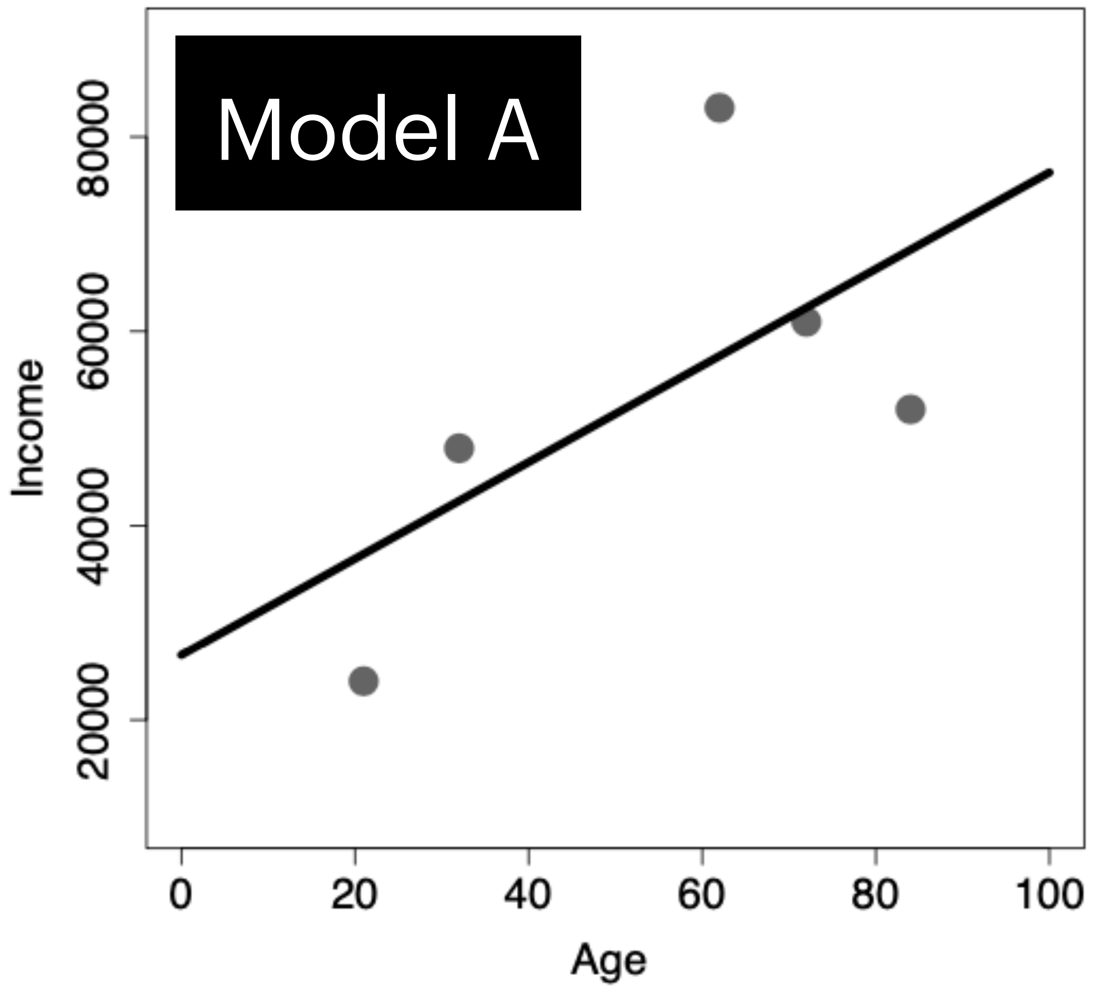
What’s the difference between deep learning and “shallow learning”?

Deep vs. "Shallow" Learning

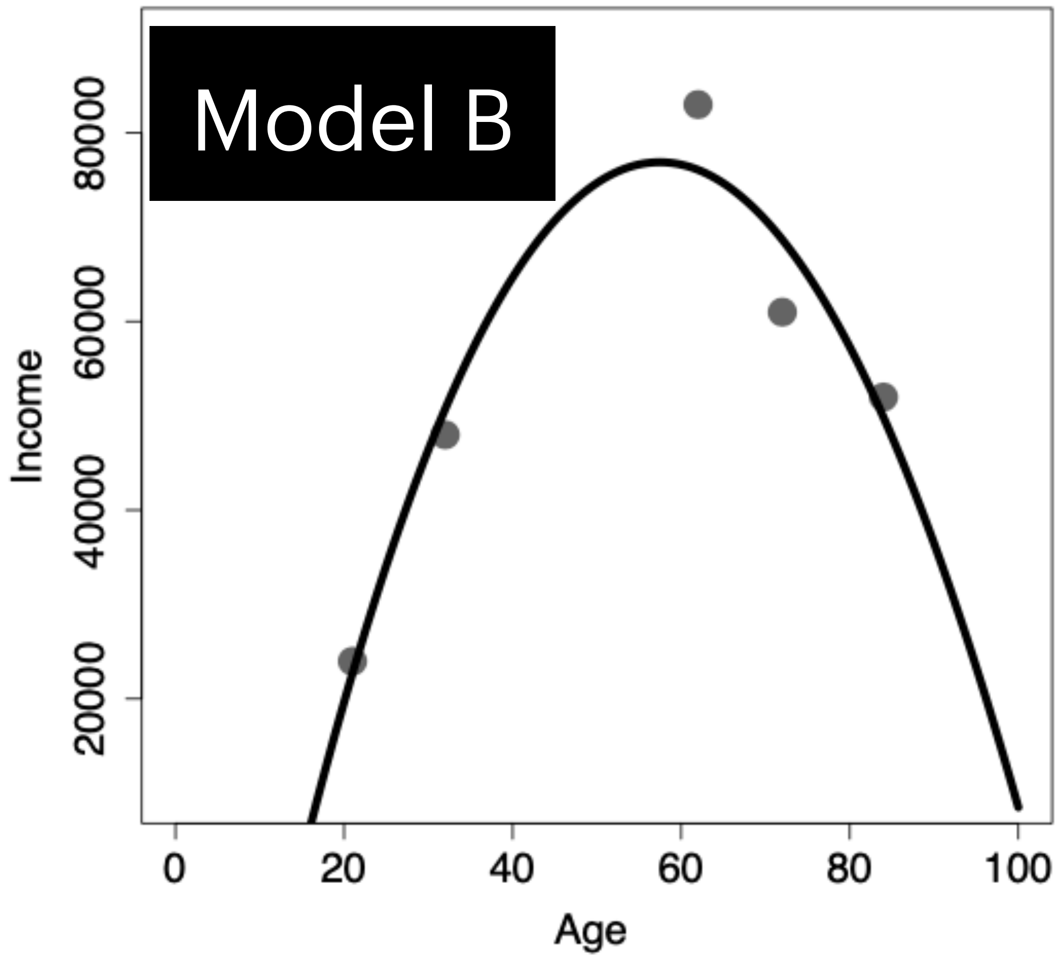
- Deep learning completely automates the most crucial step in machine learning: *feature engineering*
- Prior to deep learning, machine learning transforms the input data into one or two successive representative spaces, via, e.g.,
  - Linear regression
  - Support vector machine (SVM)
  - Decision trees
- But machine learning is known to be an "ill-posed problem" that requires significant human effort to make sense of

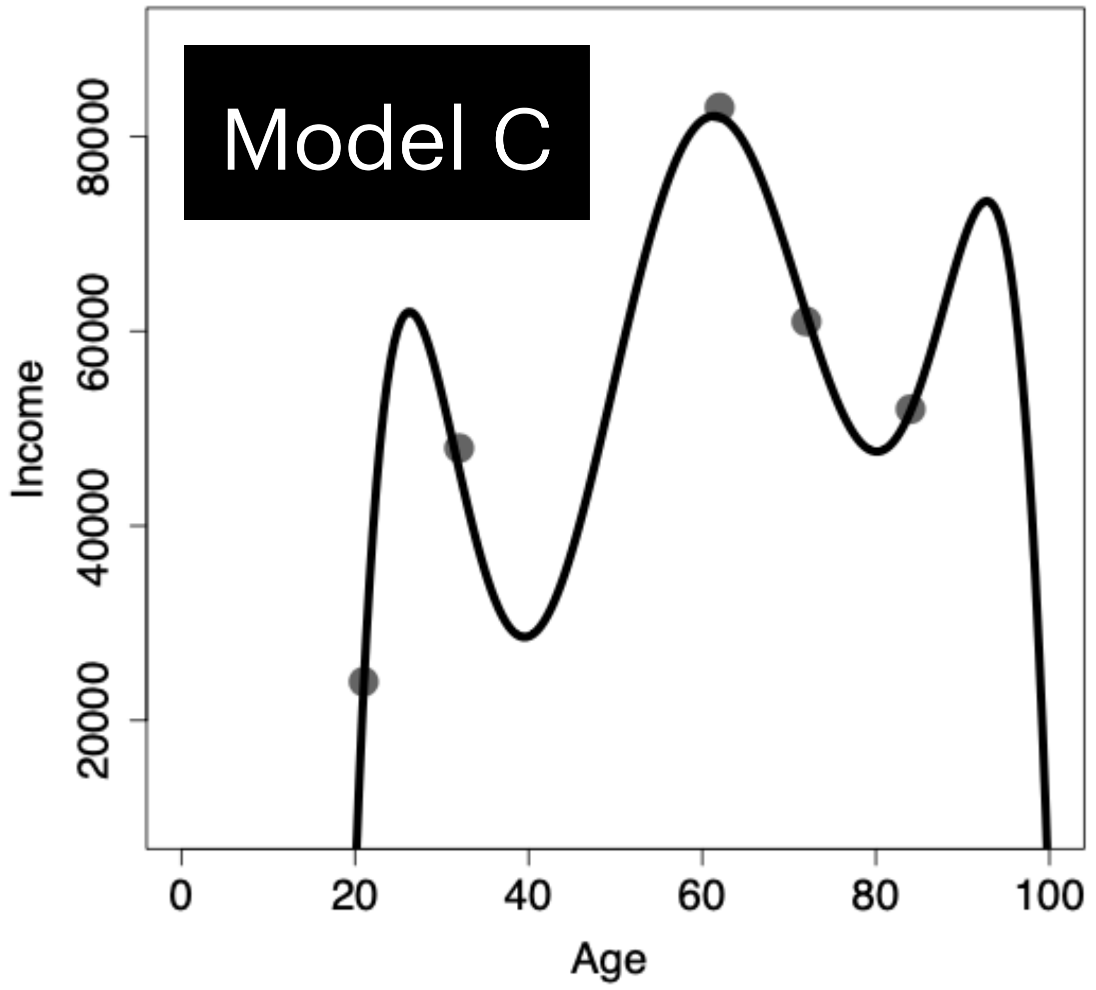
Machine Learning, "An Ill-Posed Problem"





Which model is the best?





Kelleher et al. (2015)

A Fundamental Tension in ML: Overfitting vs. Underfitting

Optimization vs. Generalization

- Machine learning has two tasks:
  - Optimization: adjust the model to best fit the training data
  - Generalization: ensure the model performs well on data it has *never* seen before

- Example: What's the best way to prepare for the U.S. medical licensing exam?
  - Memorizing answers to the sample tests ≠ doing well in the real test

- **Overfitting** occurs when the algorithm has learned too much from the training data

- **Underfitting** occurs when the algorithm hasn't modeled all relevant patterns in the training data

Overfitting?

Friends S3.E8 “The One with the Giant Poking Device”

The Washington Post

Is Taylor Swift’s Eras Tour a curse on NBA teams?

A Redditor’s theory says yes

By María Luisa Paúl

June 5, 2023 at 6:23 a.m. EDT

Weeks ago, 24-year-old Matt Moses noticed a trend: Whenever Taylor Swift’s Eras Tour came to an NBA team’s city during the playoffs, that team was doomed to lose.

Those three Atlanta shows in late April? Boom, the Hawks lost their series to the Celtics just before Swift arrived. When Swift played in Philadelphia in May? The 76ers lost the series, also to the Celtics, after a Game 7 defeat on Swift’s final night at Lincoln Financial Field.

Example: Using Decision Trees to Predict Loan Outcome

ID	Occupation	Age	Loan-Salary Ratio	Outcome
1	industrial	39	3.40	default
2	industrial	22	4.02	default
3	professional	30	2.7 0	repay
4	professional	27	3.32	default
5	professional	40	2.04	repay
6	professional	50	6.95	default
7	industrial	27	3.00	repay
8	industrial	33	2.60	repay
9	industrial	30	4.5 0	default
10	professional	45	2.78	repay

Model 1

if Loan-Salary Ratio > 3.00 **then**
  Outcome = *default*
else
  Outcome = *repay*

Model 2

if Age= 50 **then**
  Outcome = *default*
else if Age= 39 **then**
  Outcome = *default*
else if Age= 30 **and** Occupation = *industrial* **then**
  Outcome = *default*
else if Age= 27 **and** Occupation = *professional* **then**
  Outcome = *default*
else
  Outcome = *repay*

Which model is better?

Kelleher et al. (2015)

Example: Predicting Loan Outcome

Which model is better?

Model 1

if LOAN-SALARY RATIO > 3.00 **then**
  OUTCOME = *default*
else
  OUTCOME = *repay*

Model 2

if AGE= 50 **then**
  OUTCOME = *default*
else if AGE= 39 **then**
  OUTCOME = *default*
else if AGE= 30 **and** OCCUPATION = *industrial* **then**
  OUTCOME = *default*
else if AGE= 27 **and** OCCUPATION = *professional* **then**
  OUTCOME = *default*
else
  OUTCOME = *repay*

- Model 2 is overfitting because each of the decision rules specified in Model 2 is specific to a single instance. It lacks generalizability
- Model 1 appears to generalize beyond the data

Kelleher et al. (2015)

Feature Leakage

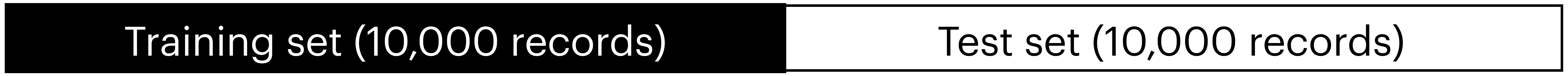
- **Feature leakage** occurs when some of the features specified in the machine learning model leaks information and allows the system to "cheat"

- Example: Cancer prediction (Kaufman et al. 2011)
  - Objective: Predicting the probability that a patient has cancer
  - Features: patient age, gender, medical history, hospital name, vital signs, tests
  - The machine learning model has excellent performance on test data
  - Yet real-world performance was surprisingly poor. What happened?

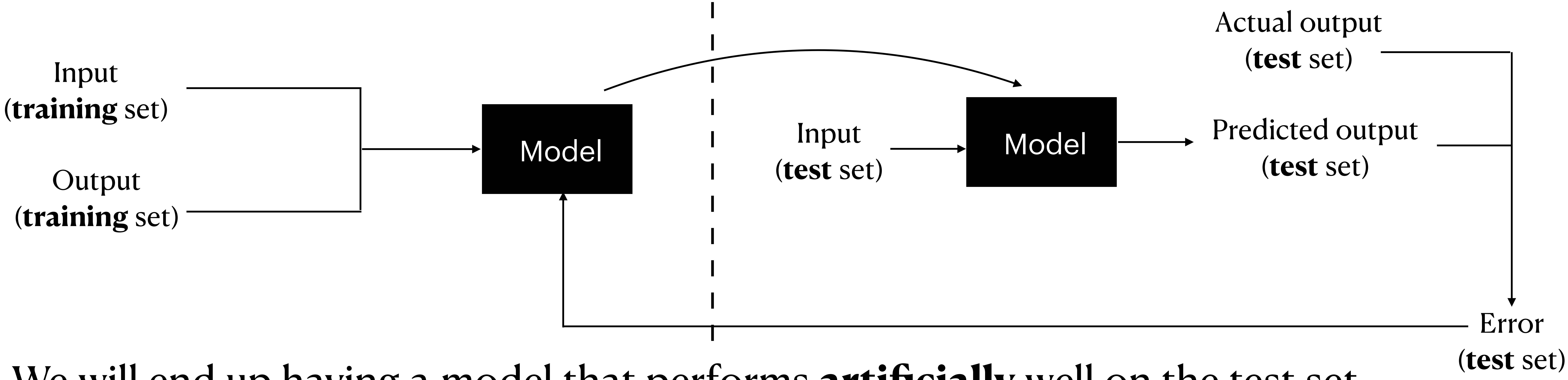
"Hospital name" leaks the information about cancer diagnosis. So the model doesn't work for patients whose hospitals have yet to be assigned.

Training Leakage

- Imagine when you try to build a diagnosis model using 20,000 patient records
- Let's divide patient records into two parts:



- What could possibly go wrong?



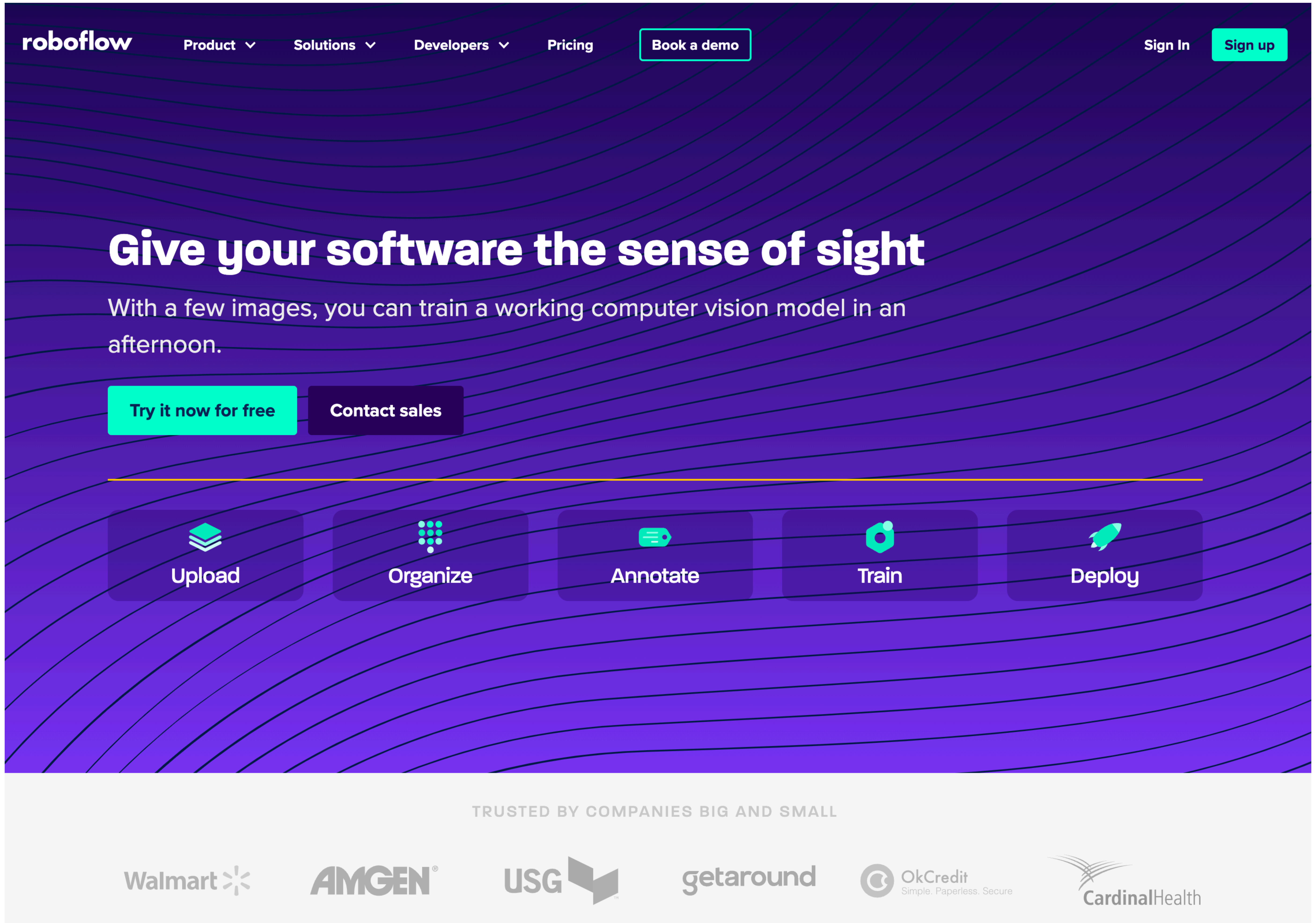
- We will end up having a model that performs **artificially** well on the test set

Solution to Training Leakage

- Principle: To avoid leakage, our model shouldn't have had access to *any* information about the test set, either directly or indirectly
- Solution: Split data into *three* parts: training, **validation**, and test sets
    - The test set is also known as "holdout set" and should never be used to train the model
- Common options: splitting datasets into 6/2/2, 7/2/1, or 8/1/1
- We will discuss more sophisticated strategies in MLOps (session 4)

