

# Artificial Intelligence

ACTL3143 & ACTL5111 Deep Learning for Actuaries  
Patrick Laub



# Lecture Outline

- **Artificial Intelligence**
- Deep Learning Successes (Images)
- Deep Learning Successes (Text)
- Classifying Machine Learning Tasks
- Neural Networks



# Different goals of AI

Artificial intelligence describes an agent which is capable of:

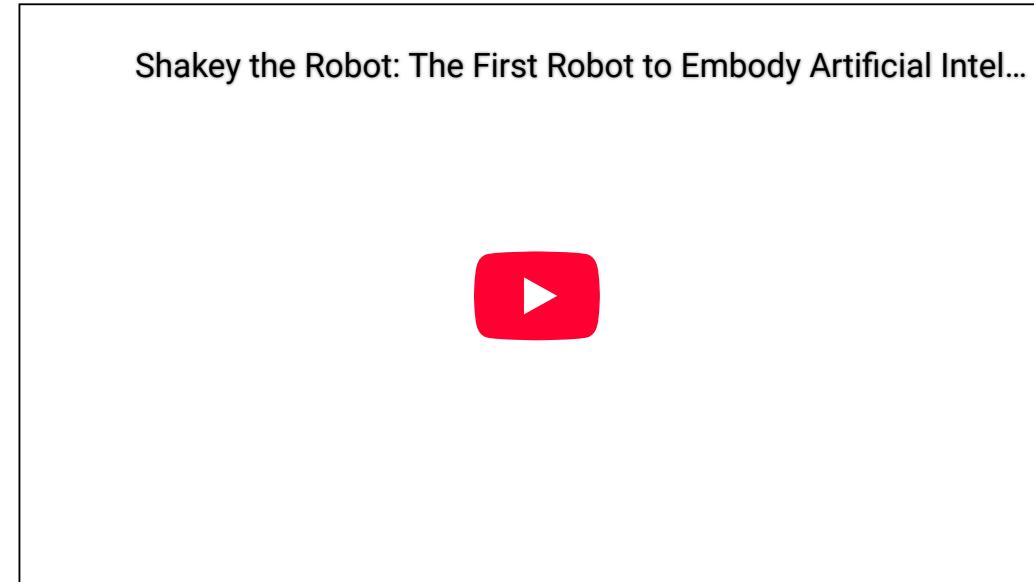
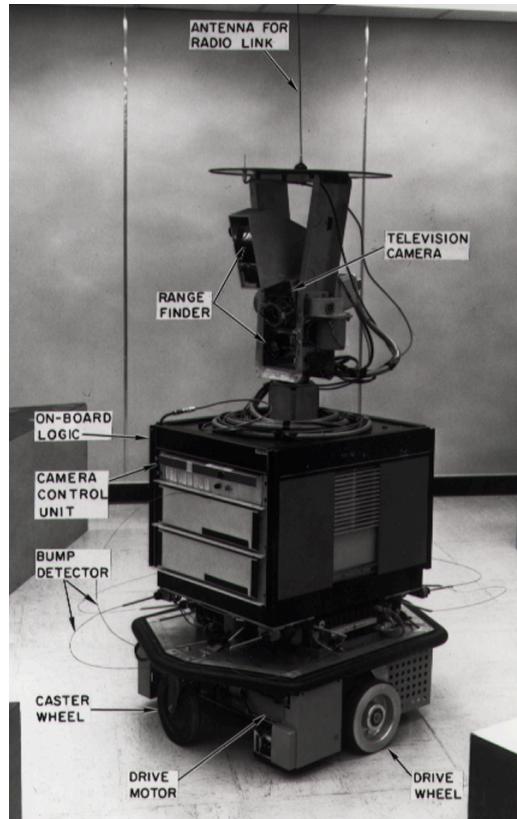
Thinking humanly	Thinking rationally
Acting humanly	Acting rationally

AI eventually become dominated by one approach, called *machine learning*, which itself is now dominated by *deep learning* (neural networks).

There are AI algorithms for simple tasks that don't use machine learning though.



# Shakey the Robot (~1966 – 1972)



Shakey the Robot



Source: Wikipedia page for the Shakey Project



UNSW  
SYDNEY

# Route-finding I

At its core, a pathfinding method searches a graph by starting at one vertex and exploring adjacent nodes until the destination node is reached, generally with the intent of finding the cheapest route. Although graph searching methods such as a breadth-first search would find a route if given enough time, other methods, which “explore” the graph, would tend to reach the destination sooner. An analogy would be a person walking across a room; rather than examining every possible route in advance, the person would generally walk in the direction of the destination and only deviate from the path to avoid an obstruction, and make deviations as minor as possible. (Source: [Wikipedia](#))



A\* algorithm (1968).