No-code AI Example: Roboflow

Training, Validation, and Testing Sets



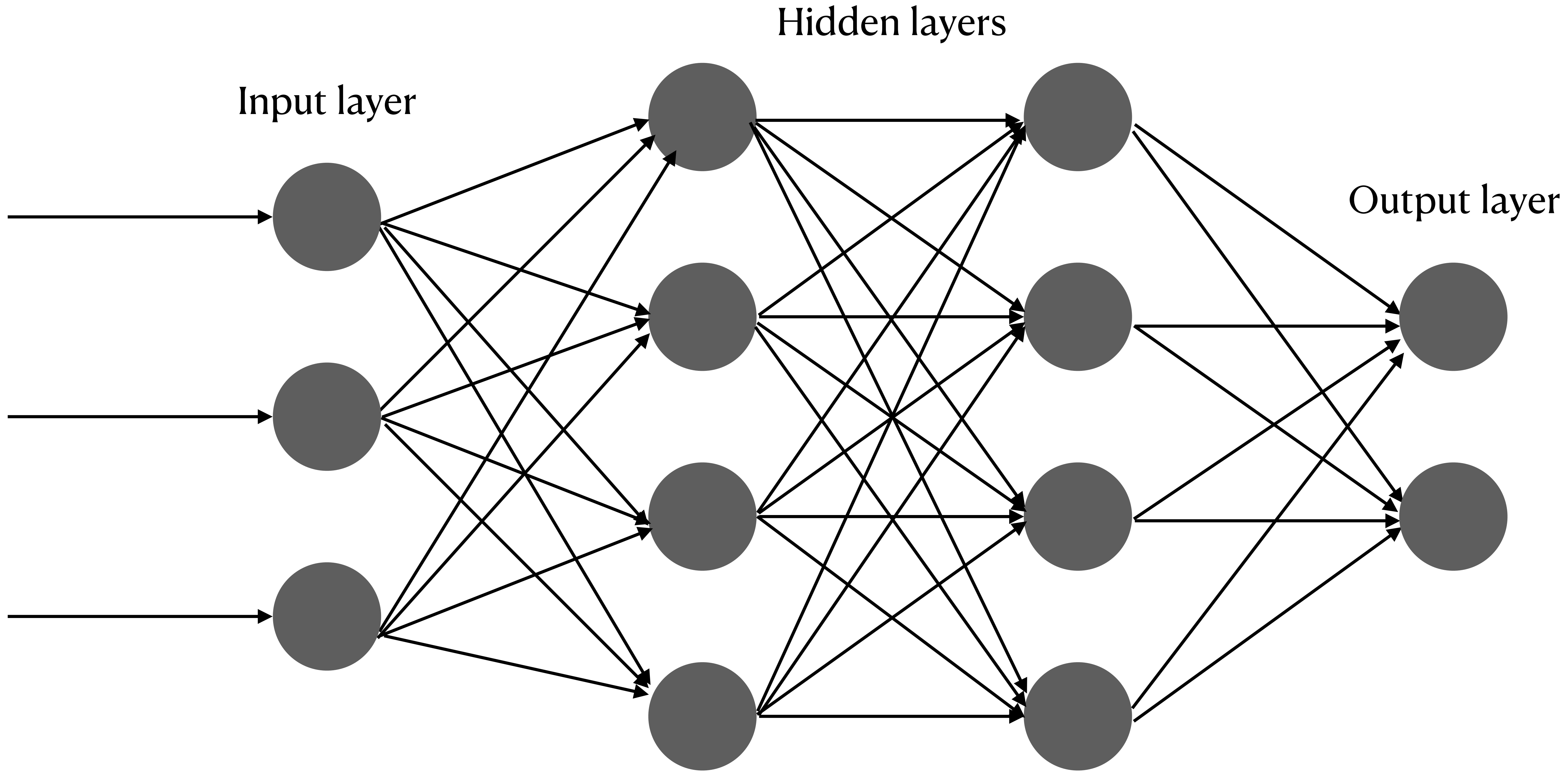
https://roboflow.com/

Deep Learning

Artificial Neural Networks & Convolutional Neural Networks

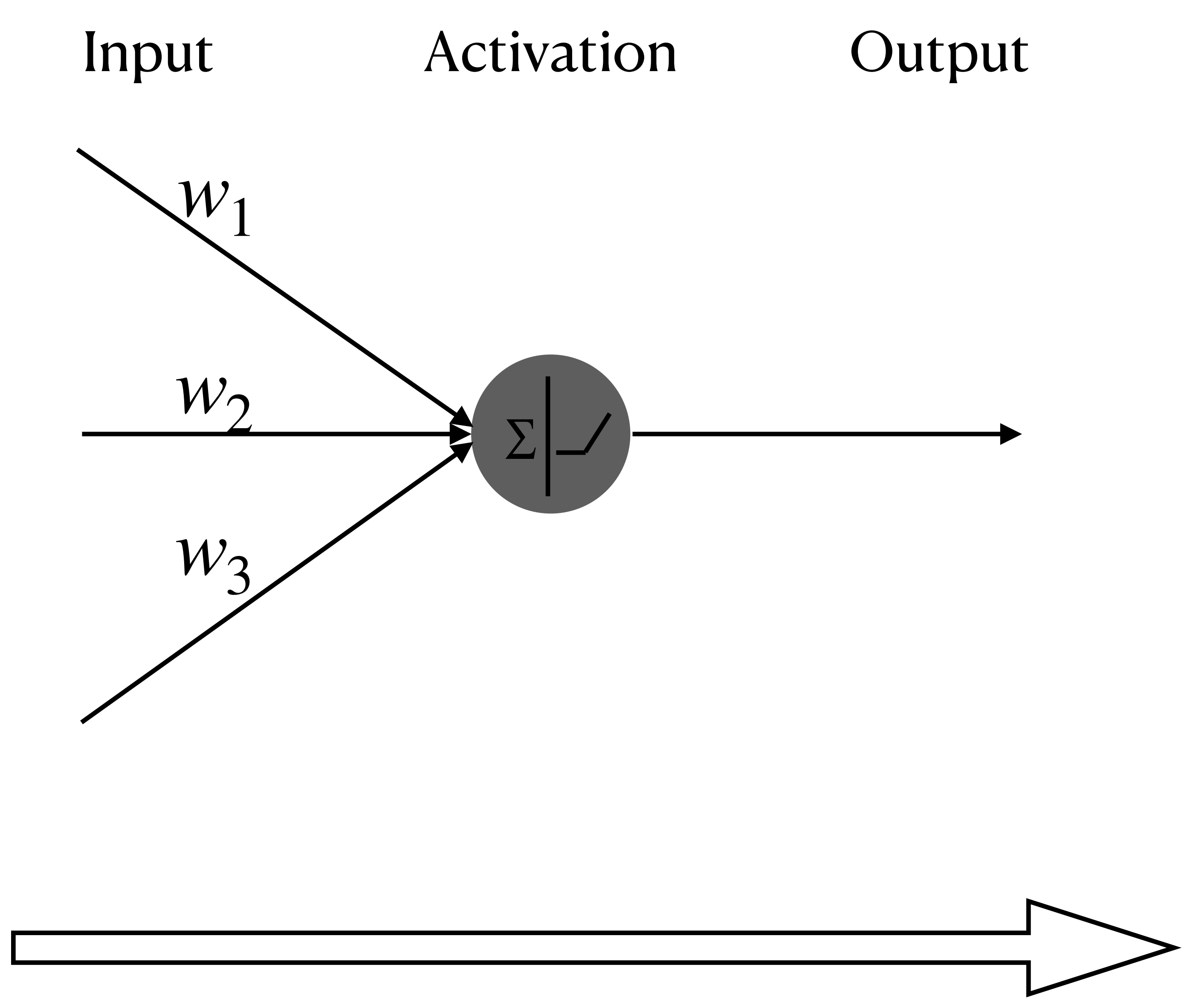
Deep Learning

- Deep learning is an incremental, layer-by-layer learning approach in which increasingly complex representations are developed to map the input to output

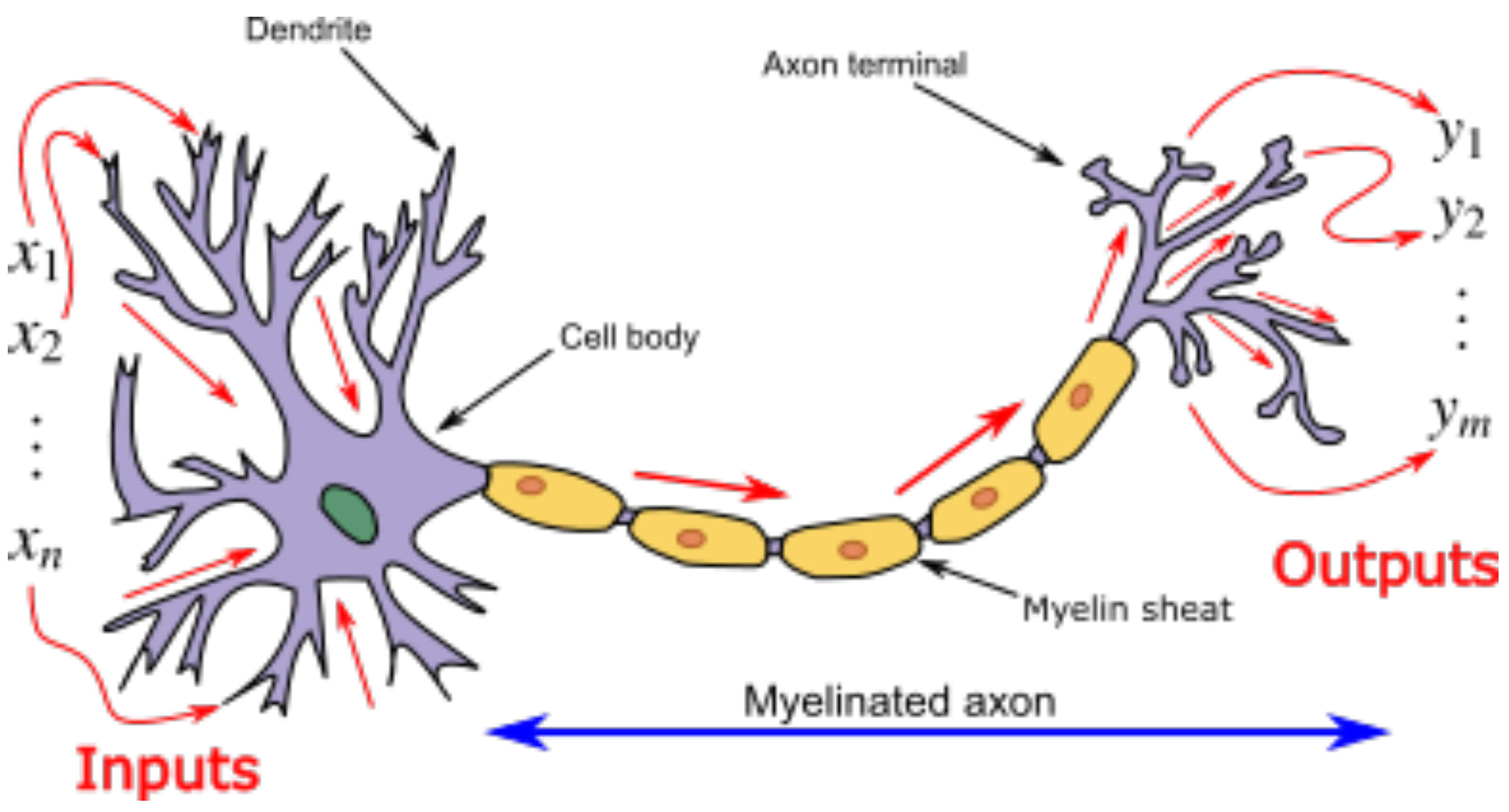


Artificial Neural Network

Artificial Neurons vs. Neurons

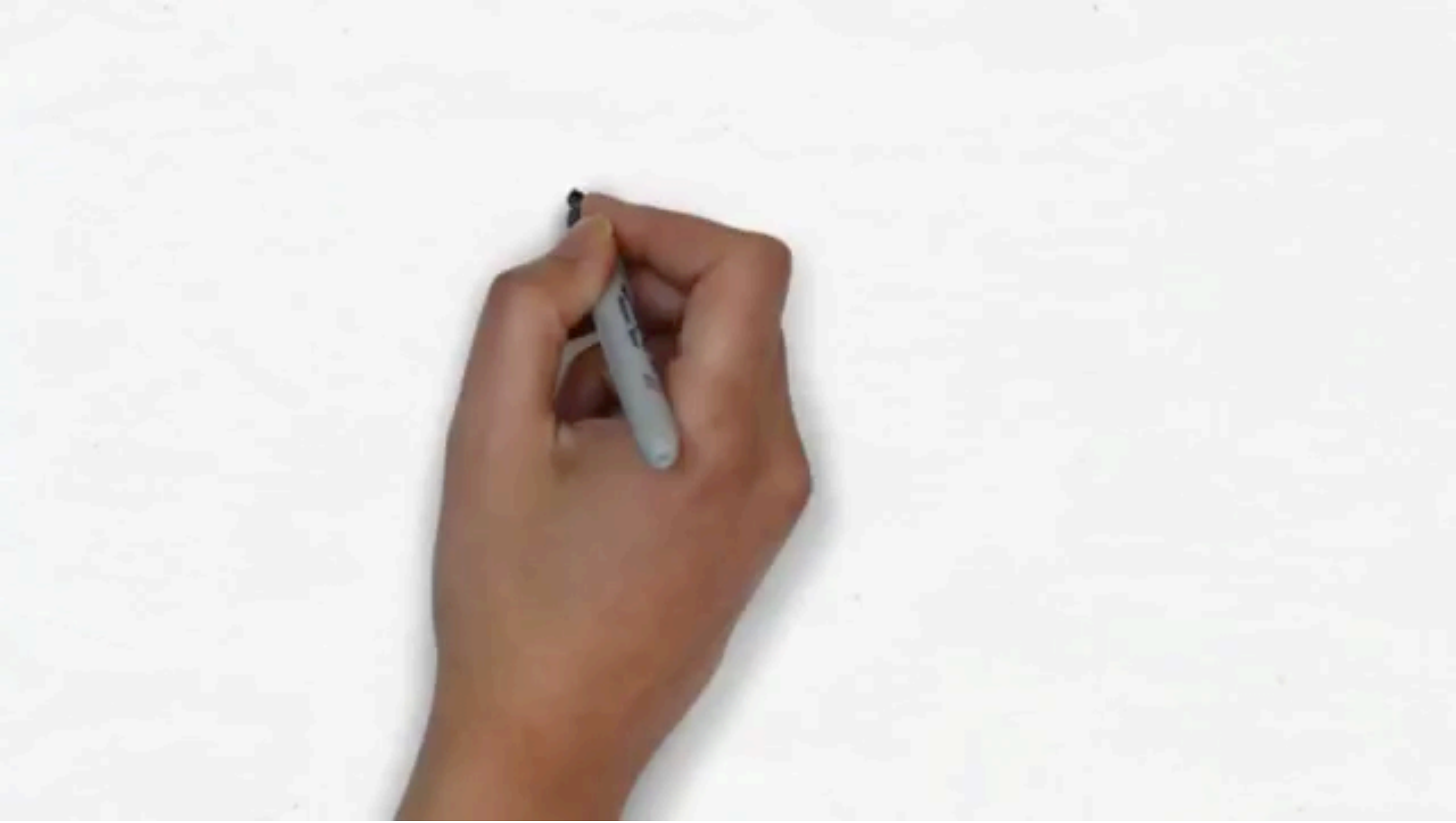


Artificial Neuron



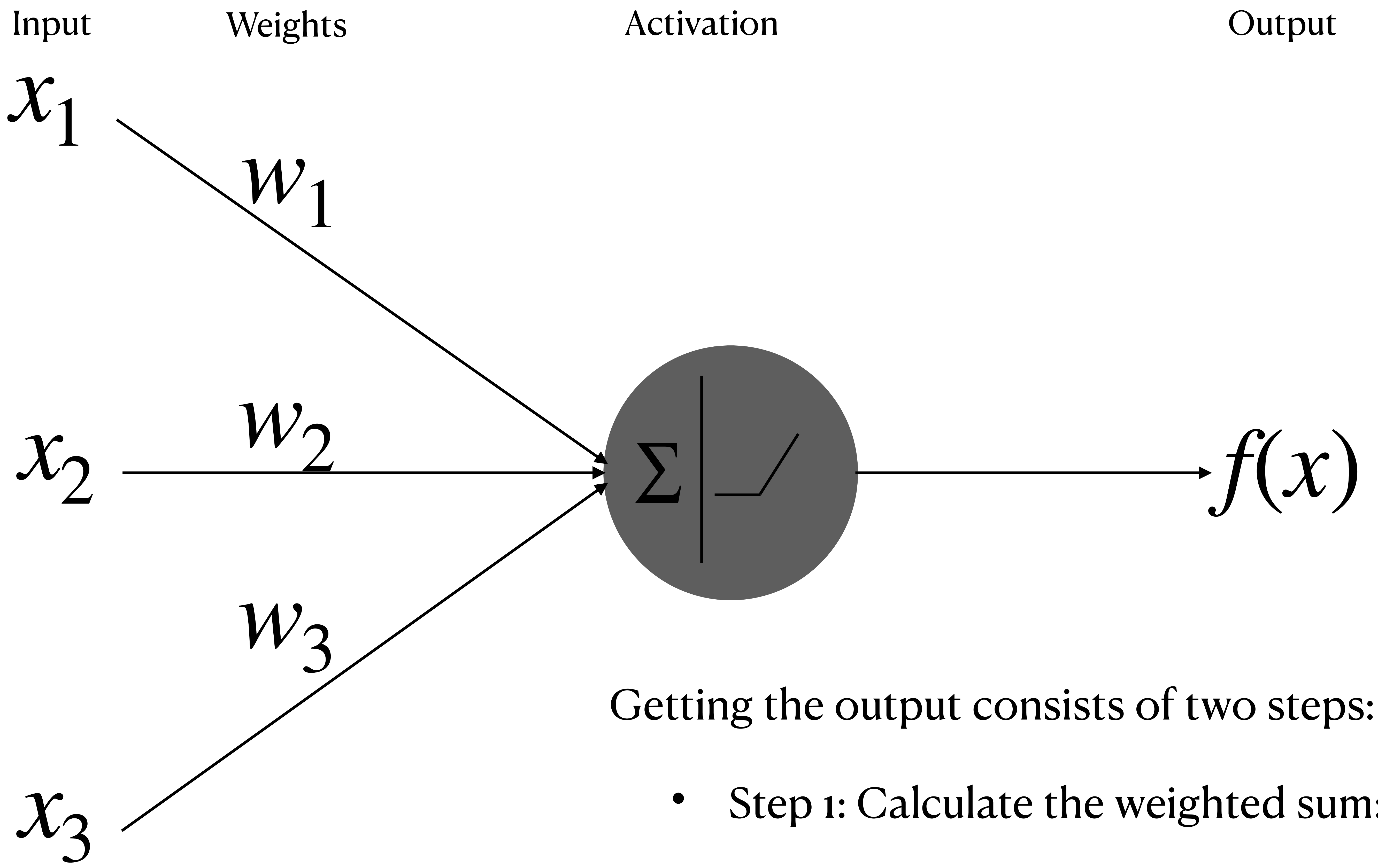
Neuron and Myelinated Axon

with signal flow from inputs at dendrites to outputs at axon terminals



Key insight: Part of what makes us intelligent is our capacity to filter information before processing it. Not everything will trigger the activation of neurons in our brains.

How Feedforward Neural Networks Work



Getting the output consists of two steps:

- Step 1: Calculate the weighted sum: $w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3$
- Step 2: Applying the activation function a:
  $f(x) = a(w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3)$

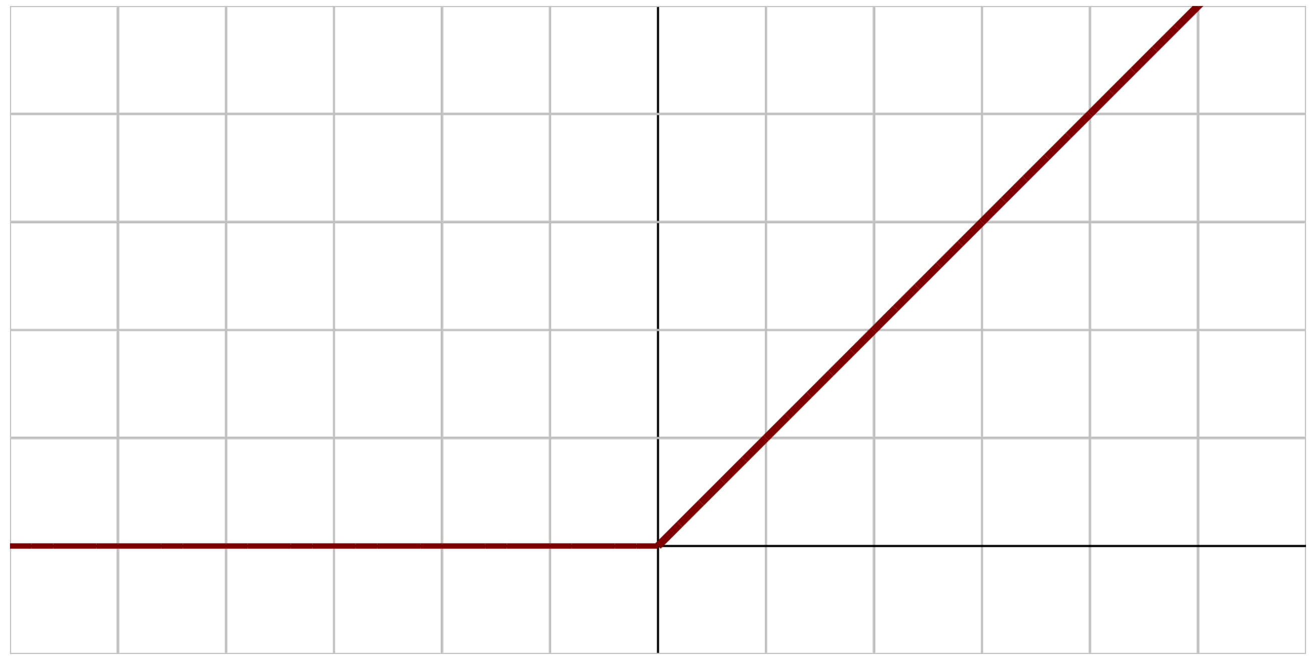
Activation Functions

ReLU

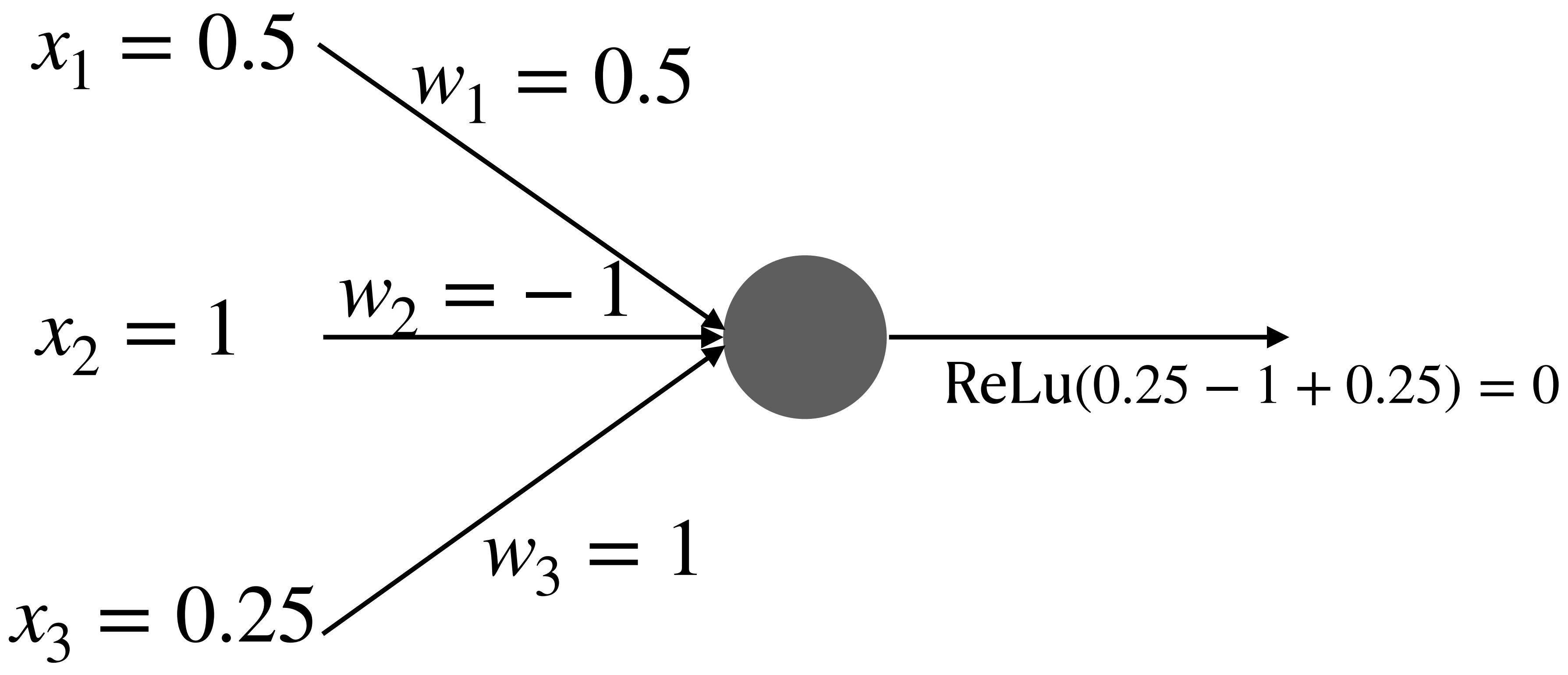
- The most widely used activation function is the **ReLU** (Rectified Linear Unit) function:

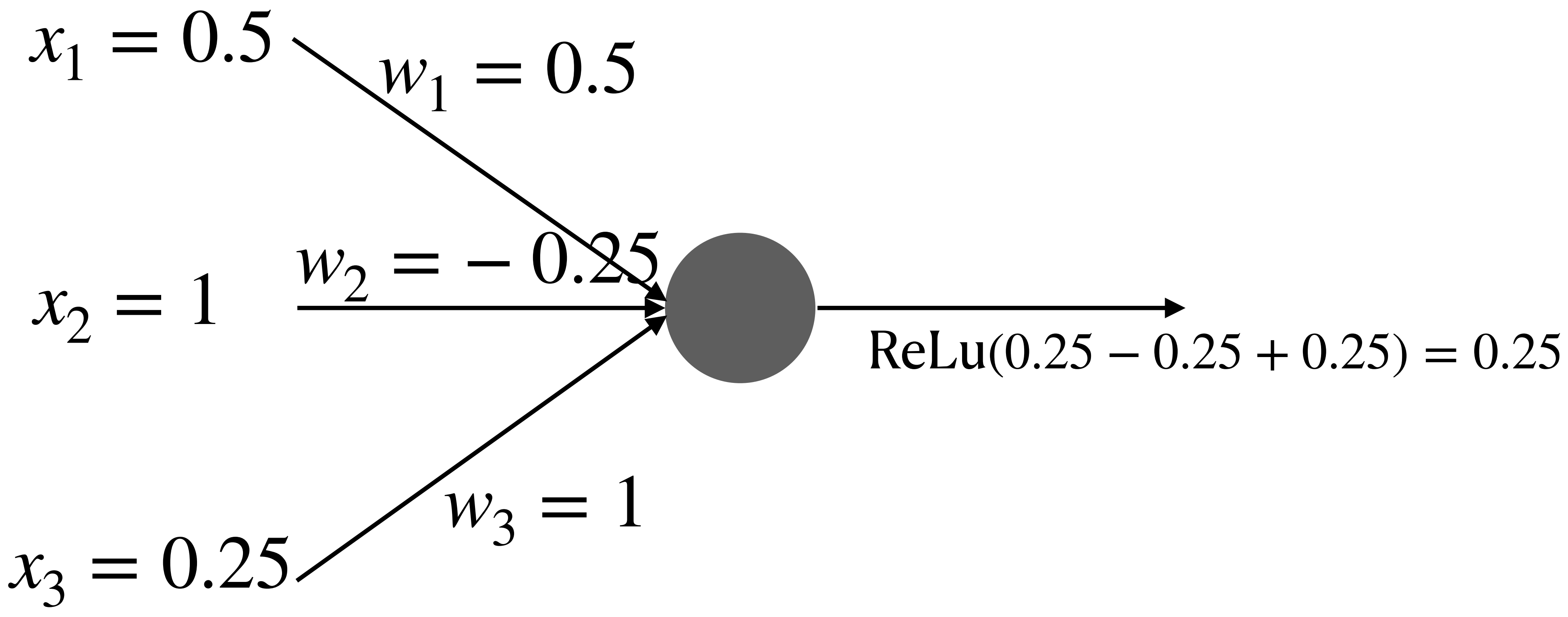
$$\text{ReLU}(x) = \max(x, 0)$$

  - Given an input *x*, the output is *x* if it is positive and zero otherwise.



- Examples:

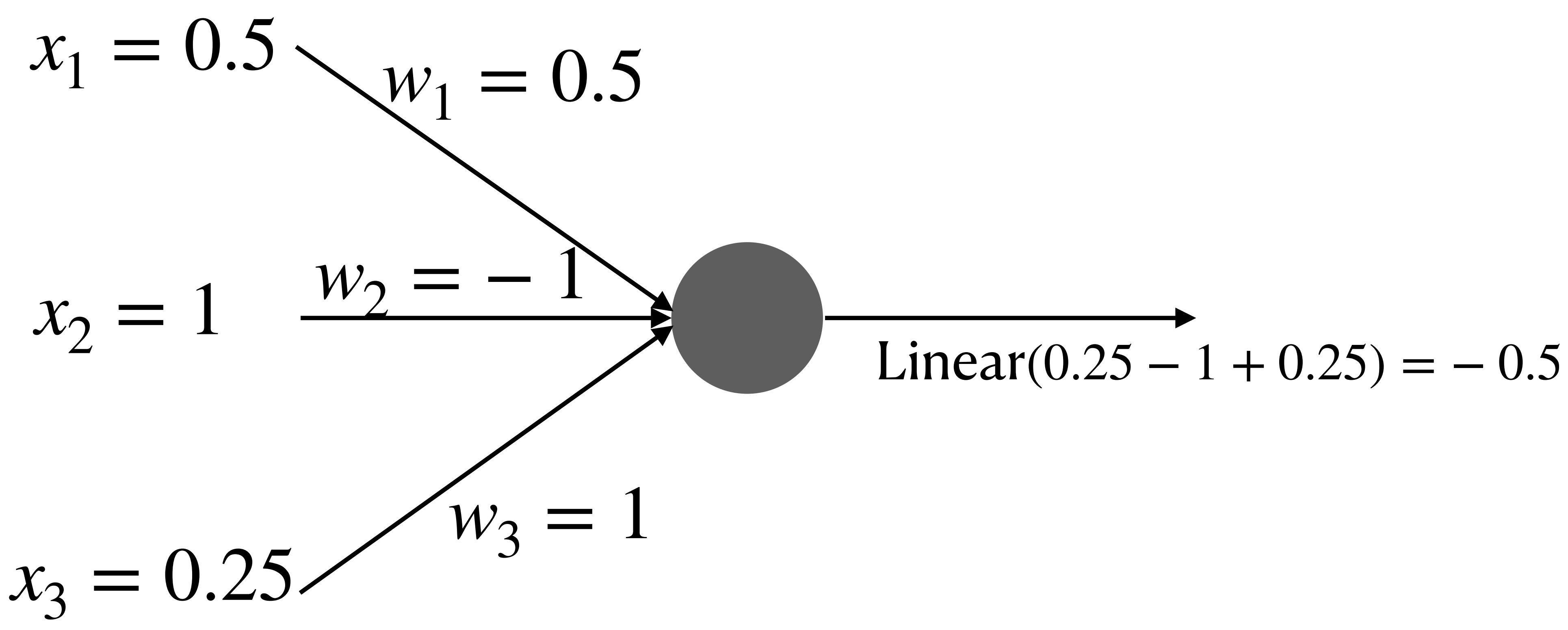


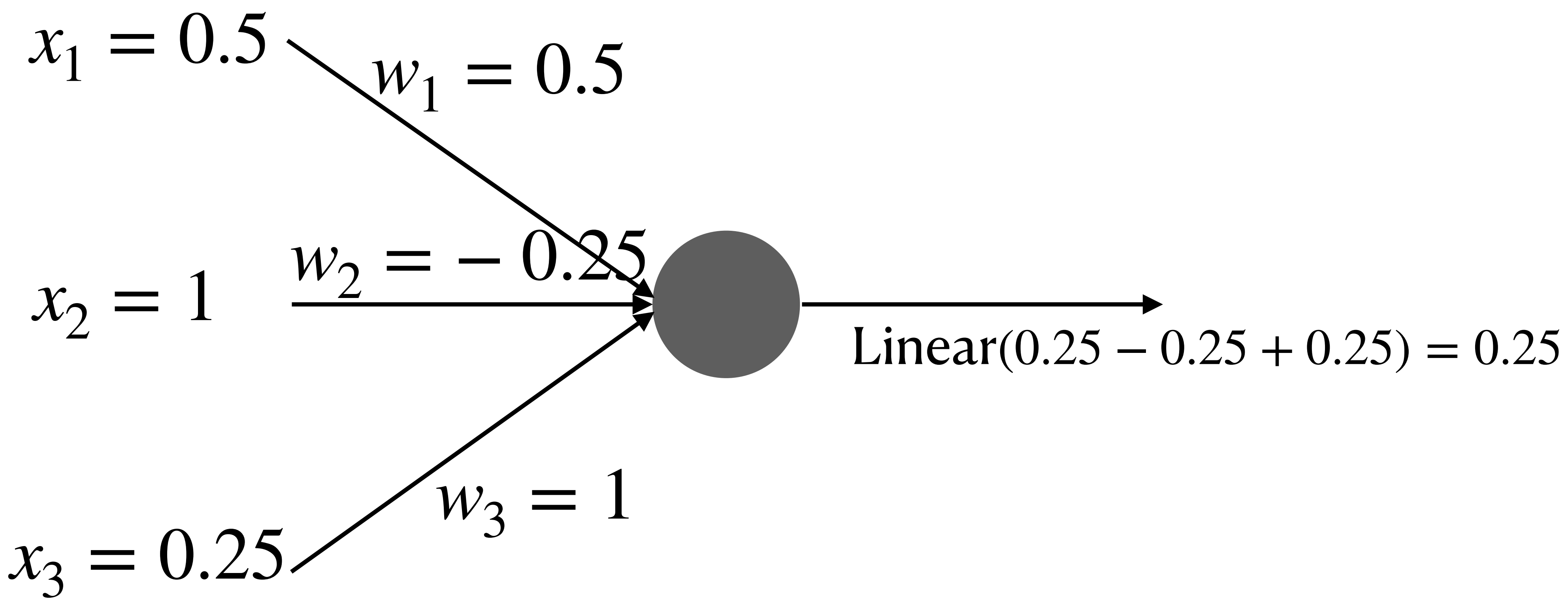


Activation Function

Linear Function

- Under a **linear** activation function ($y = x$):
  - Given an input x, the output is the same as x
- Examples:

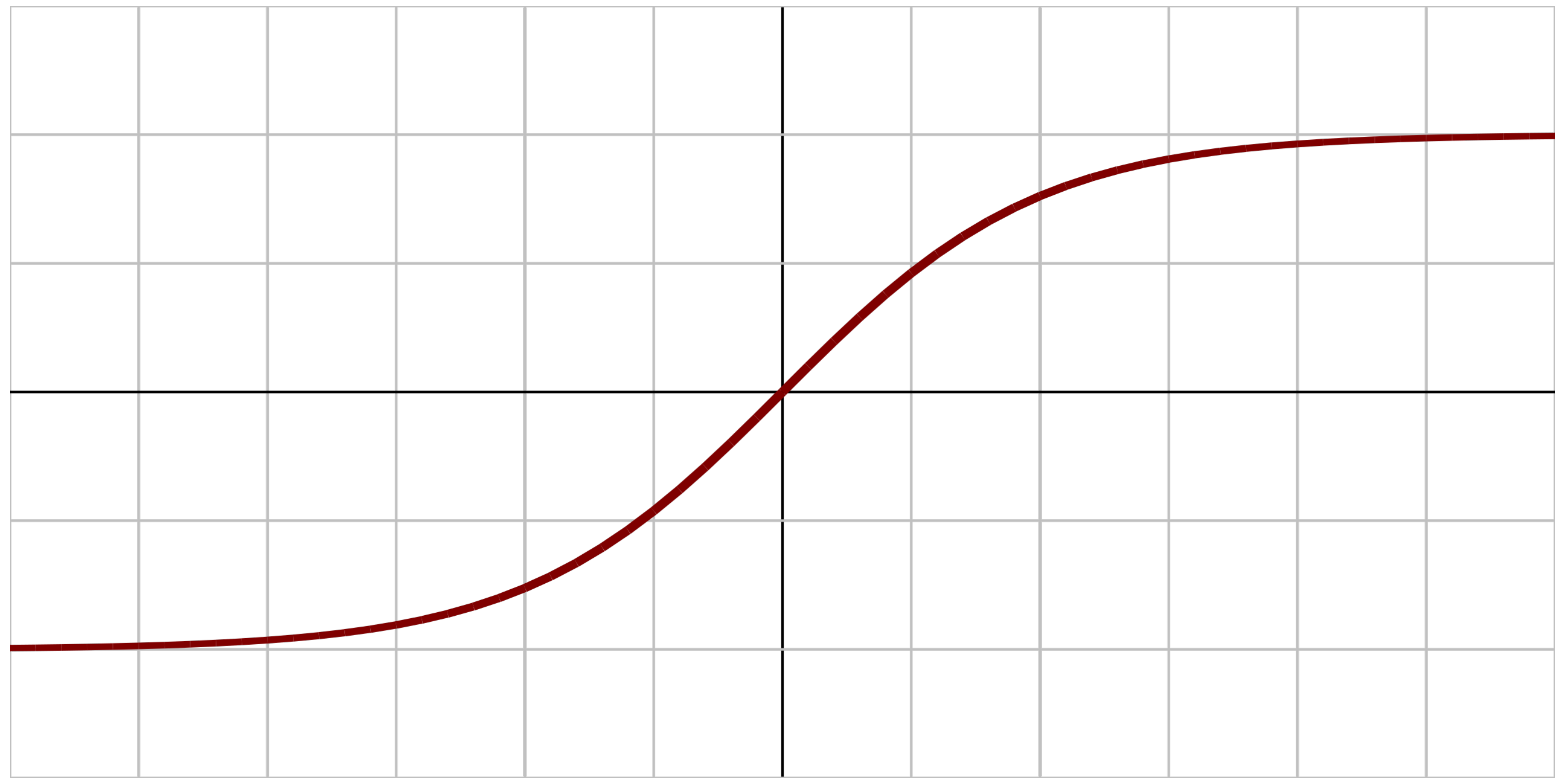




A linear activation function means a linear model regardless of the number of hidden layers

Other Activation Functions

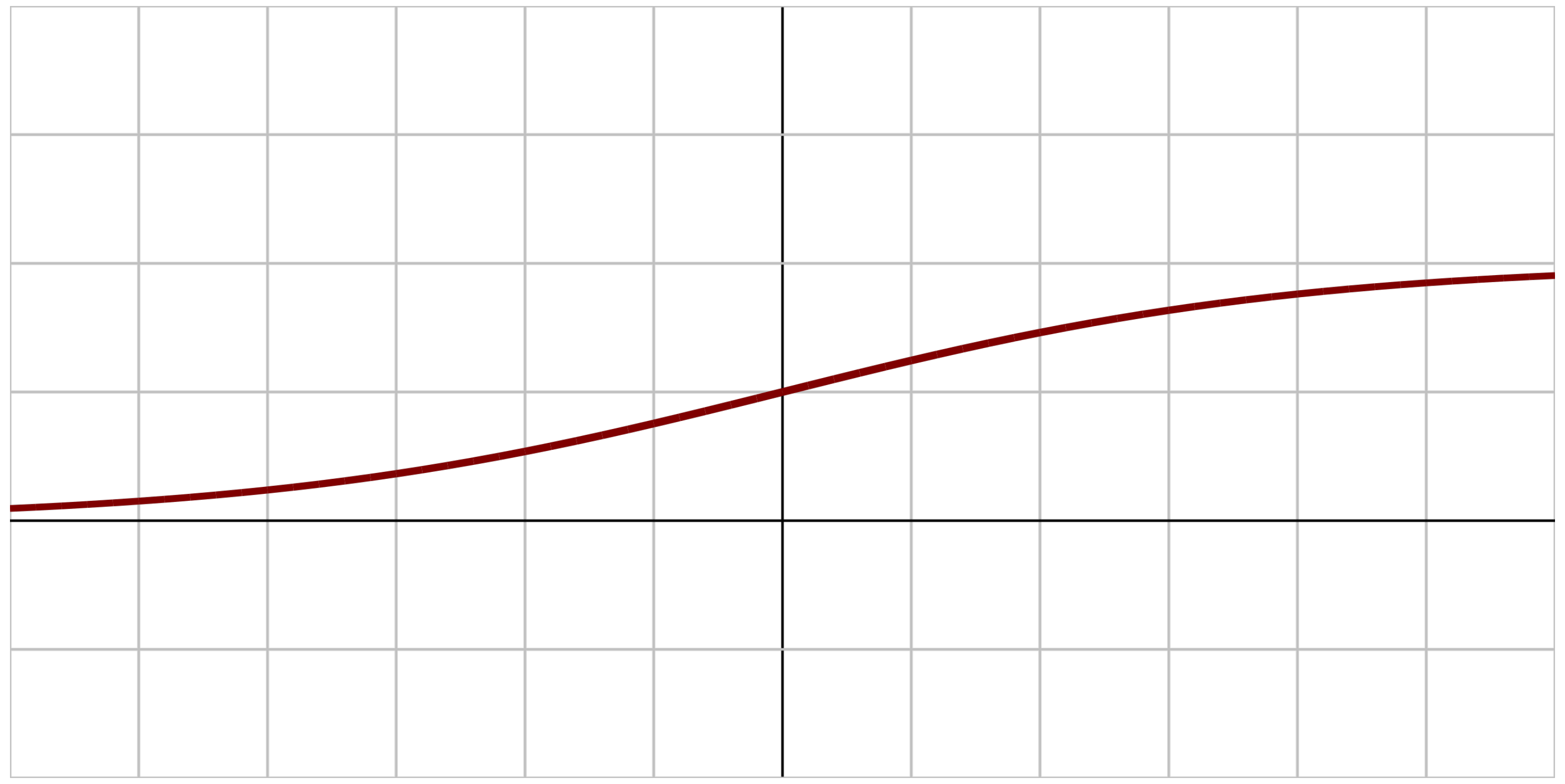
Tanh and Sigmoid — “S”-shaped functions



Tanh function

$$\text{Tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

“Tanh” rhymes with “branch”