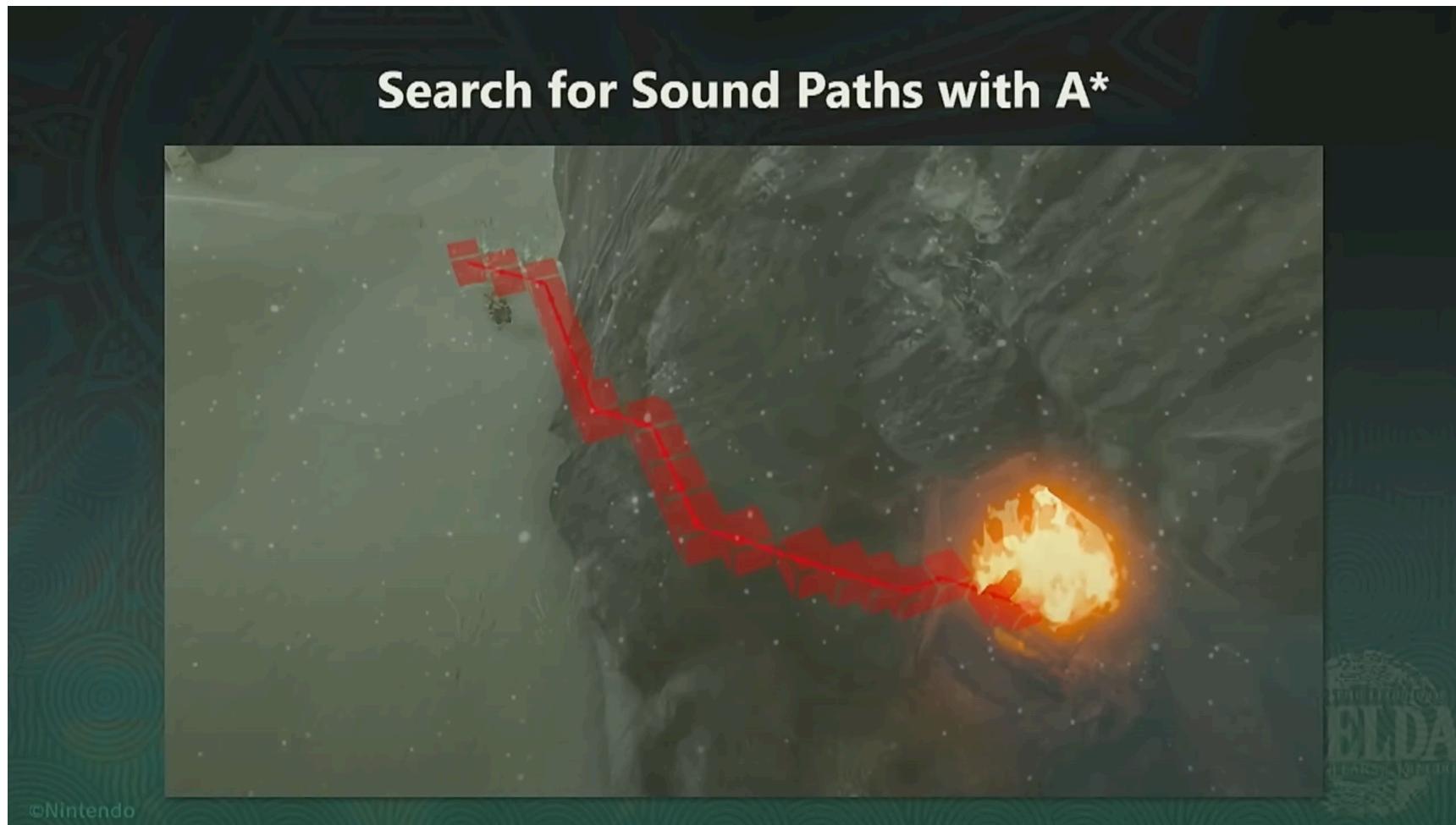


Source: Wikipedia page for [the A\\* search algorithm](#).



UNSW  
SYDNEY

# Route-finding II

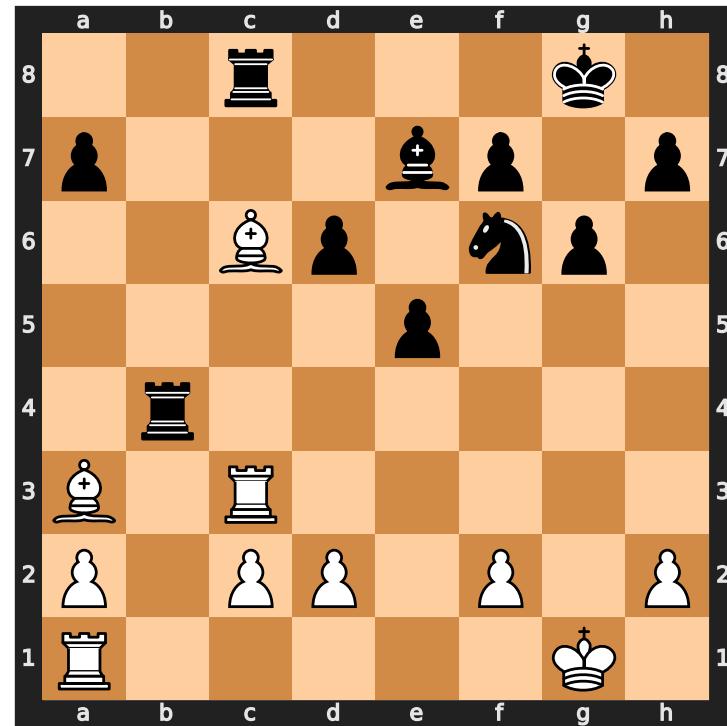


Tunes of the Kingdom: Evolving Physics and Sounds for 'The Legend of Zelda: Tears of the Kingdom', GDC 2024



# Evaluating a chess game I

Who's winning this game?

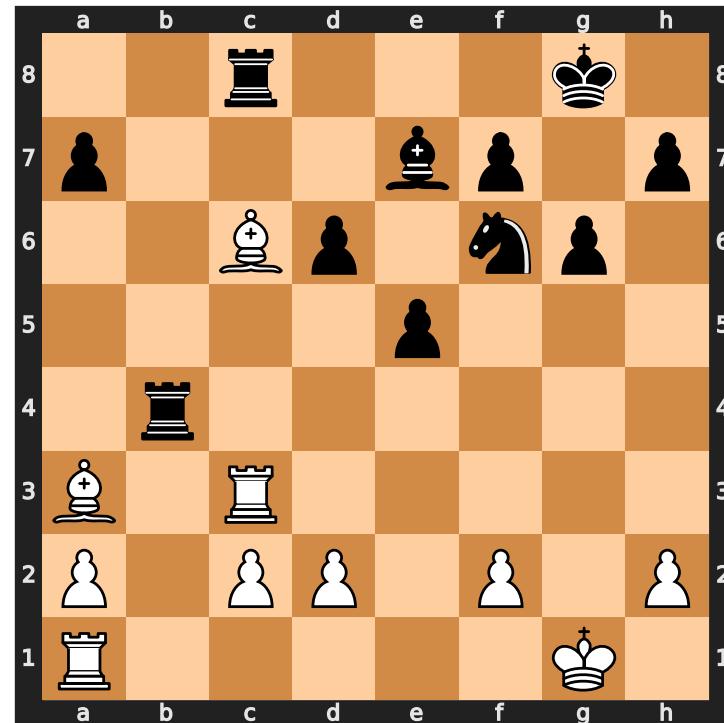


	$5 \times 1 = 5$
	$0 \times 3 = 0$
	$2 \times 3 = 6$
	$2 \times 5 = 10$
	$0 \times 9 = 0$
	$1 \times 0 = 0$

White      21

# Evaluating a chess game II

Just add up the pieces for each player.