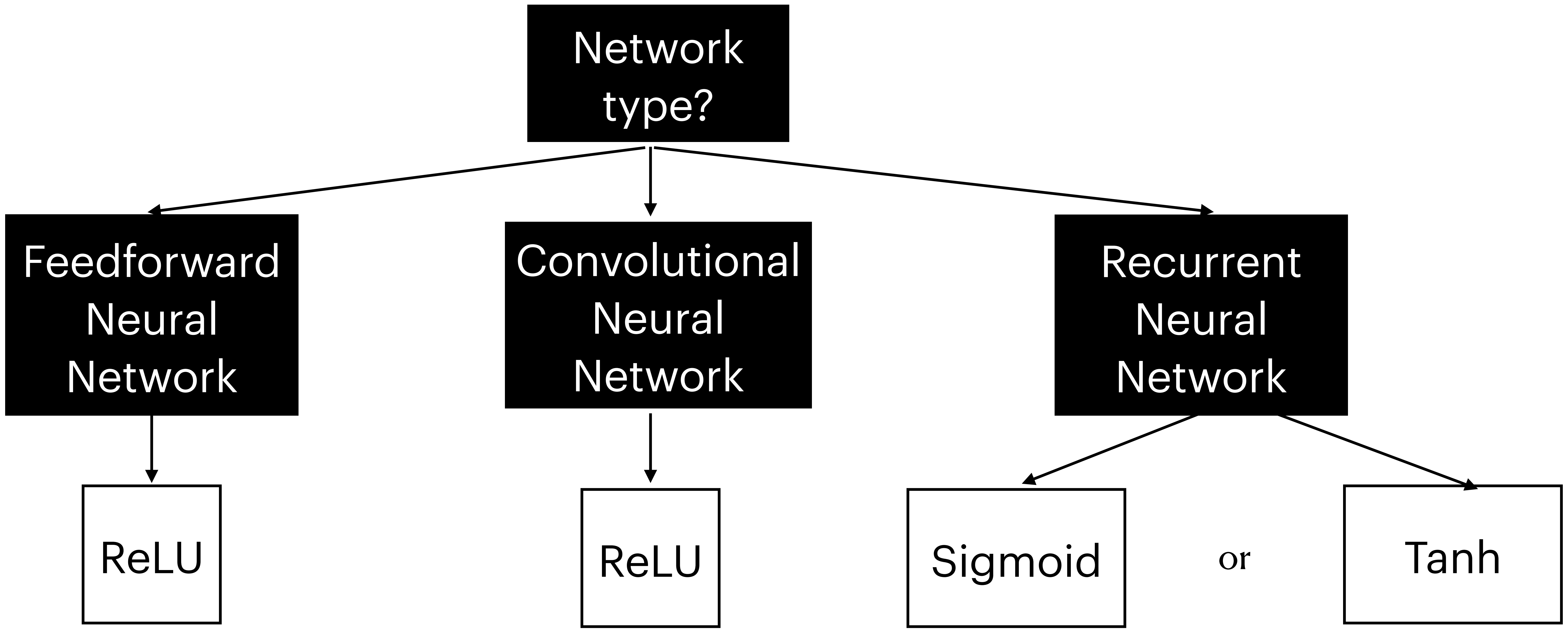
Sigmoid function

$$\text{Sigmoid}(x) = \frac{e^x}{e^x + 1}$$

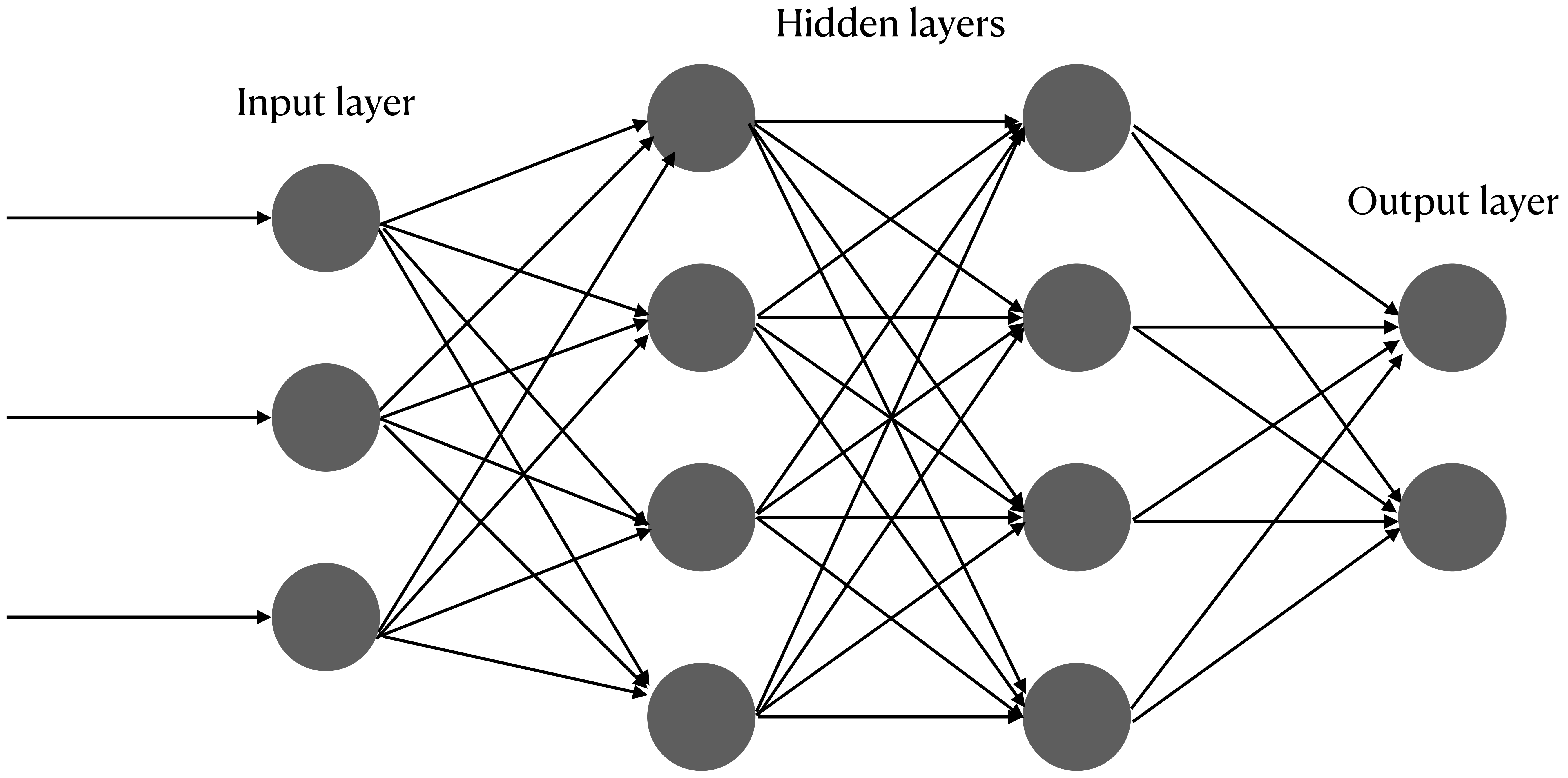
How to Choose Activation Function

A bit of a history lesson

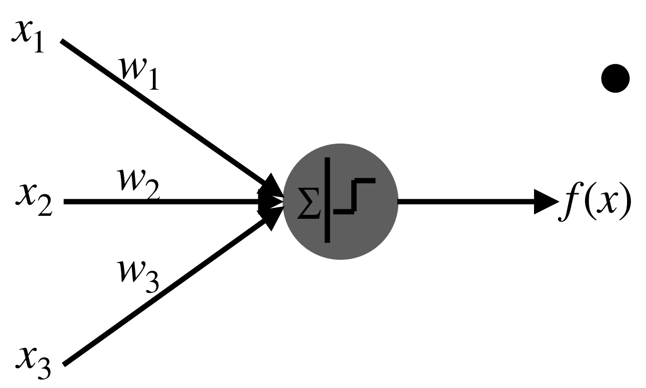
- Until mid-1990s: Sigmoid was the default activation function for training neural networks
- Between mid-1990s to 2010s: Tanh became default activation function for hidden layers
- 2010s to now: the default recommendation is **ReLU** for most of the modern neural networks



Why Feedforward Neural Network Works



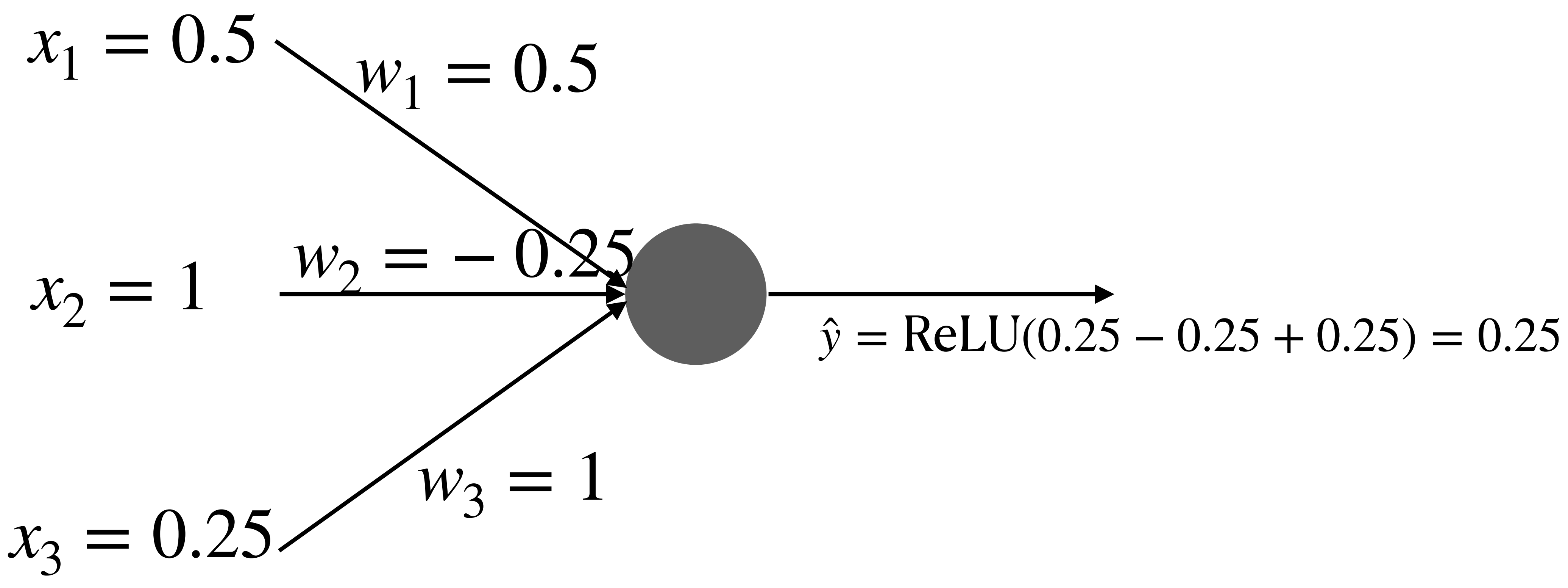
Feedforward Neural Network



- Universal Approximation Theorem: A feedforward network with at least one hidden layer with any "squashing" activation function (e.g., ReLU, Sigmoid, and Tanh) can learn any continuous input-output function

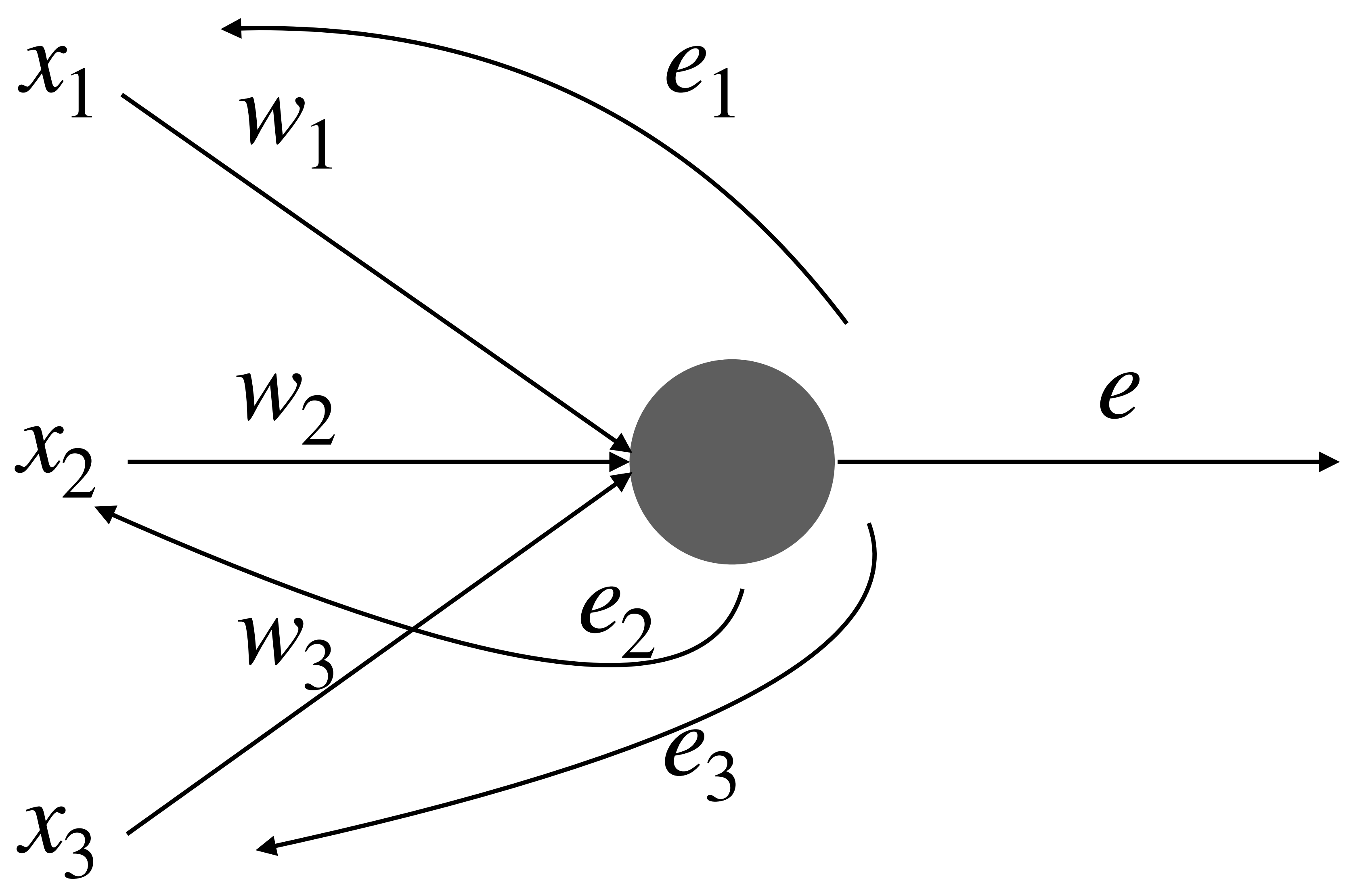
How Feedforward Neural Networks Learn

It's Essentially About Learning the Weights!



- We start with randomly assigned weights.
- In the above example, the predicted output $\hat{y} = 0.25$
- If the actual output is 0.5, then the MSE (mean squared error) is $e = (y - \hat{y})^2 = 0.0625$
- Now what? How can we use the *feedback* to adjust the weights?

Error Backpropagation



$$e_1 = \left(\frac{w_1}{w_1 + w_2 + w_3}\right) \times e$$

$$e_2 = \left(\frac{w_2}{w_1 + w_2 + w_3}\right) \times e$$

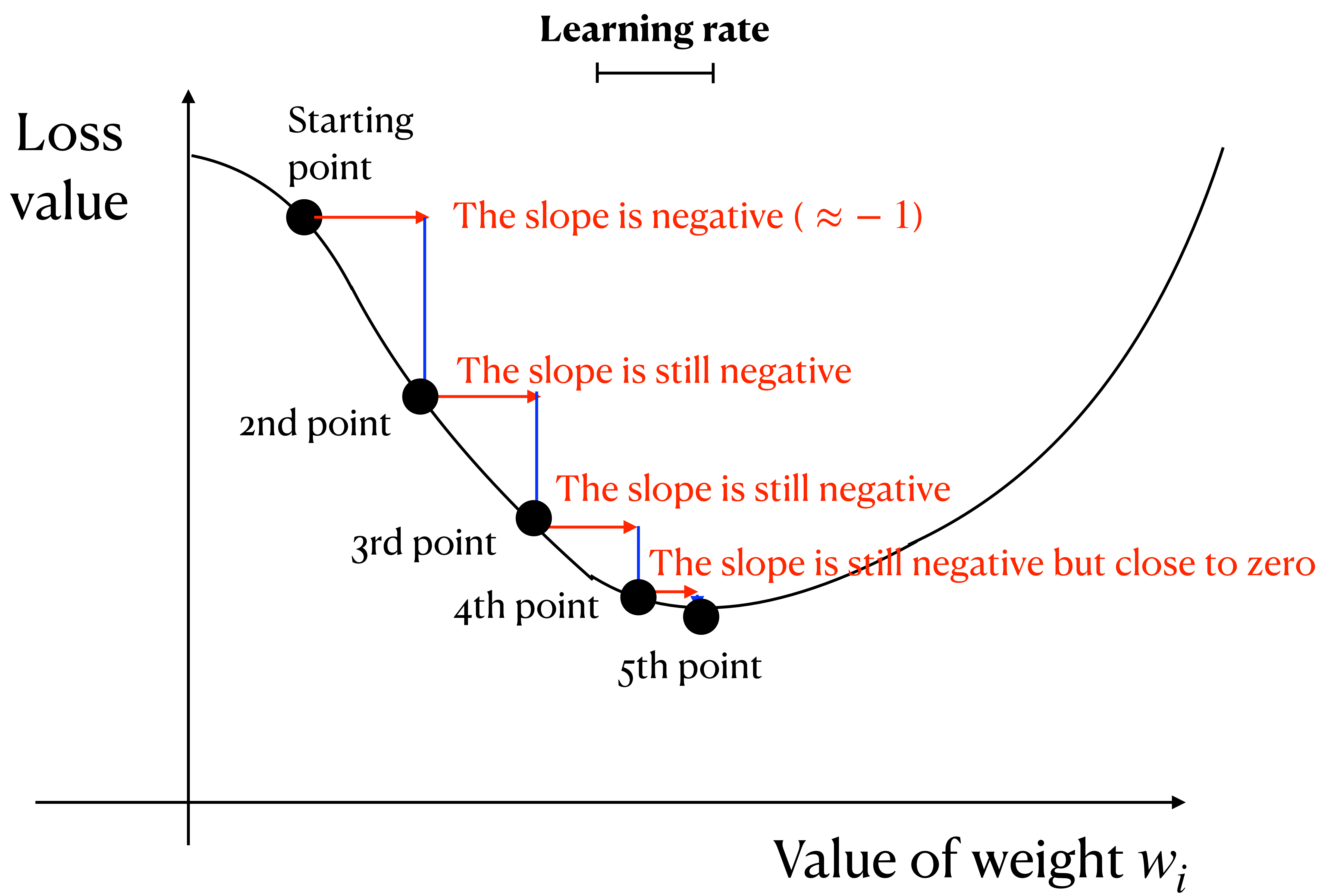
$$e_3 = \left(\frac{w_3}{w_1 + w_2 + w_3}\right) \times e$$

- We can next apply the gradient descent algorithm calculate the **slope** (i.e., gradient) of each weight —how increasing each weight will contribute to the error
- Once we know the slope, we can update the weight:

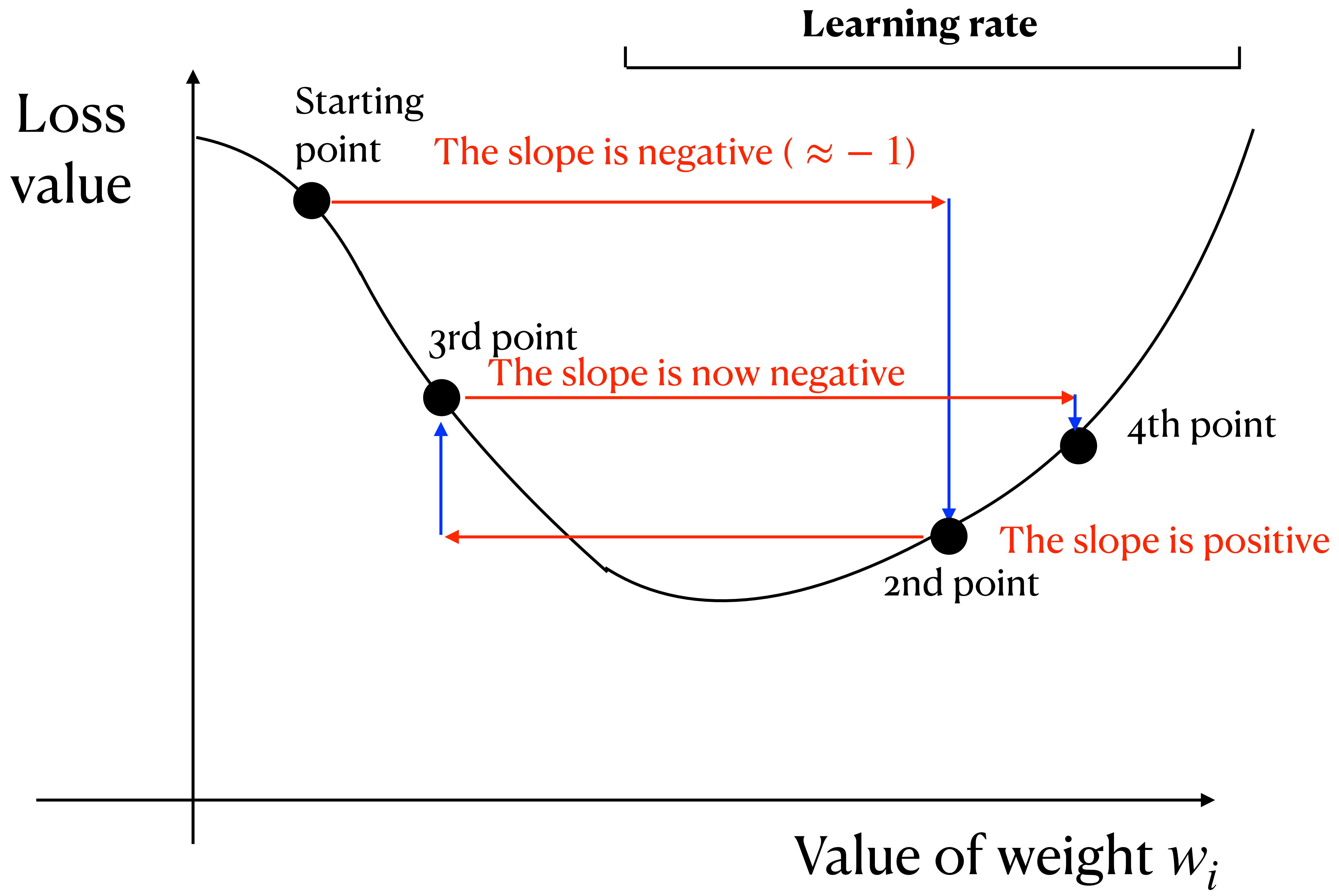
$$\textbf{new } w_i = \textbf{old } w_i - \textbf{learning rate} \cdot \textbf{slope}$$

Intuition Behind "new w_i = old w_i − learning rate · slope"

Adjusting the Weight to Minimize the Loss



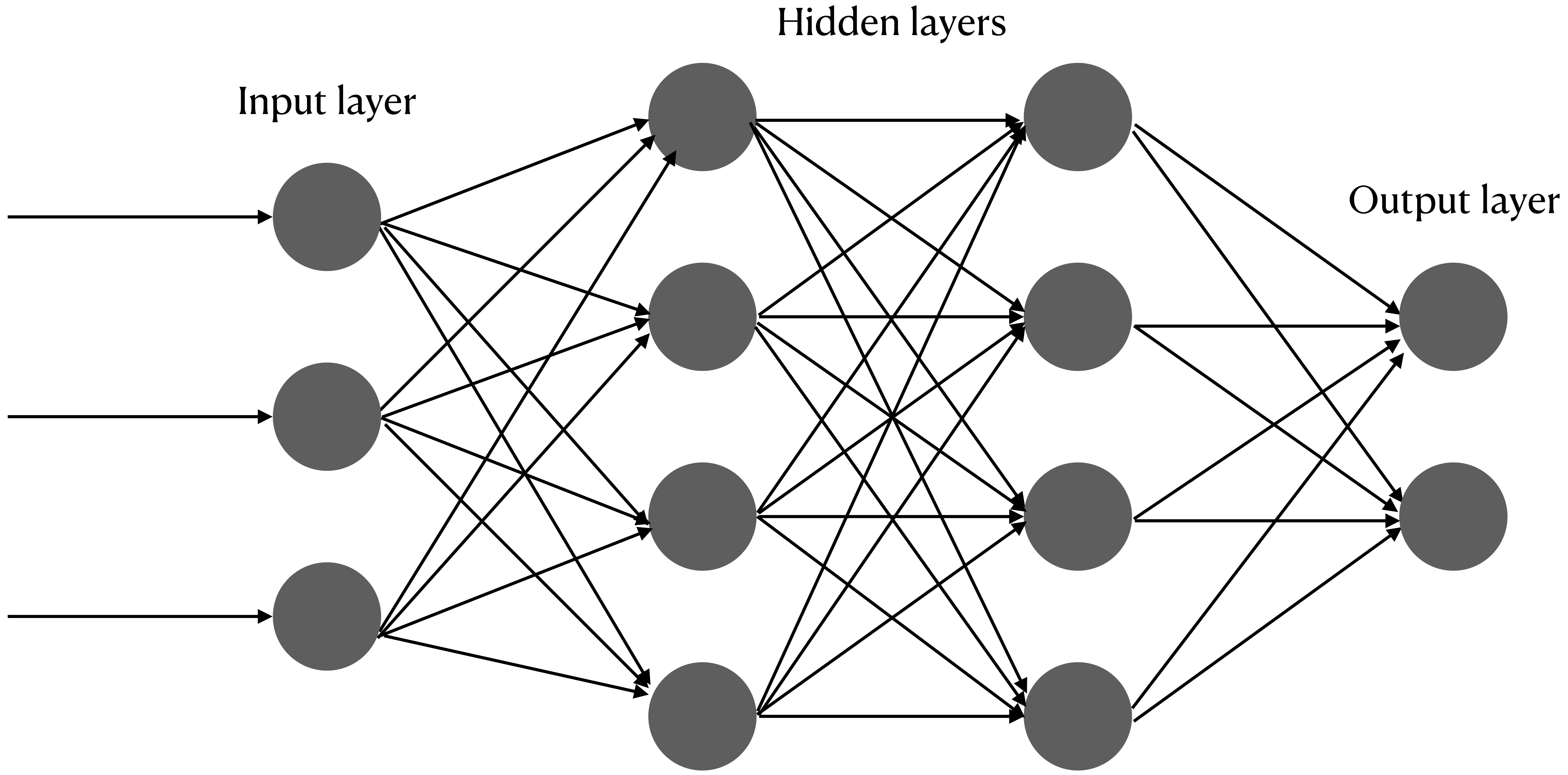
Avoid Excessively High Learning Rates



Excessively large learning rates may lead to a behavior like a quantum mechanics experiment gone horribly wrong

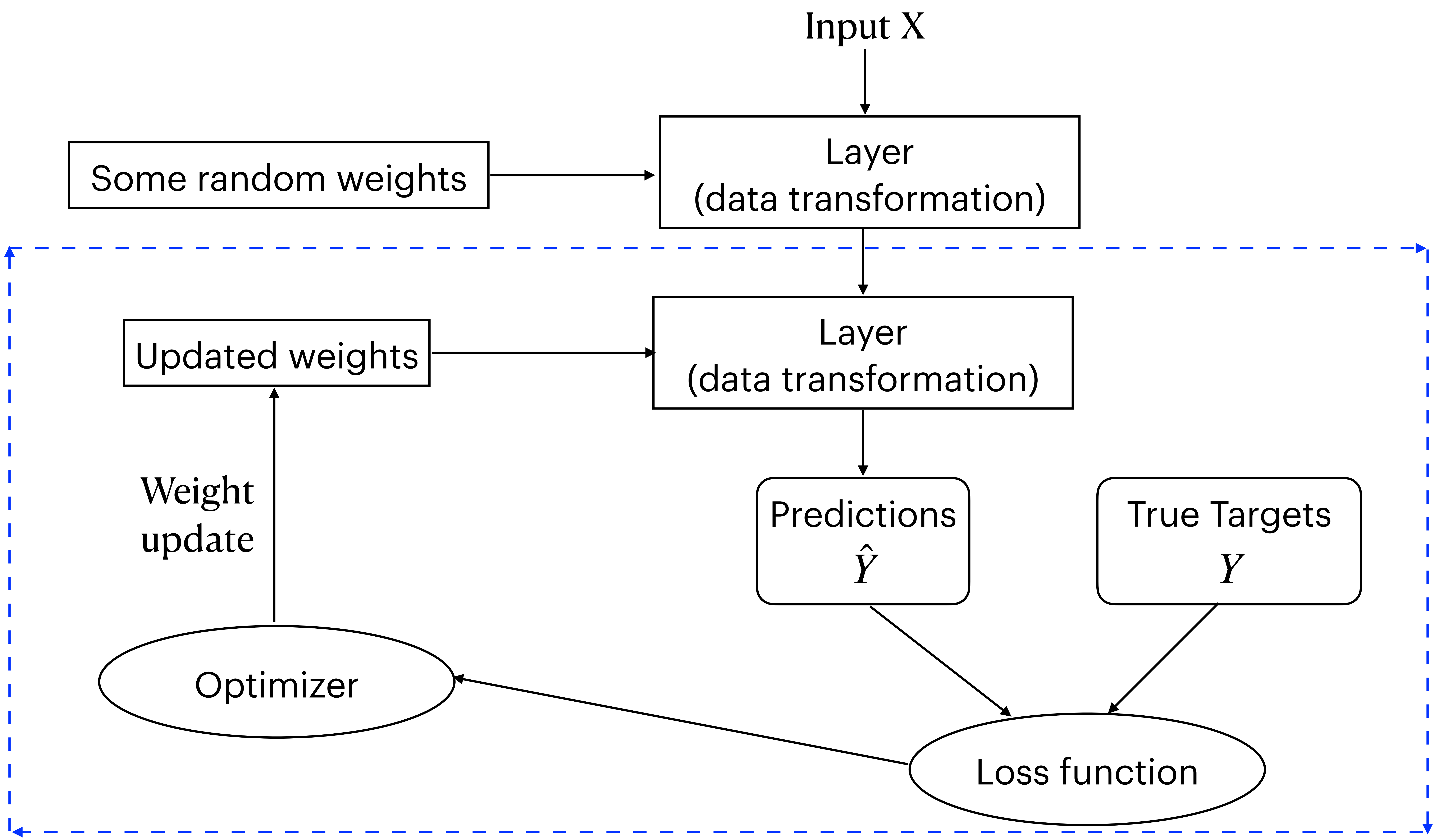
Deep Learning

- Deep learning is an incremental, layer-by-layer learning approach in which increasingly complex representations are developed to map the input to output



Feedforward Neural Network

Summary: Feedforward Neural Networks



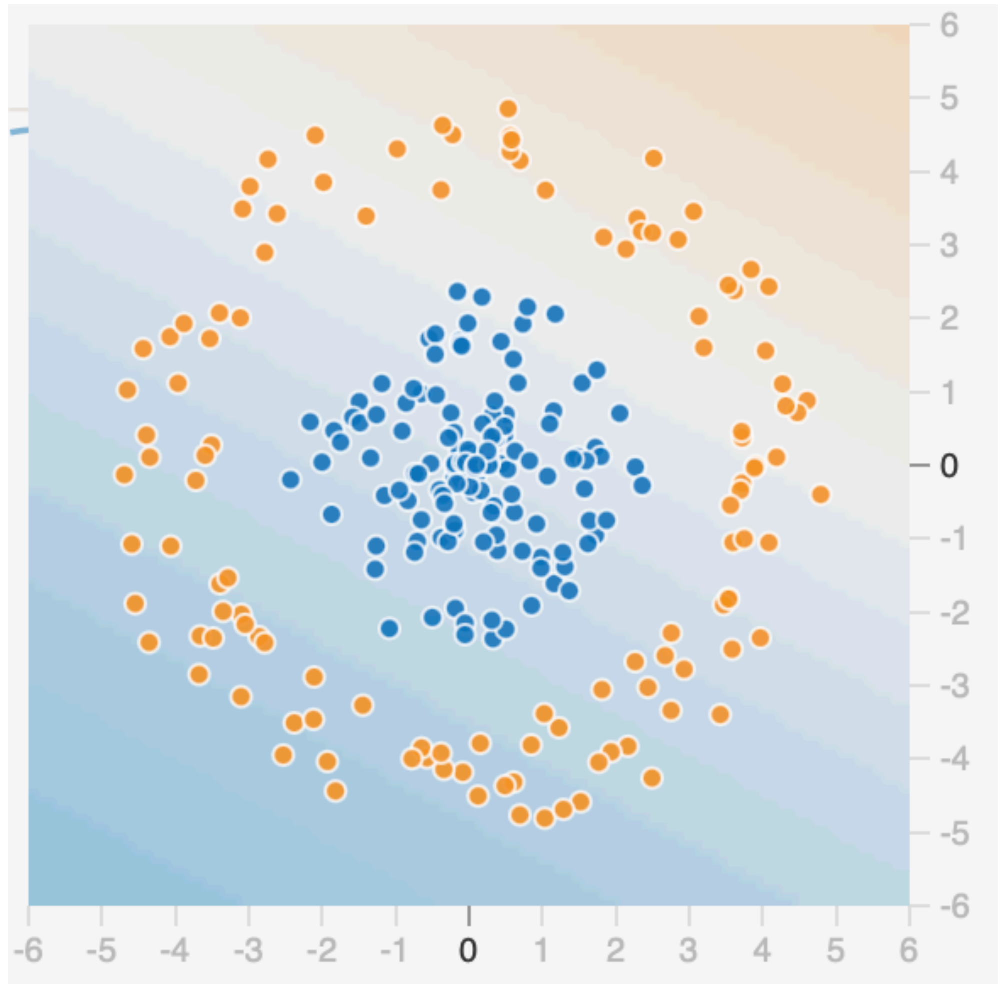
The loop goes on until the loss reaches a pre-specified objective

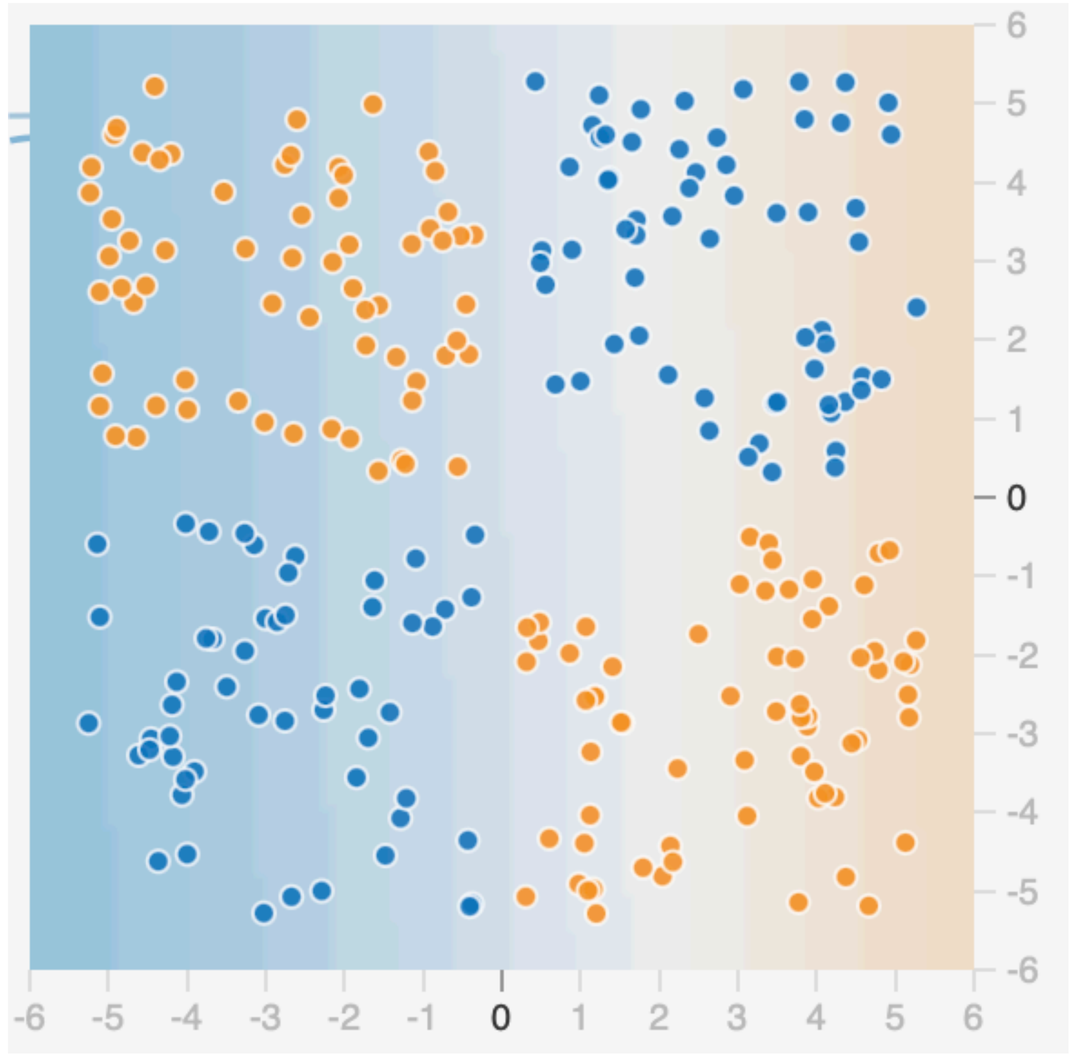
Time to Play!

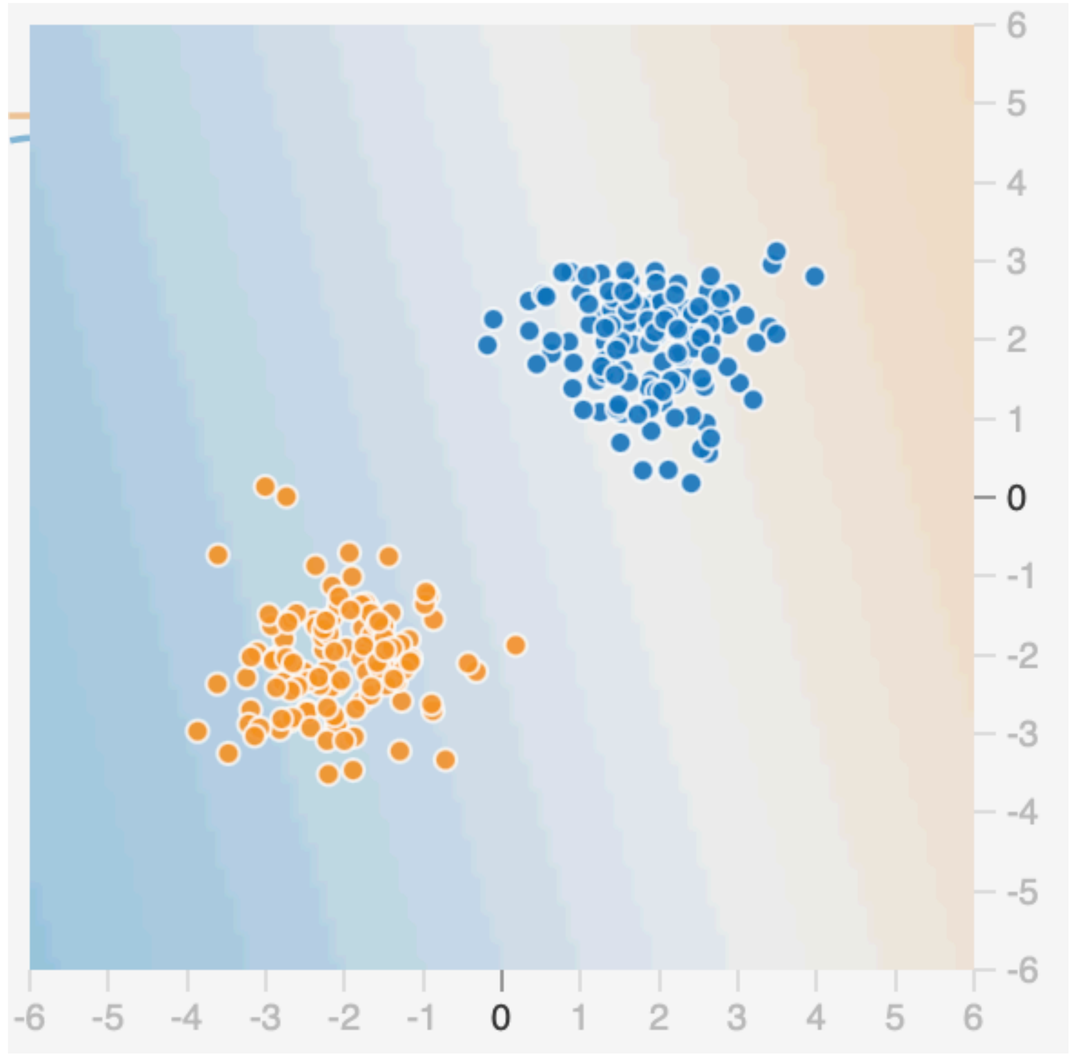
Let's go to the playground:
https://bit.ly/anntest