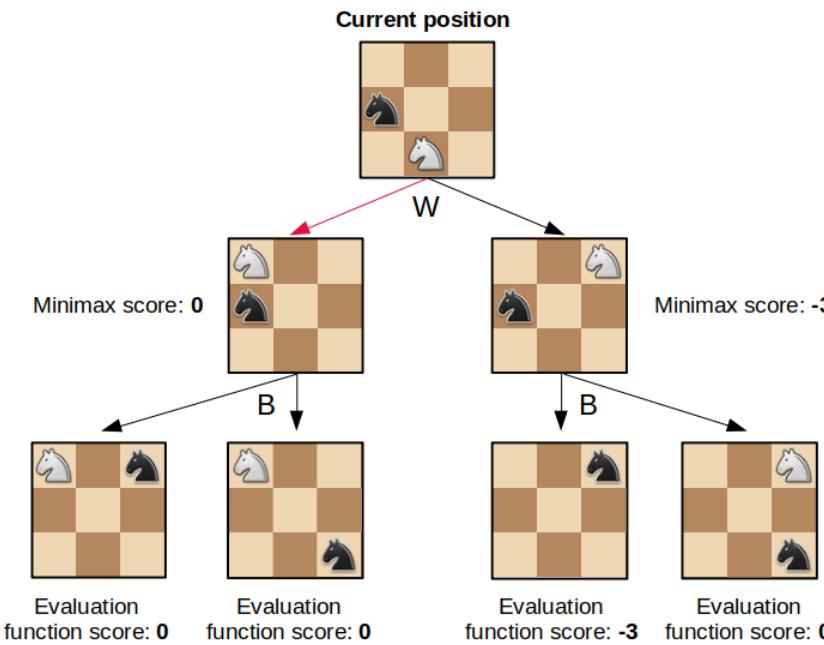
	$6 \times 1 = 6$
	$1 \times 3 = 3$
	$1 \times 3 = 3$
	$2 \times 5 = 10$
	$0 \times 9 = 0$
	$1 \times 0 = 0$

Black      22

Overall:  $21 - 22 = -1$ .



# The minimax algorithm



```

function minimax(position, depth, maximizingPlayer)
  if depth == 0 or game over in position
    return static evaluation of position

  if maximizingPlayer
    maxEval = -infinity
    for each child of position
      eval = minimax(child, depth - 1, false)
      maxEval = max(maxEval, eval)
    return maxEval

  else
    minEval = +infinity
    for each child of position
      eval = minimax(child, depth - 1, true)
      minEval = min(minEval, eval)
    return minEval
  
```

Pseudocode for the minimax algorithm.

The minimax algorithm for chess.

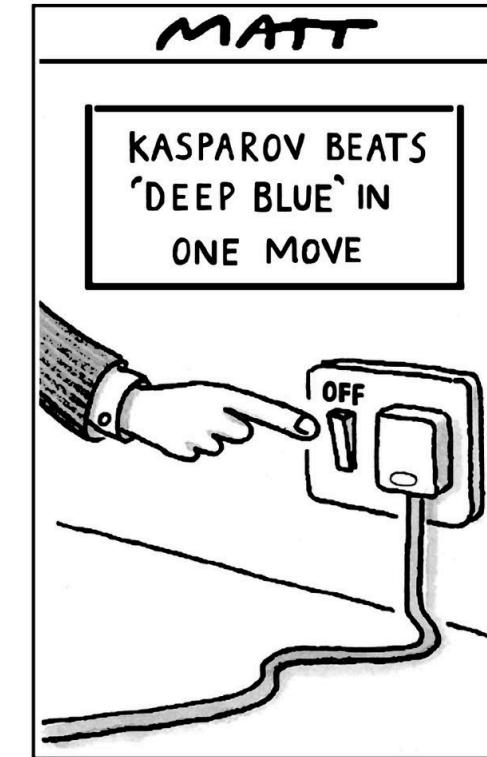


# Chess

Deep Blue (1997)



Gary Kasparov playing Deep Blue.

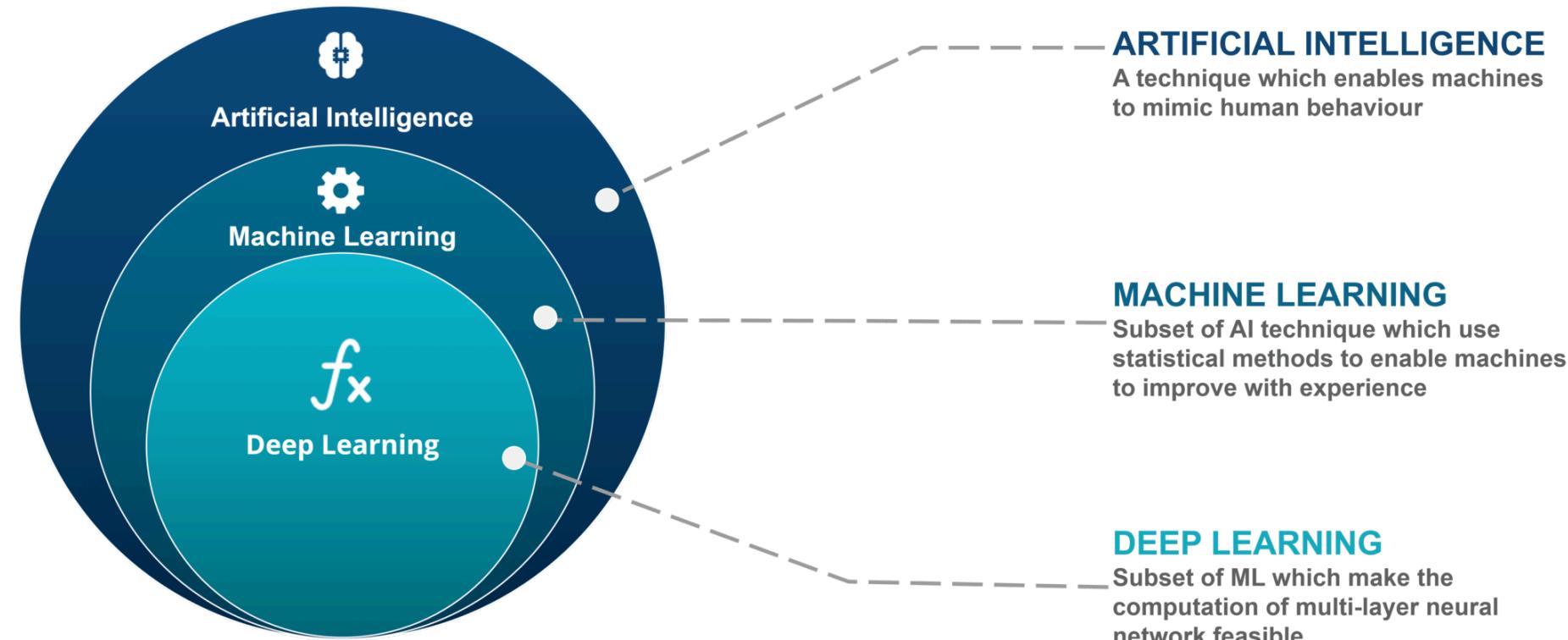


Cartoon of the match.

# Machine Learning

Tried *making a computer smart*, too hard!

Make a computer that can **learn** to be smart.



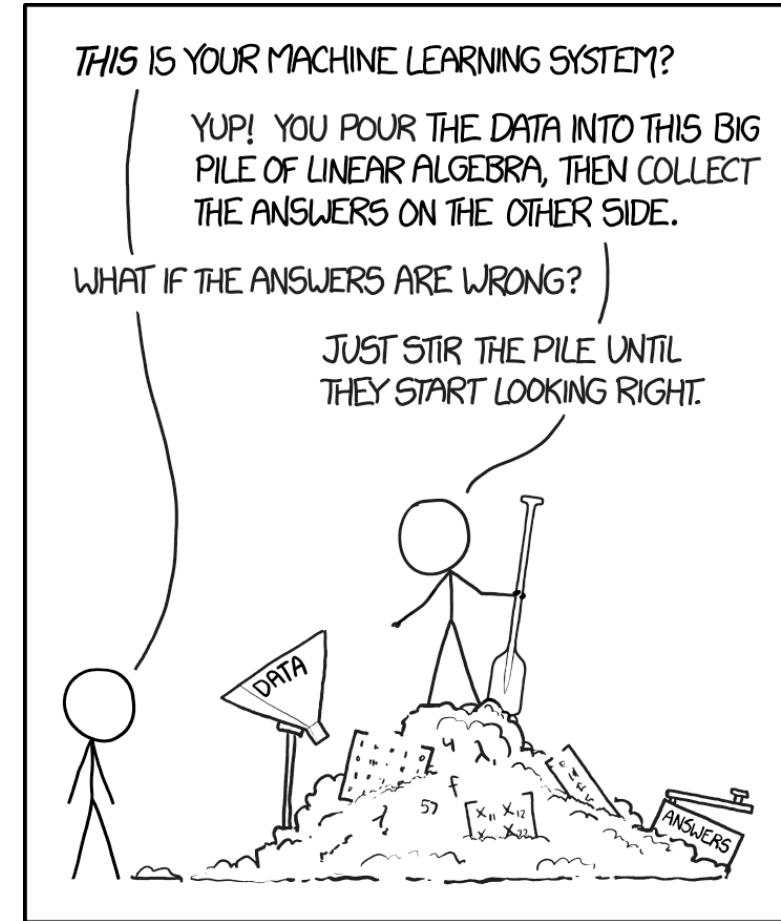
The Venn diagram of Artificial Intelligence, Machine Learning, and Deep Learning.



Source: Edureka (2020), [AI Vs Machine Learning Vs Deep Learning Edureka](#).

# Definition

“[Machine Learning is the] field of study that gives computers the ability to learn without being explicitly programmed” Arthur Samuel (1959)



# Lecture Outline

- Artificial Intelligence
- **Deep Learning Successes (Images)**
- Deep Learning Successes (Text)
- Classifying Machine Learning Tasks
- Neural Networks



# Image Classification I

What is this?



Options:

1. punching bag
2. goblet
3. red wine
4. hourglass
5. balloon

 **Note**

Hover over the options to see AI's prediction  
(i.e. the probability of the photo being in that category).

Source: [Wikipedia](#)



UNSW  
SYDNEY

# Image Classification II

What is this?



Options:

1. sea urchin
2. porcupine
3. echidna
4. platypus
5. quill

# Image Classification III

What is this?

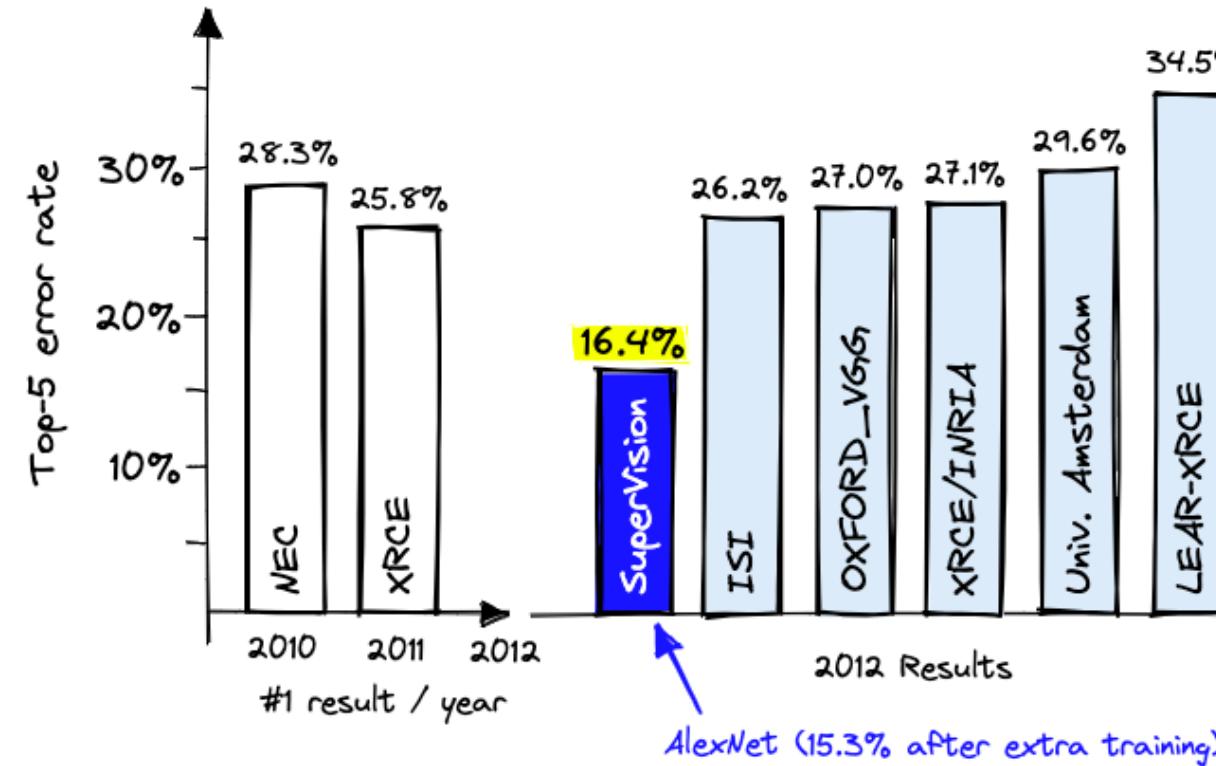


Options:

1. dingo
2. malinois
3. German shepherd
4. muzzle
5. kelpie

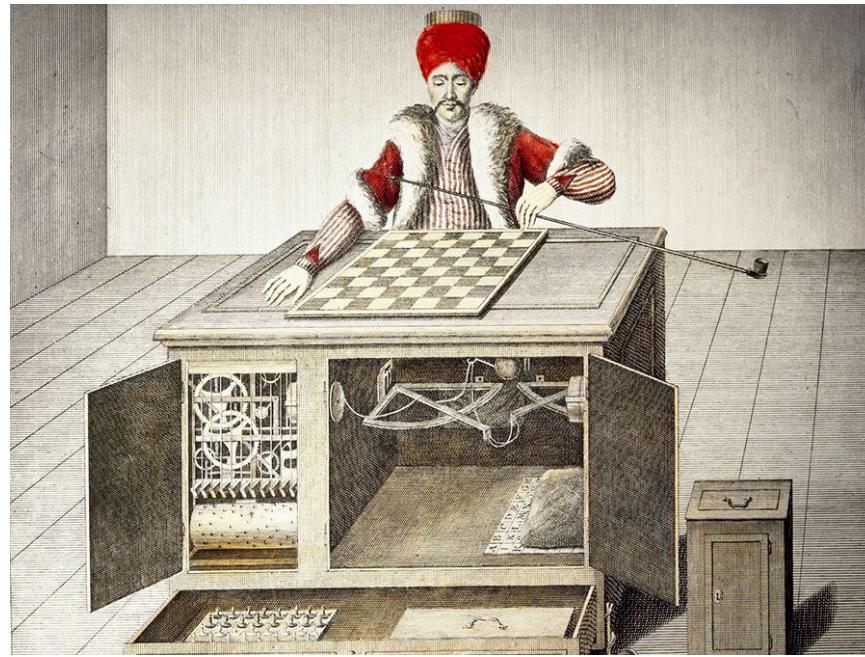
# ImageNet Challenge

**ImageNet** and the *ImageNet Large Scale Visual Recognition Challenge (ILSVRC)*; originally **1,000 synsets**.



AlexNet — a neural network developed by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton — won the ILSVRC 2012 challenge convincingly.

# How were the images labelled?



The original ‘mechanical turk’ (1770)

“Two years later, the first version of ImageNet was released with 12 million images structured and labeled in line with the WordNet ontology. If one person had annotated one image/minute and did nothing else in those two years (including sleeping or eating), it would have taken 22 years and 10 months.

To do this in under two years, Li turned to Amazon Mechanical Turk, a crowdsourcing platform where anyone can hire people from around the globe to perform tasks cost-effectively.”



James Briggs & Laura Carnevali, *AlexNet and ImageNet: The Birth of Deep Learning*, Embedding Methods for Image Search, Pinecone Blog



UNSW  
SYDNEY

# Needed a graphics card

A graphics processing unit (GPU)



My deep learning PC

**“4.2. Training on multiple GPUs**

A single GTX 580 GPU has only 3GB of memory, which limits the maximum size of the networks that can be trained on it. It turns out that 1.2 million training examples are enough to train networks which are too big to fit on one GPU. Therefore we spread the net across two GPUs.”



Source: Krizhevsky, Sutskever and Hinton (2017), *ImageNet Classification with Deep Convolutional Neural Networks*, Communications of the ACM



UNSW  
SYDNEY

# Lee Sedol plays AlphaGo (2016)

Deep Blue was a win for AI, AlphaGo a win for ML.



Lee Sedol playing AlphaGo AI

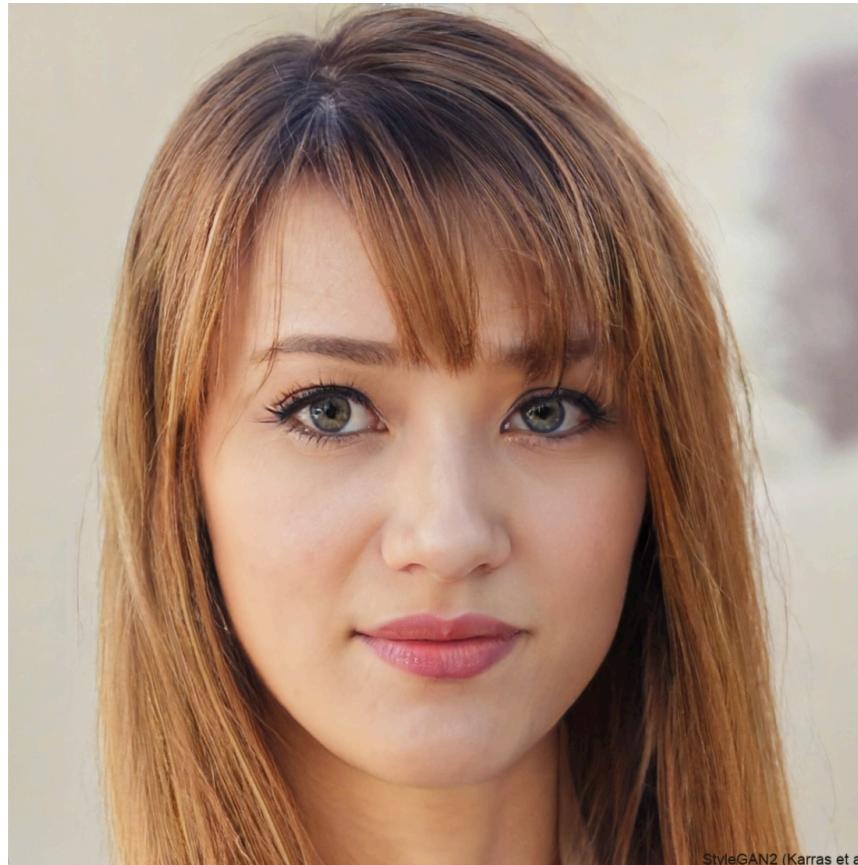
I highly recommend **this documentary about the event**.