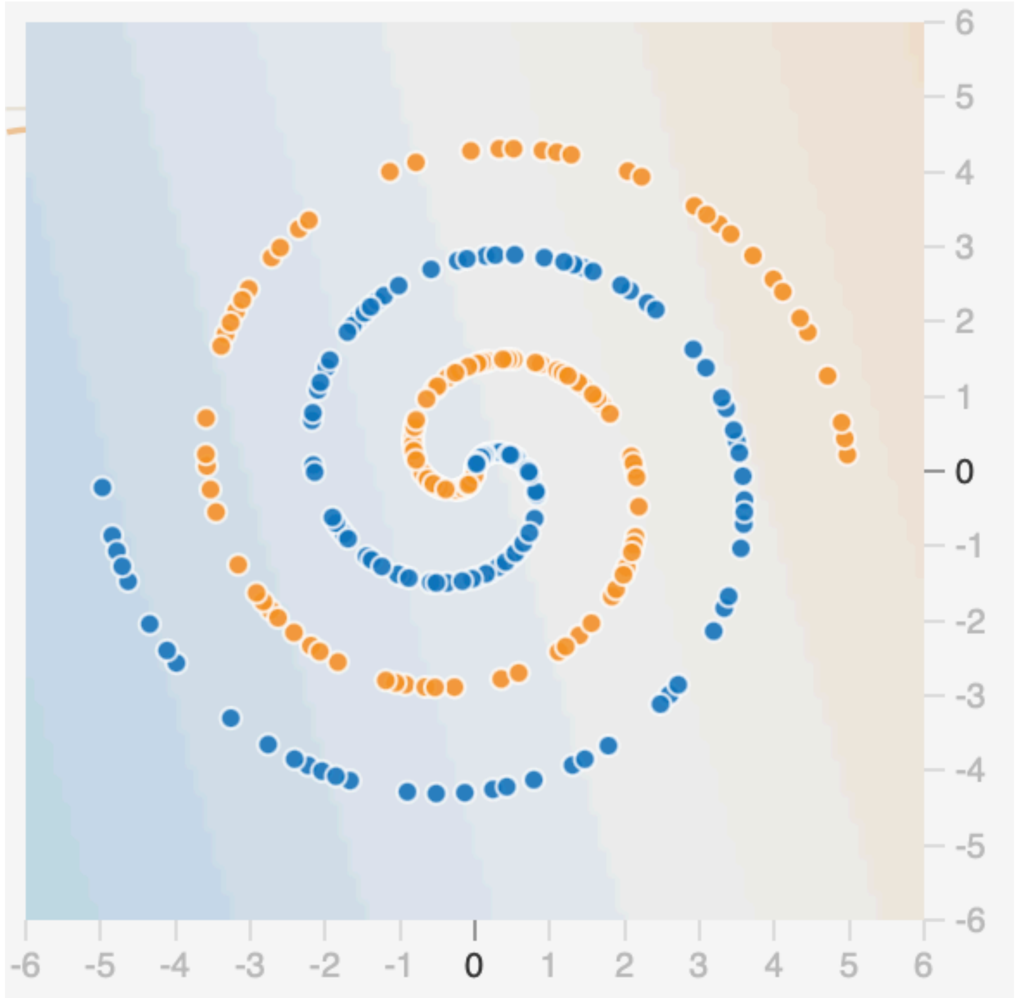
Let’s Start with the Four Problems

Solving Classification Problems Using Artificial Neural Networks

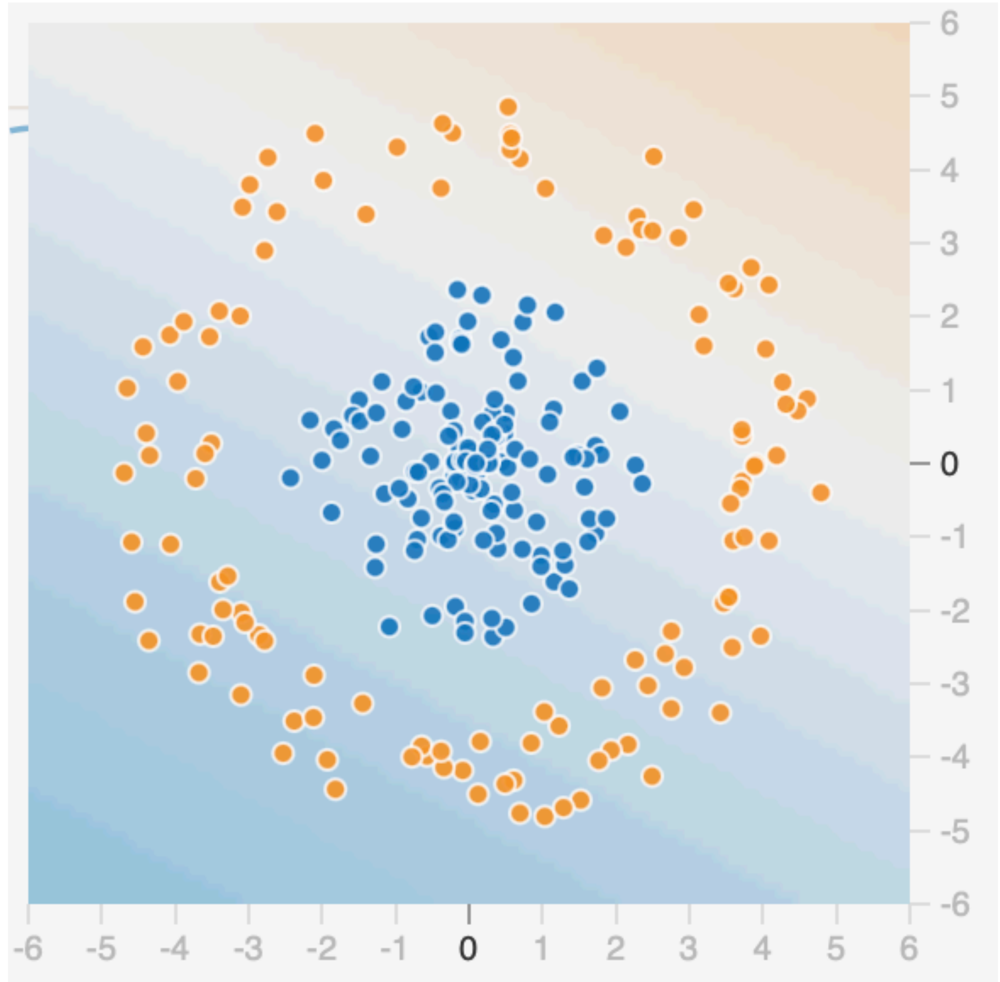


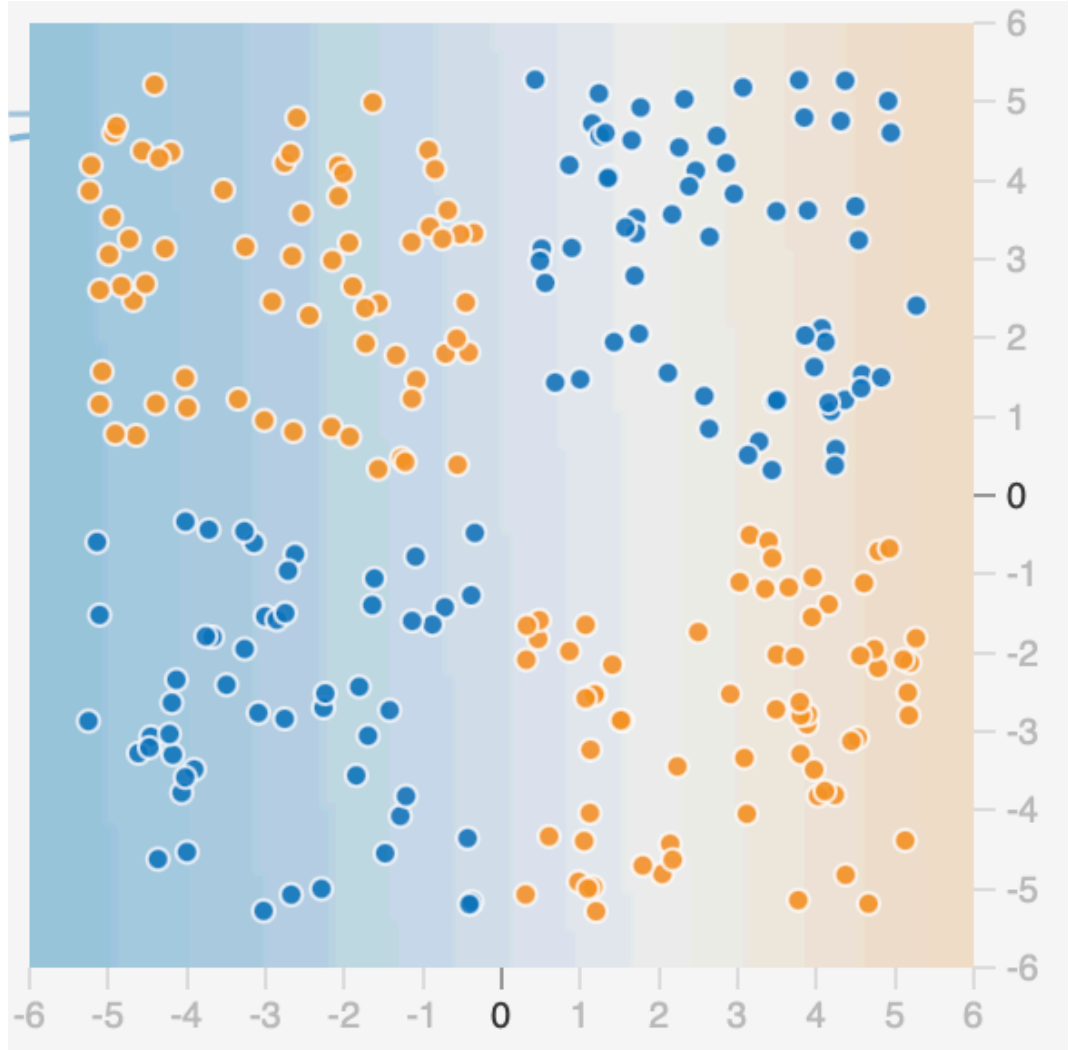


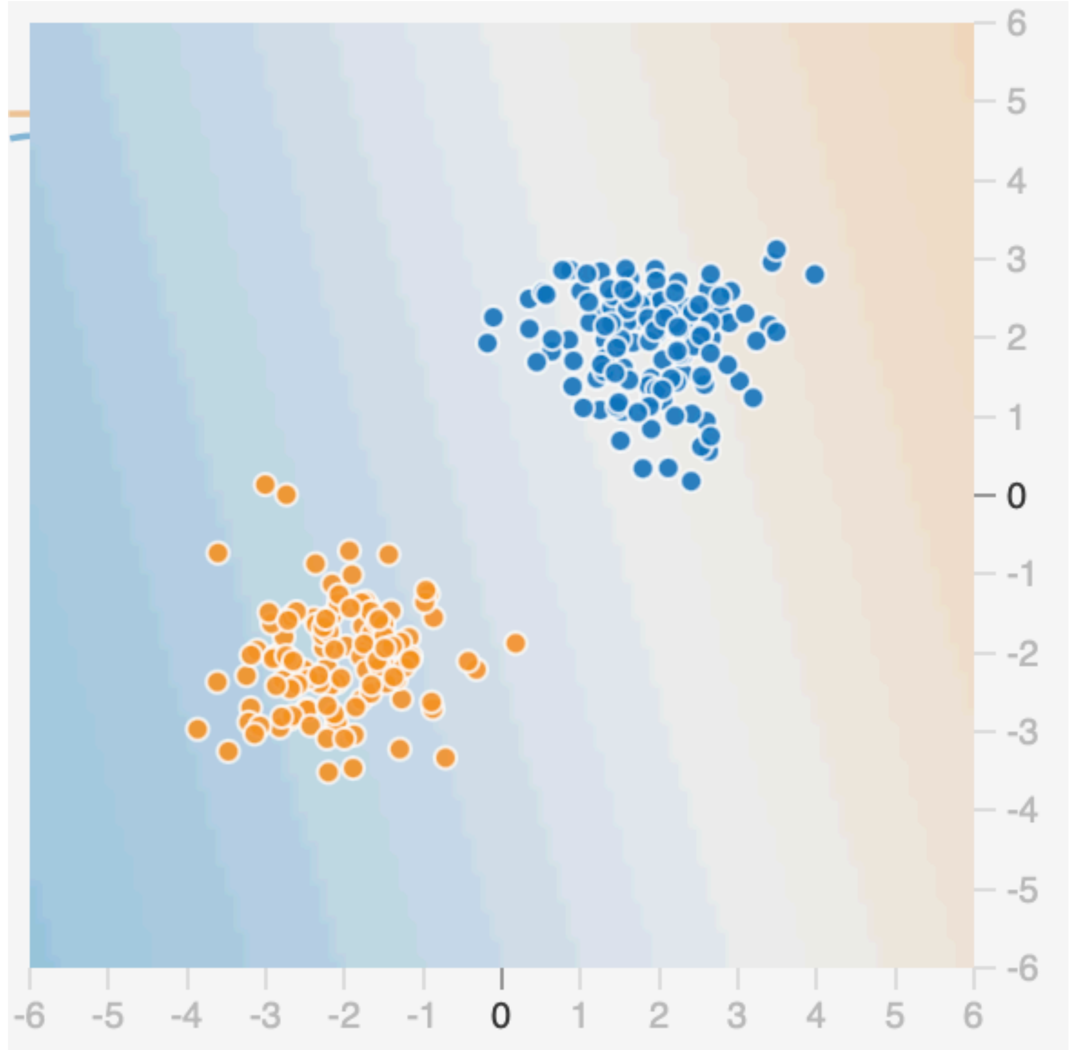


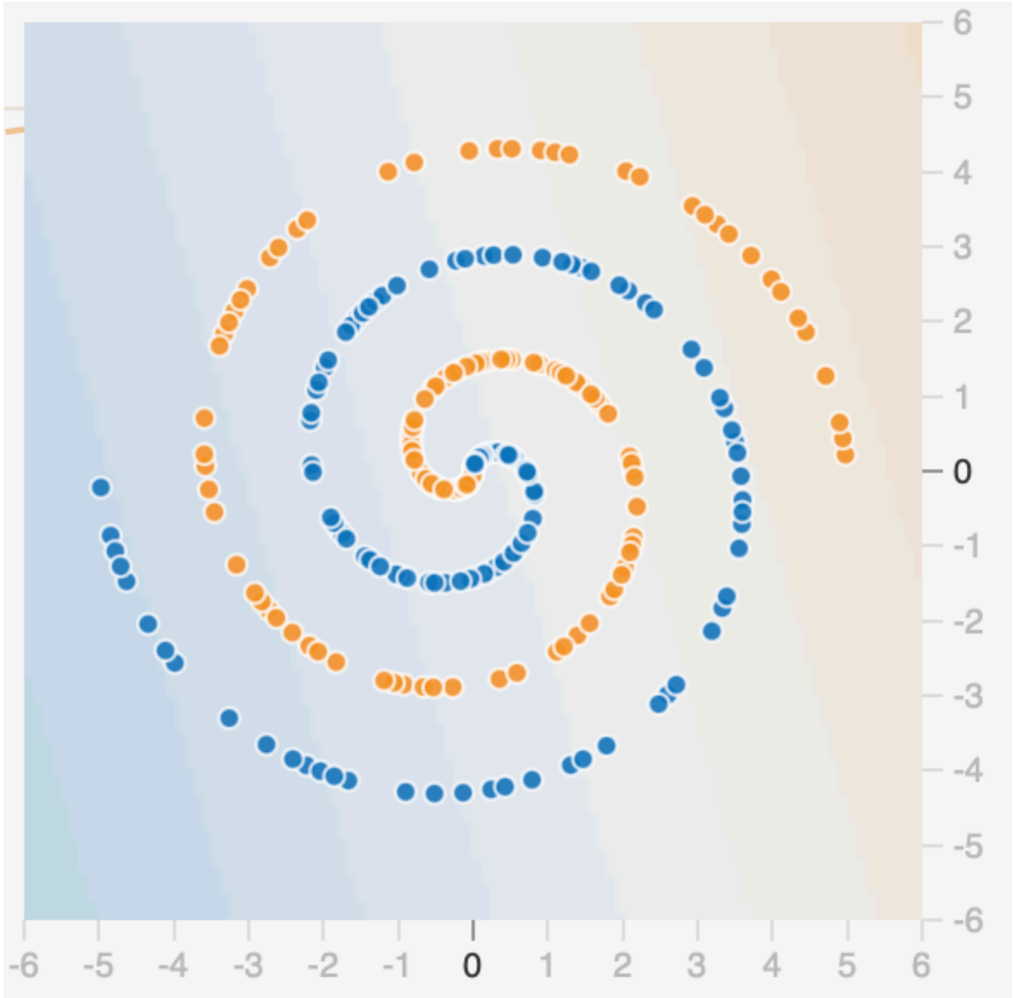


Are They Linearly Classifiable?





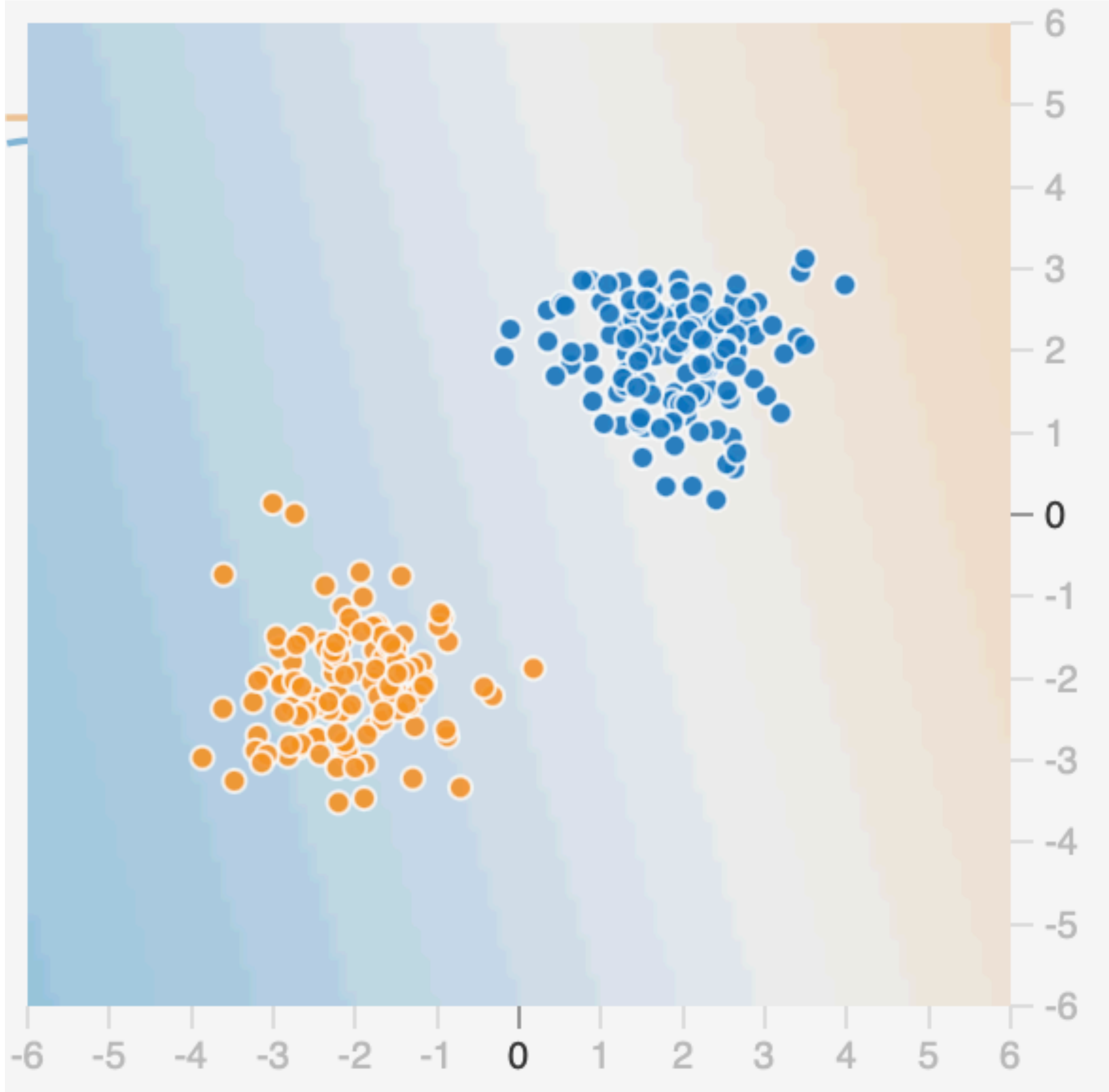




Definition: A pattern is **linearly classifiable** if there exists one line in the plane such that (1) all the blue dots are on one side of the line and (2) all the yellow dots are on the other side.

Which of these patterns are **linearly classifiable?**

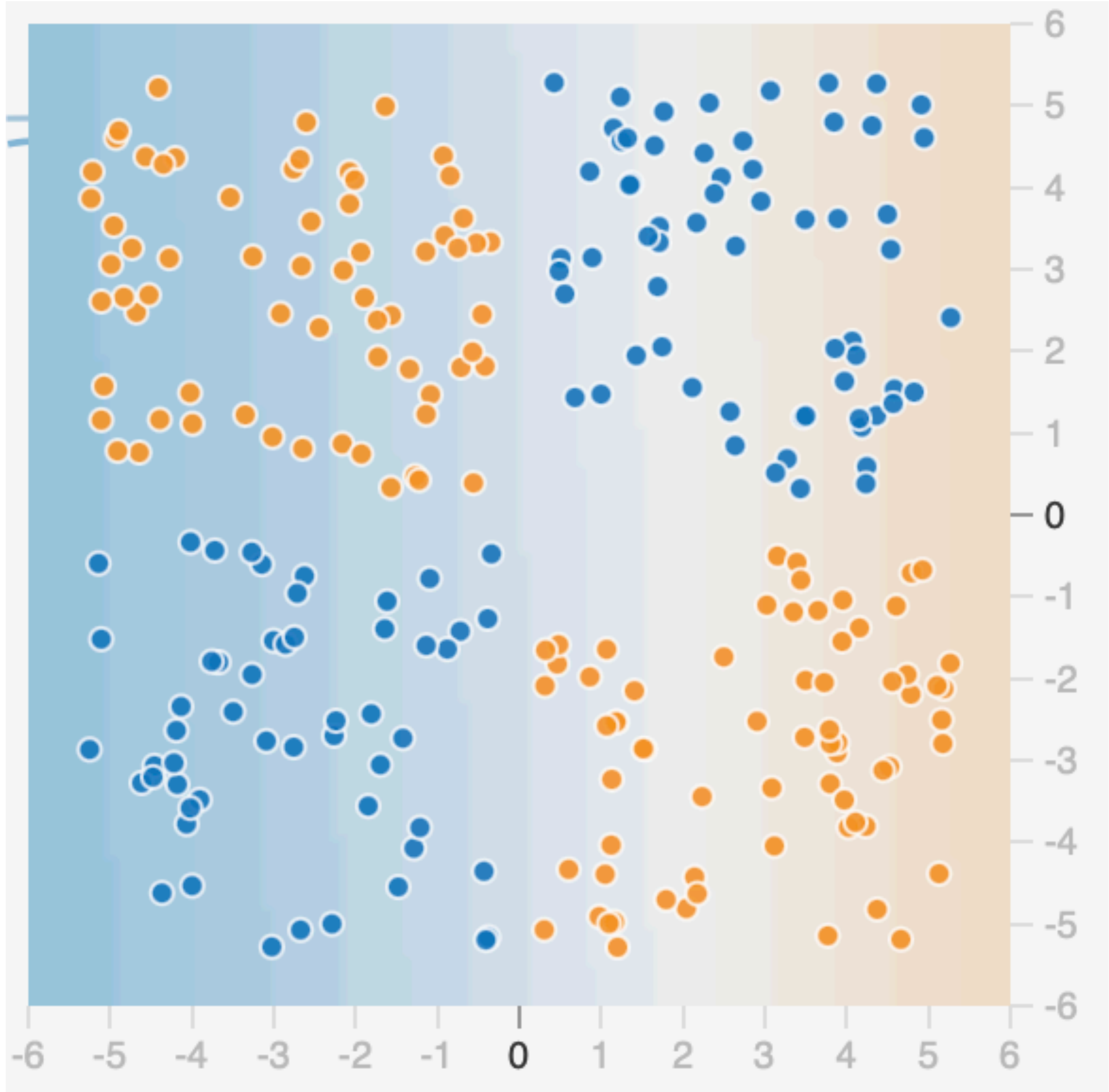
Task 1: Learning a Linearly Classifiable Pattern



- Try different activation functions
  - ReLU
  - Tanh
  - Sigmoid
  - Linear

- Does it matter which activation function we use?