Source: Patrick House (2016), [AlphaGo, Lee Sedol, and the Reassuring Future of Humans and Machines](#), New Yorker article.

# Generative Adversarial Networks (2014)

<https://thispersondoesnotexist.com/>



StyleGAN2 (Karras et al.)

A GAN-generated face



StyleGAN2 (Karras et al.)

A GAN-generated face

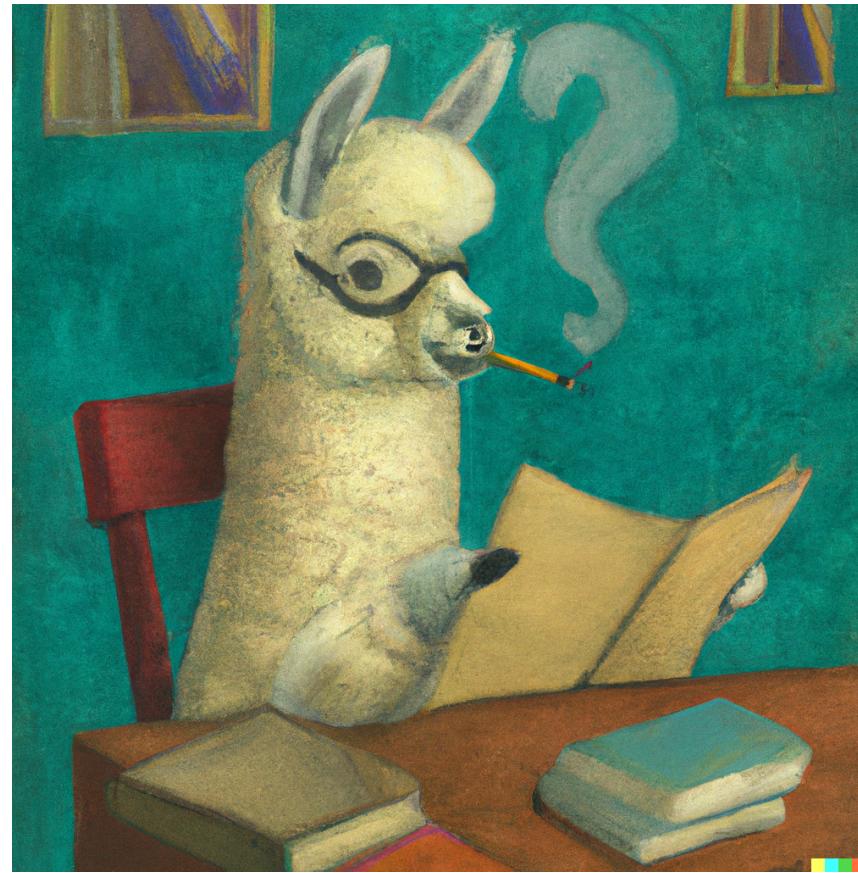


UNSW  
SYDNEY

# Diffusion models



Painting of avocado skating while wearing a hoodie



A surrealist painting of an alpaca studying for an exam



Source: Dall-E 2 images, prompts by ACTL3143 students in 2022.



# Dall-E 2 (2022) vs Dall-E 3 (2023)

Same prompt: “A beautiful calm photorealistic view of an waterside metropolis that has been neglected for hundreds of years and is overgrown with nature”



Dall-E 3 rewrites it as: “Photo of a once-majestic metropolis by the water, now <sup>Dall-E<sup>3</sup></sup> abandoned for centuries. The city's skyscrapers and buildings are cloaked in <sup>Dall-E<sup>3</sup></sup> thick green vines...”



# Lecture Outline

- Artificial Intelligence
- Deep Learning Successes (Images)
- **Deep Learning Successes (Text)**
- Classifying Machine Learning Tasks
- Neural Networks



# GPT

ChatGPT 4.0



Write code to demonstrate the ImageNet problem; i.e. show an image from the test set, and show the top 5 categories which it could be, according to some pre-trained model. Do this for some simple category (like a photo of a balloon or something) and a really difficult category (some specific obscure animal)



To demonstrate the ImageNet problem, we'll use a pre-trained deep learning model, such as ResNet, which is commonly used for image classification tasks. We'll use the PyTorch library to handle the model and display the predictions.

Here is the code to demonstrate this:

```
python
import torch
from torchvision import models, transforms
from PIL import Image
import requests
from io import BytesIO
import matplotlib.pyplot as plt

# Load pre-trained ResNet model
model = models.resnet50(pretrained=True)
model.eval()
```



Message ChatGPT



AI predictions in the classification demo were from GPT code.



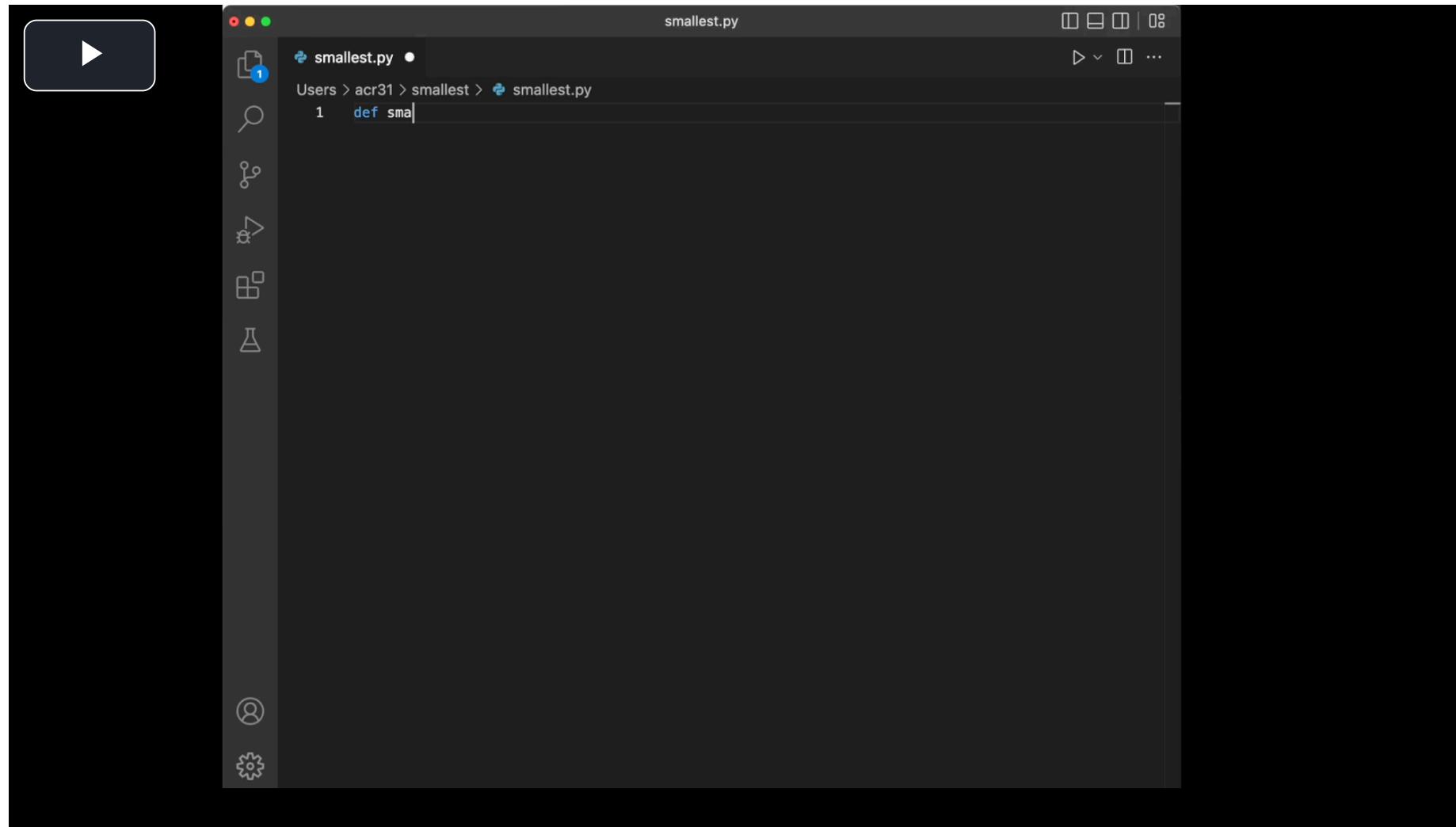
Source: [ChatGPT conversation](#).

## Homework Get ChatGPT to:

- generate images
- translate code
- explain code
- run code
- analyse a dataset
- critique code
- critique writing
- voice chat with you

Compare to Copilot.

# Code generation (GitHub Copilot)

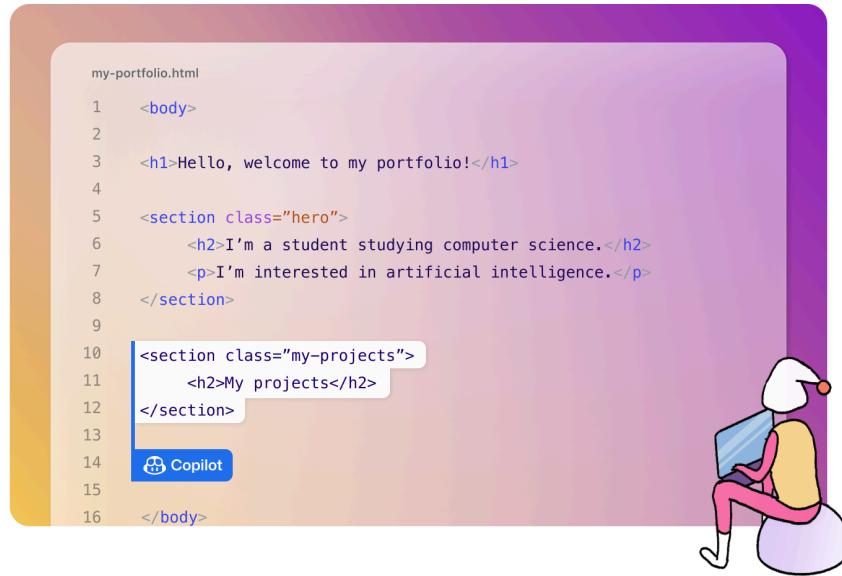


Source: GitHub Blog



UNSW  
SYDNEY

# Students get Copilot for free



Use a free trial then sign up for free education account

A student post from last year:

I strongly recommend taking a photo holding up your Academic Statement to your phone's front facing camera when getting verified for the student account on GitHub. No other method of taking/uploading photo proofs worked for me. Furthermore, I had to make sure the name on the statement matched my profile exactly and also had to put in a bio.

Good luck with this potentially annoying process!