Task 2: Learning a Non-Linearly-Classifiable Pattern



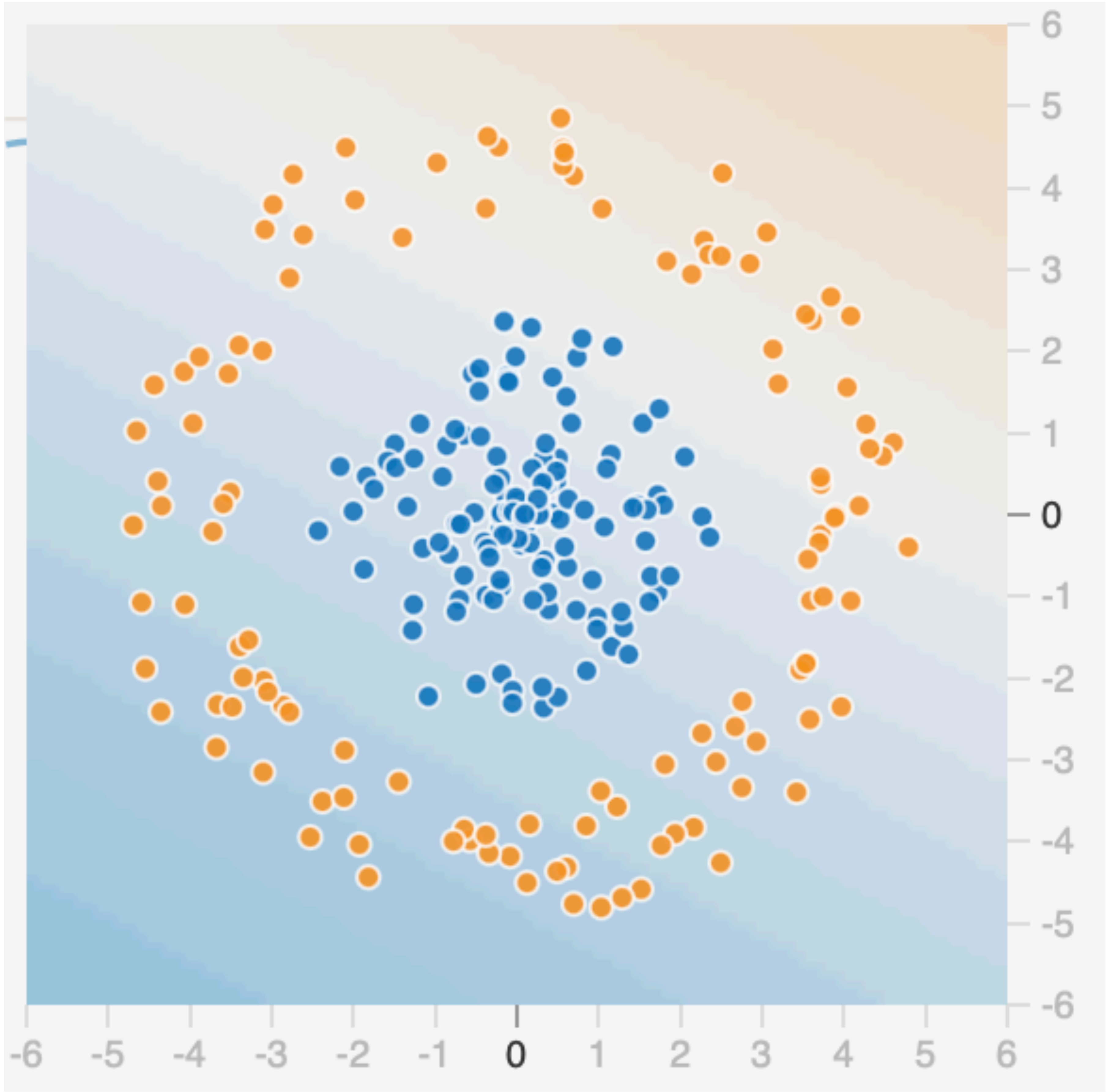
- Before each trial, hit the Reset the network button to get a new random initialization

- Start with a Linear activation function. Does it work? Why?

- Try other activation functions
  - ReLU
  - Tanh
  - Sigmoid

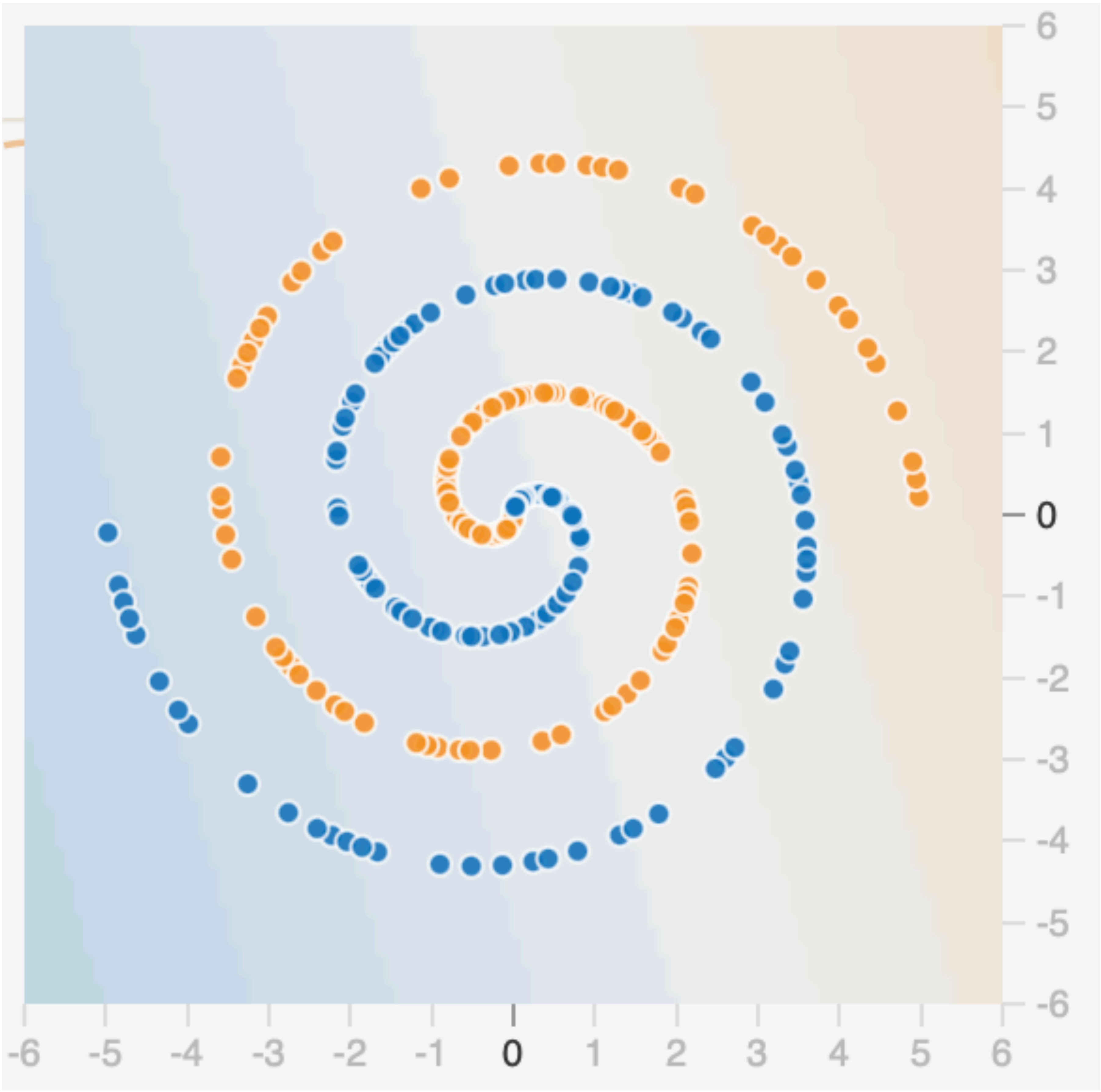
- Try different learning rates (0.00001, 0.003, 1, 10)

Task 3: Try a More Challenging Classification Problem



- Start with a Linear activation function. Does it work? Why?
- Try other activation functions
  - ReLU
  - Tanh
  - Sigmoid
- Try different learning rates (0.00001, 0.003, 1, 10)

Task 4: Learning A Spiral Pattern



- Try other activation functions
  - ReLU
  - Tanh
  - Sigmoid
- Try different learning rates (0.00001, 0.003, 1, 10)
- Can you generate observations about when the model overfits or underfits?

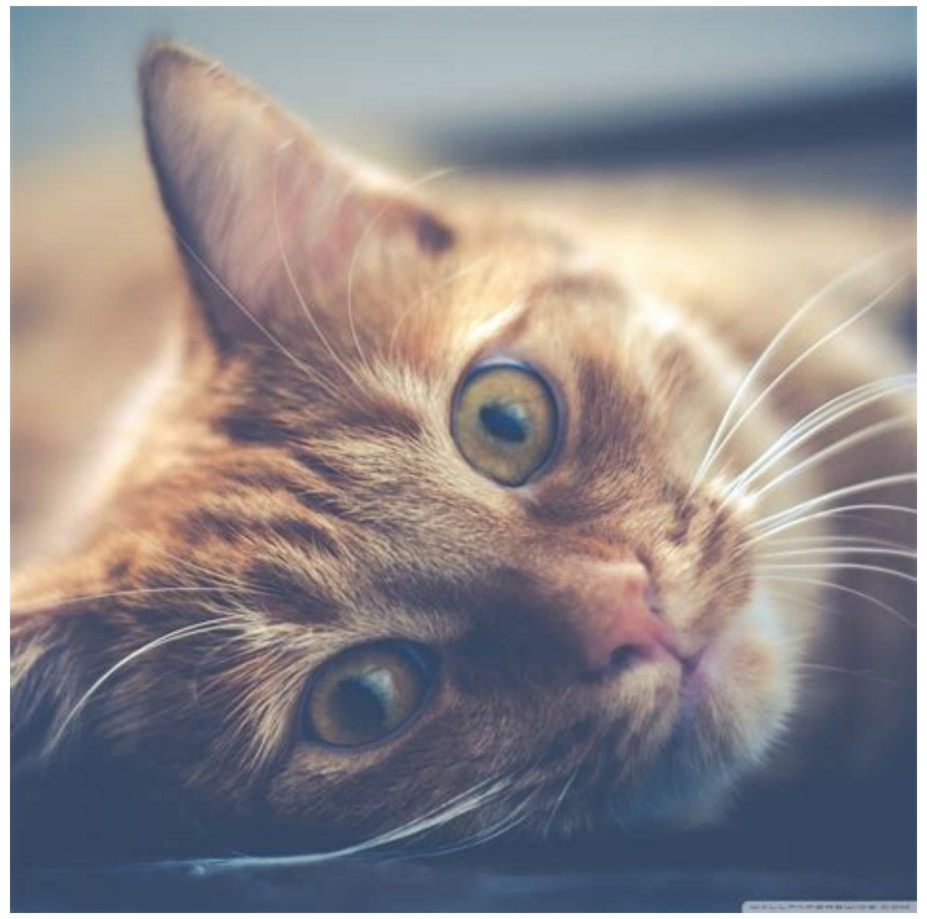
Let's now work *individually* to work on these tasks:

https://bit.ly/anntest

Time: 7minutes

Can Feedforward Neural Network Recognize Cats or Dogs?



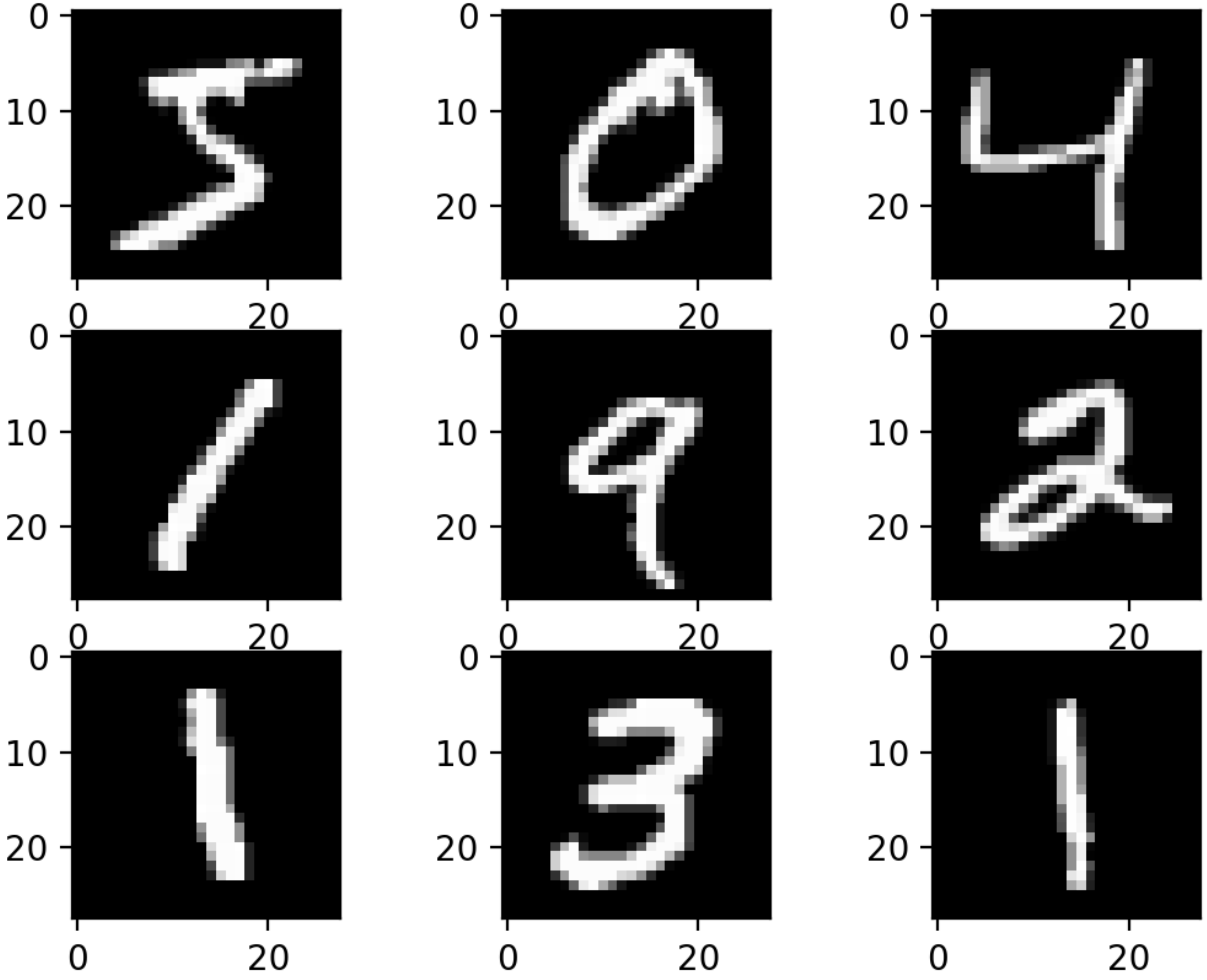



Can Feedforward Neural Network Recognize Cats or Dogs?

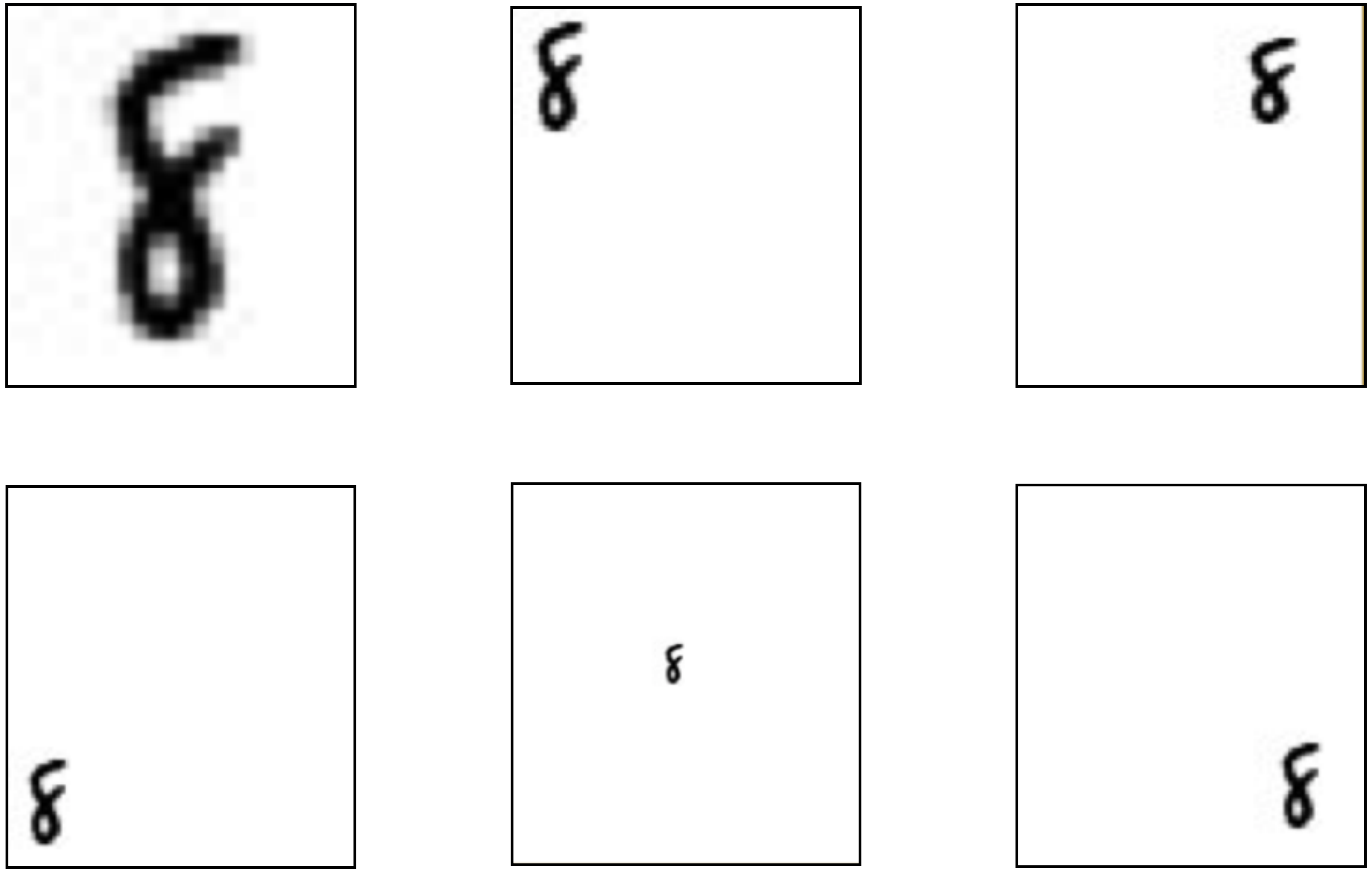




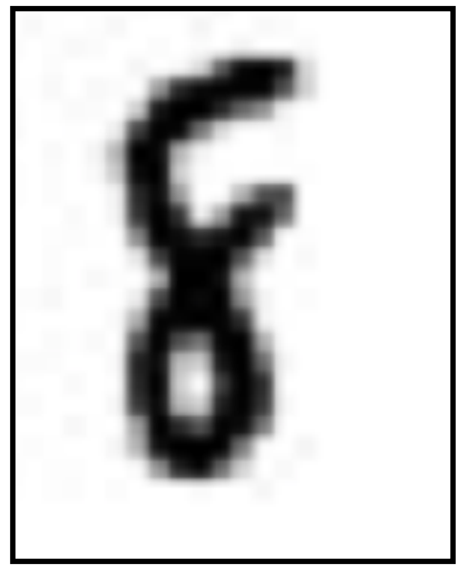

What about the MNIST Dataset?



Practical Training Images



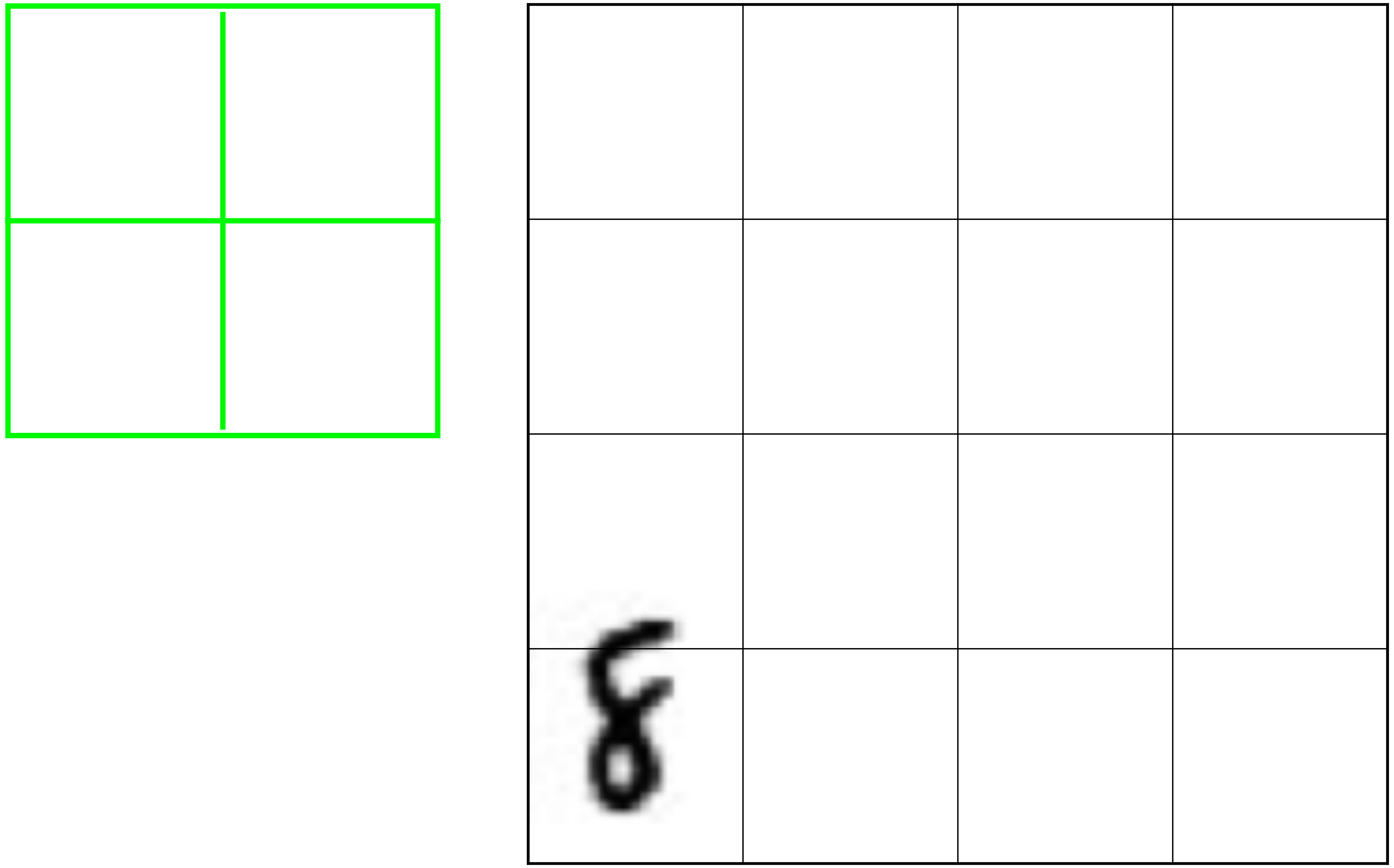
Original Training Image



How to “catch” the crucial information from ALL of these practical images?

Geitgey (2016)

Core Idea: Search with a "Sliding Window"



Geitgey (2016)

Convolutional Neural Networks (CNN)

- The “convolutional” part does the job of the sliding window
- The convolution step:

Convoltional filter

1	1
0	1

Input Feature Map

×0	×1	1	1
×0	×0	0	1
0	0	1	0
0	0	0	0

Output Feature Map

1	2	3
0	1	1
0	1	1

Exercise: Construct the Output Feature Map

What is the shape of the output feature map?

Convoltional filter

1	0	1
0	1	0
1	0	1

Input Feature Map

0	1	1	1
0	0	0	1
0	0	1	0
0	0	0	0

Output Feature Map

2	2
0	2

Are These Two Letters Identical?

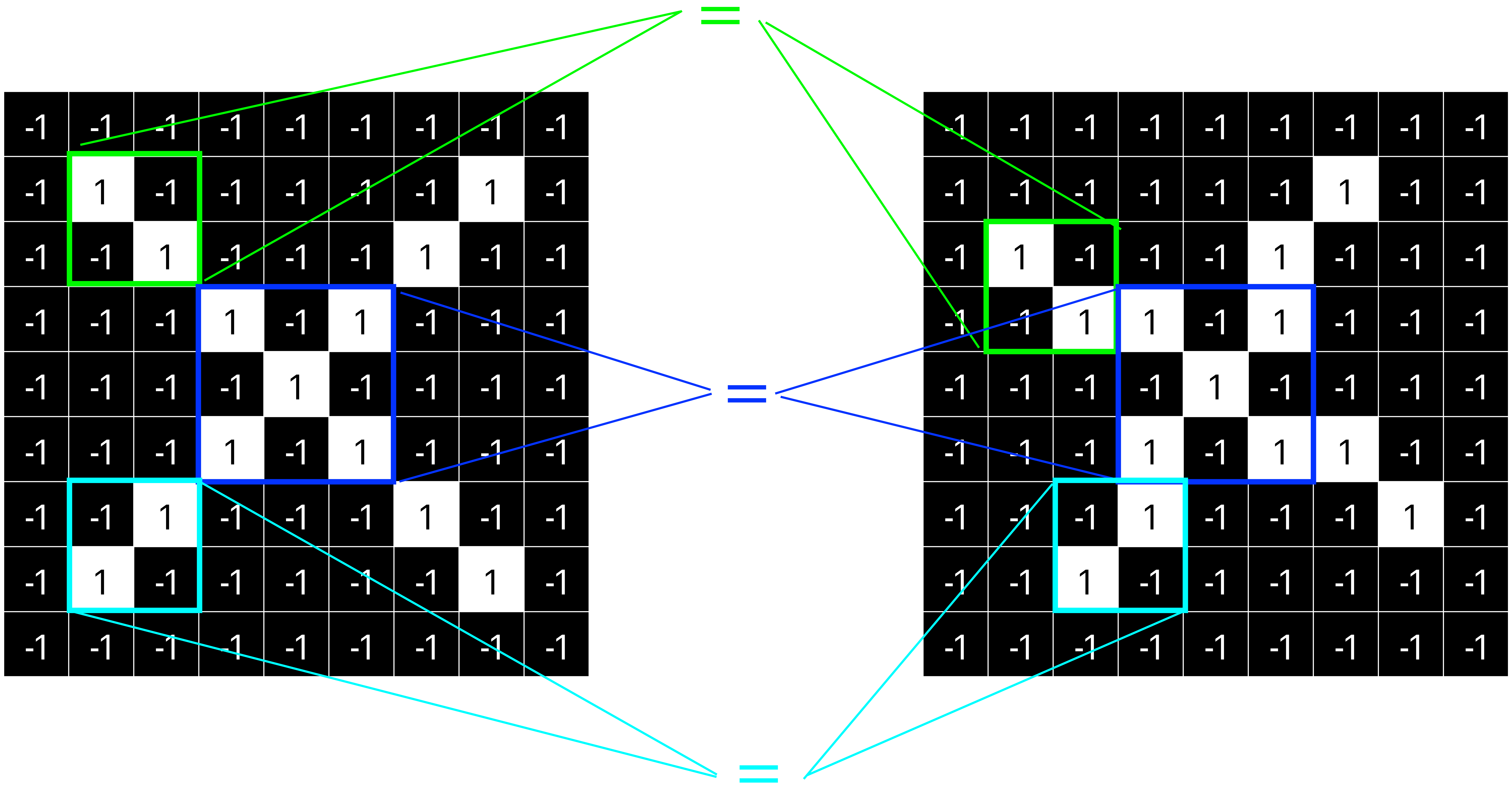
-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	1	-1	-1	-1
-1	-1	1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	1	-1	-1
-1	-1	-1	1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

Unfortunately, for a computer, the above two images can represent two very different patterns

We need to figure out a way for the computer to recognize shifted, shrunk, rotated, or deformed images.

Fundamental Features of Letter X



We Can Detect These Features Using Filters

Convoltional filter

1	-1	-1
-1	1	-1
-1	-1	1

Input Feature Map

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

Apply the convolution operation:

$$= 9$$

We Will Dive Deeper Into Convolutional Neural Networks (CNN) Next Class

Why Python?

- Python is likely the easiest programming language to learn and it’s completely free
- Python is expressive and powerful
- Python is **the language for artificial intelligence**
- Python is the most demanded programming language on the job market

TensorFlow

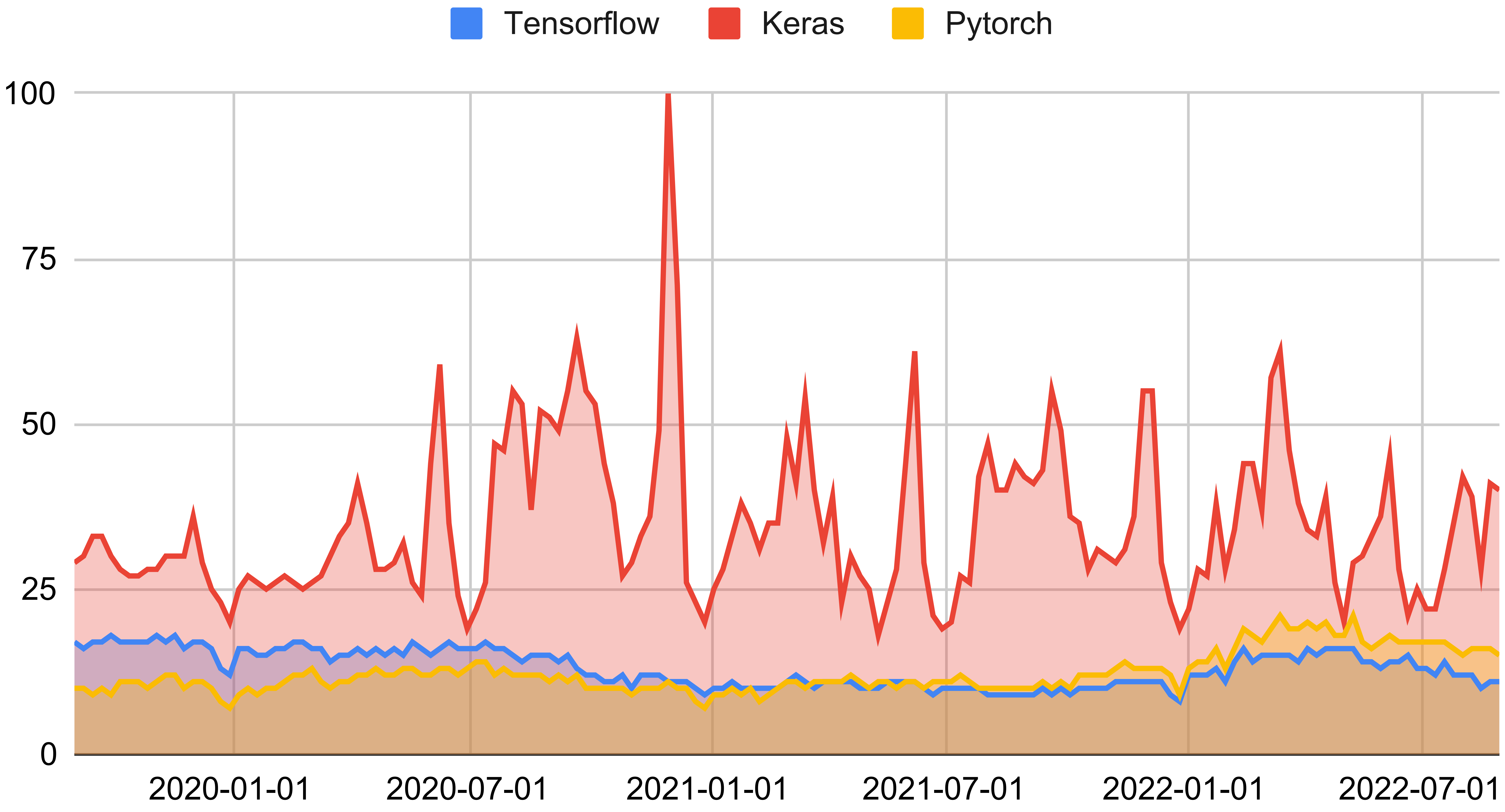
An open-source software library for machine learning (esp. deep learning) and artificial intelligence. It's developed by Google in 2015

Keras

An open-source software library that provides Python interfaces to TensorFlow. It's developed in 2015 by **François Chollet**, a Google engineer and suitable for fast experimentation with deep neural networks

PyTorch

An open-source software library for machine learning (esp. deep learning). It's developed by Facebook in 2016 and provides a Python interface to Torch.



Worldwide Popularity of TensorFlow vs. Keras vs. PyTorch (2019–2022)

Source: Google Trends

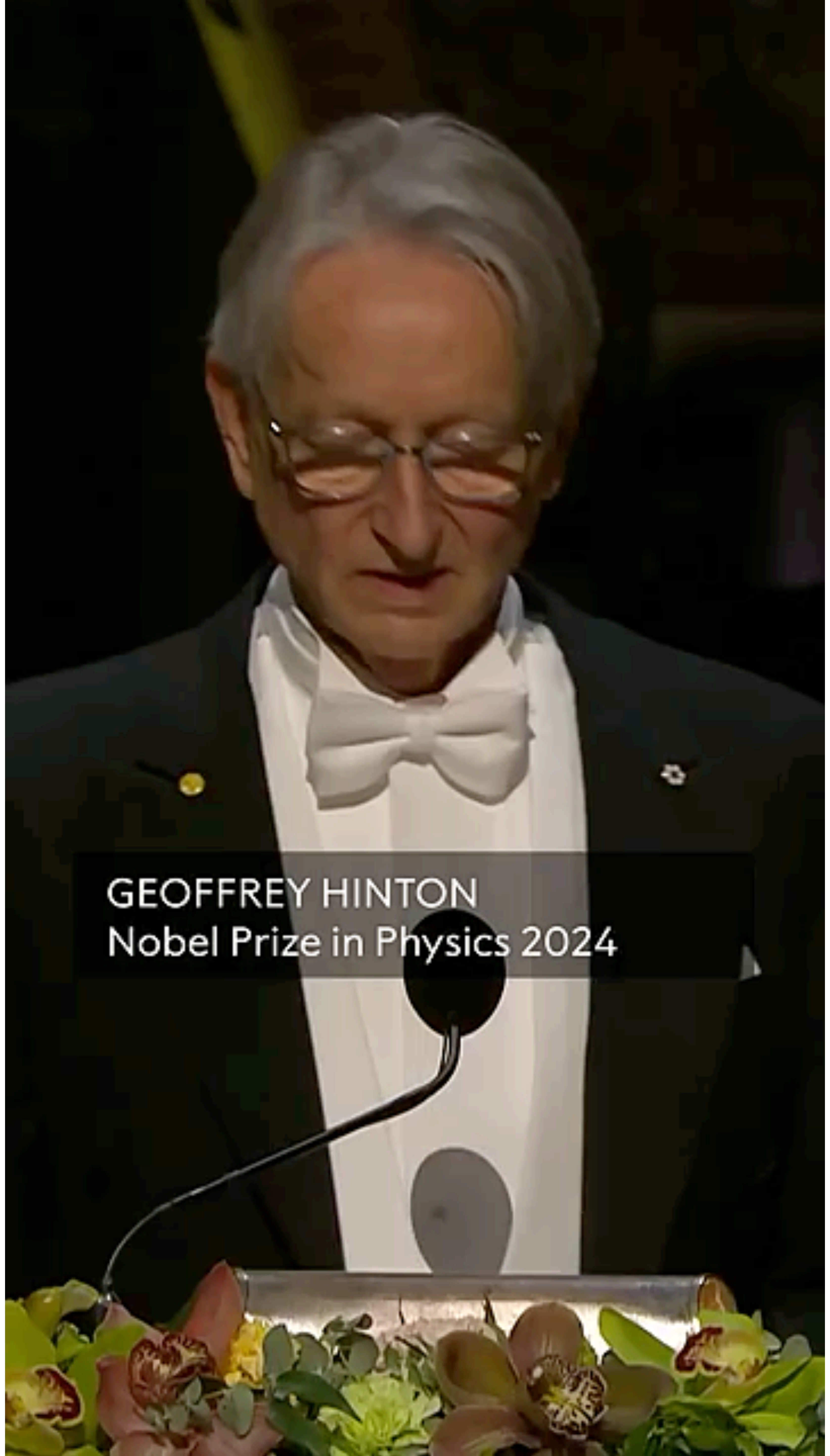
Feedforward Neuron Network in Keras

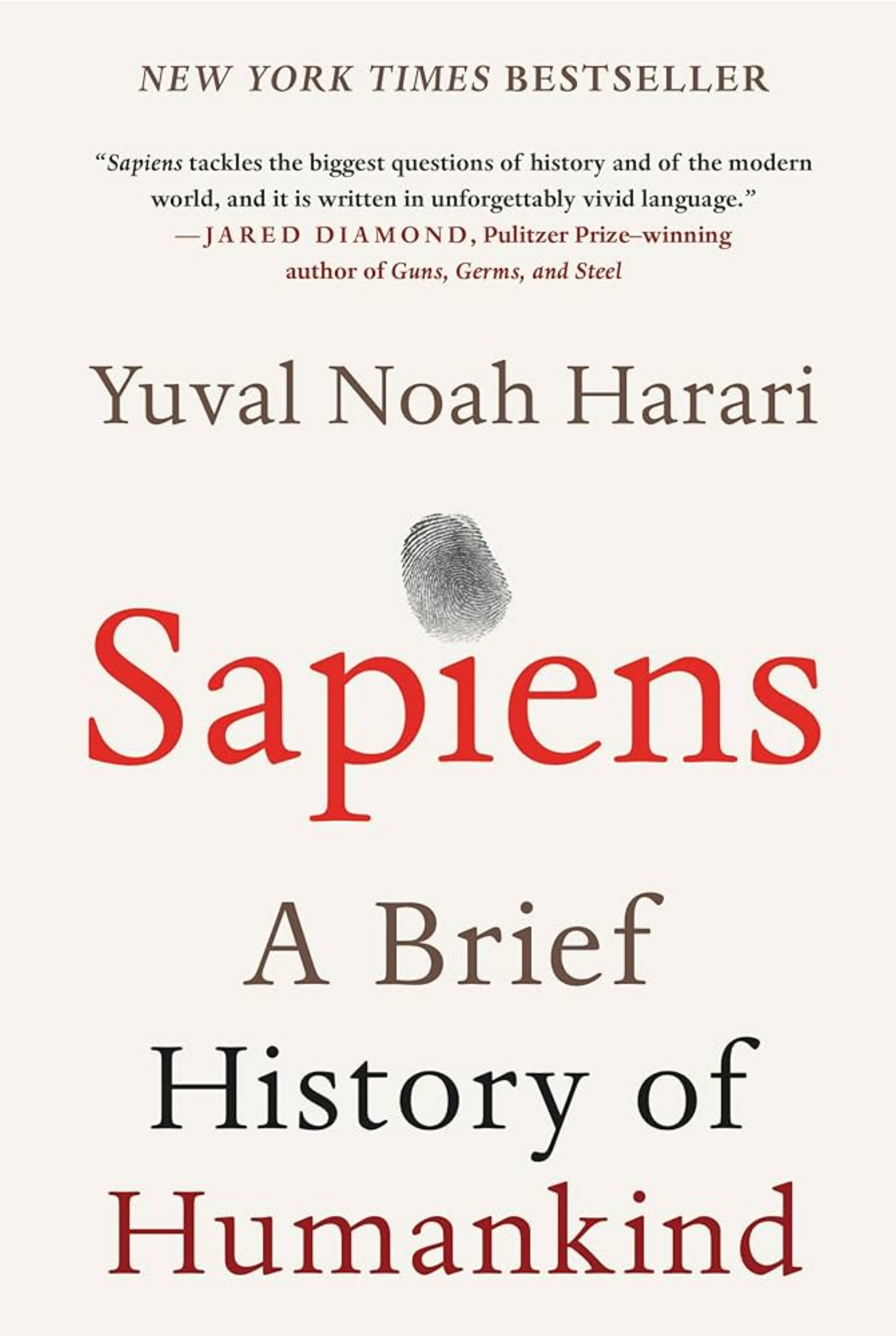
```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers

model=keras.Sequential([
    layers.Dense(512,  activation = "relu"),
    layers.Dense(10, activation = "softmax")
])
```

An Initial Taste of Deep Learning Using Python:

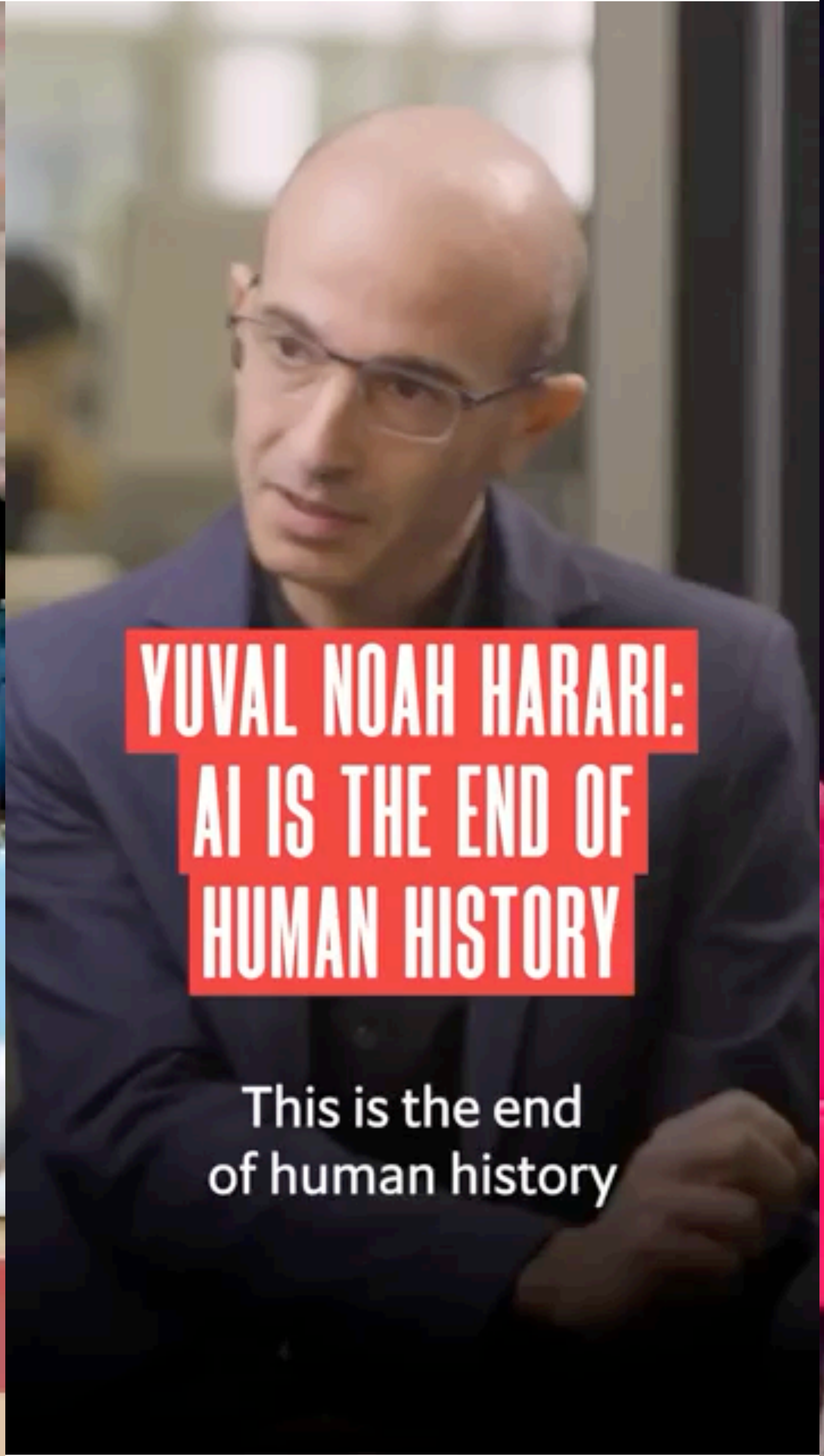
https://bit.ly/aisession3

We are Doomed...


GEOFFREY HINTON
Nobel Prize in Physics 2024


NEW YORK TIMES BESTSELLER
"Sapiens tackles the biggest questions of history and of the modern world, and it is written in unforgettably vivid language."
—JARED DIAMOND, Pulitzer Prize–winning author of Guns, Germs, and Steel
Yuval Noah Harari
Sapiens
A Brief History of Humankind


The AI Revolution


YUVAL NOAH HARARI:
AI IS THE END OF
HUMAN HISTORY
This is the end
of human history


AI HAS FEELINGS

Until Next Class

- **Group Assignment #1** due one hour before Session 3

- TA Tutorial on **Python Basics**
  - 🗓️ Friday, 10/31, 12:00–1:00 PM
  - 📍 Join via Zoom: https://bit.ly/jhuaita25
  - Attendance is optional. Materials will be posted on Canvas

- Refer to Syllabus for **Session 3 Readings**

Thank You!