**Homework** It's a slow process, so get this going early.

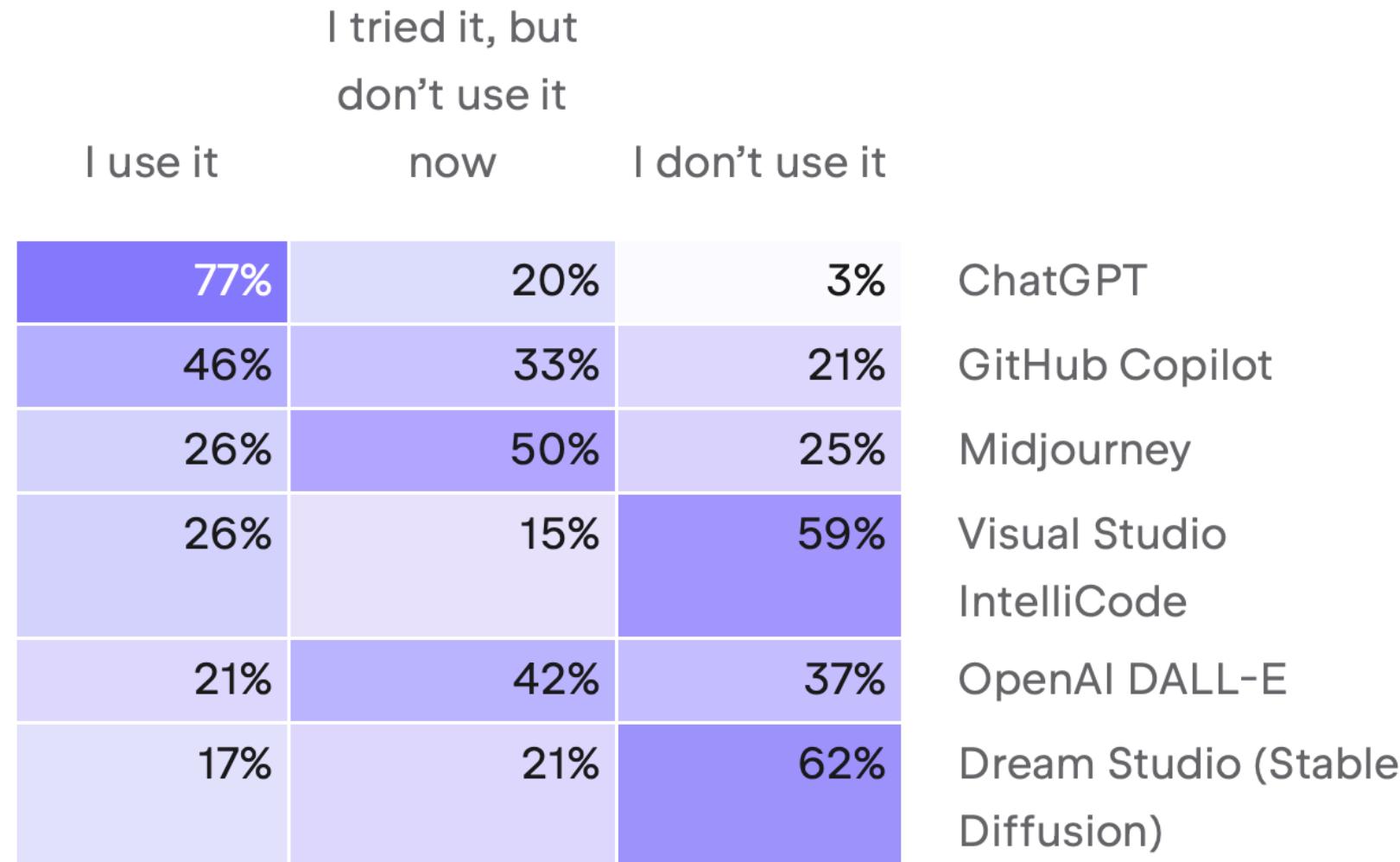


Source: [GitHub Education for Students](#)



UNSW  
SYDNEY

# Programmers are increasingly using AI



Question: What is your experience with the following AI tools?

Source: JetBrains, [The State of Developer Ecosystem 2023](#).

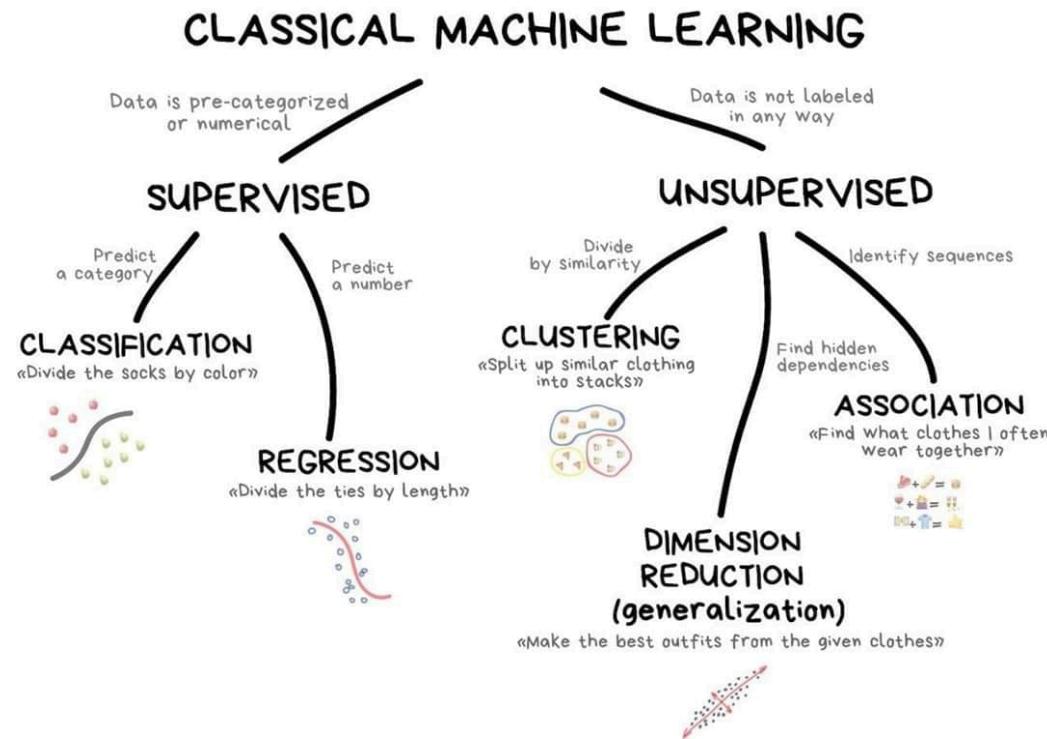


# Lecture Outline

- Artificial Intelligence
- Deep Learning Successes (Images)
- Deep Learning Successes (Text)
- **Classifying Machine Learning Tasks**
- Neural Networks



# A taxonomy of problems



Machine learning categories in ACTL3142.

New ones:

- Reinforcement learning
- Semi-supervised learning
- Active learning



Source: Kaggle, [Getting Started](#).

# Supervised learning

The main focus of this course.

## Regression

- Given policy  $\hookrightarrow$  predict the rate of claims.
- Given policy  $\hookrightarrow$  predict claim severity.
- Given a reserving triangle  $\hookrightarrow$  predict future claims.

## Classification

- Given a claim  $\hookrightarrow$  classify as fraudulent or not.
- Given a customer  $\hookrightarrow$  predict customer retention patterns.



# Supervised learning: mathematically

## Background

1. Given training data:

$$\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$$

2. Choose each of these:

- Decision function

$$\hat{\mathbf{y}} = f_{\boldsymbol{\theta}}(\mathbf{x}_i)$$

- Loss function

$$\ell(\hat{\mathbf{y}}, \mathbf{y}_i) \in \mathbb{R}$$

## A Recipe for Machine Learning

3. Define goal:

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta}} \sum_{i=1}^N \ell(f_{\boldsymbol{\theta}}(\mathbf{x}_i), \mathbf{y}_i)$$

4. Train with SGD:

(take small steps  
opposite the gradient)

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta_t \nabla \ell(f_{\boldsymbol{\theta}}(\mathbf{x}_i), \mathbf{y}_i)$$

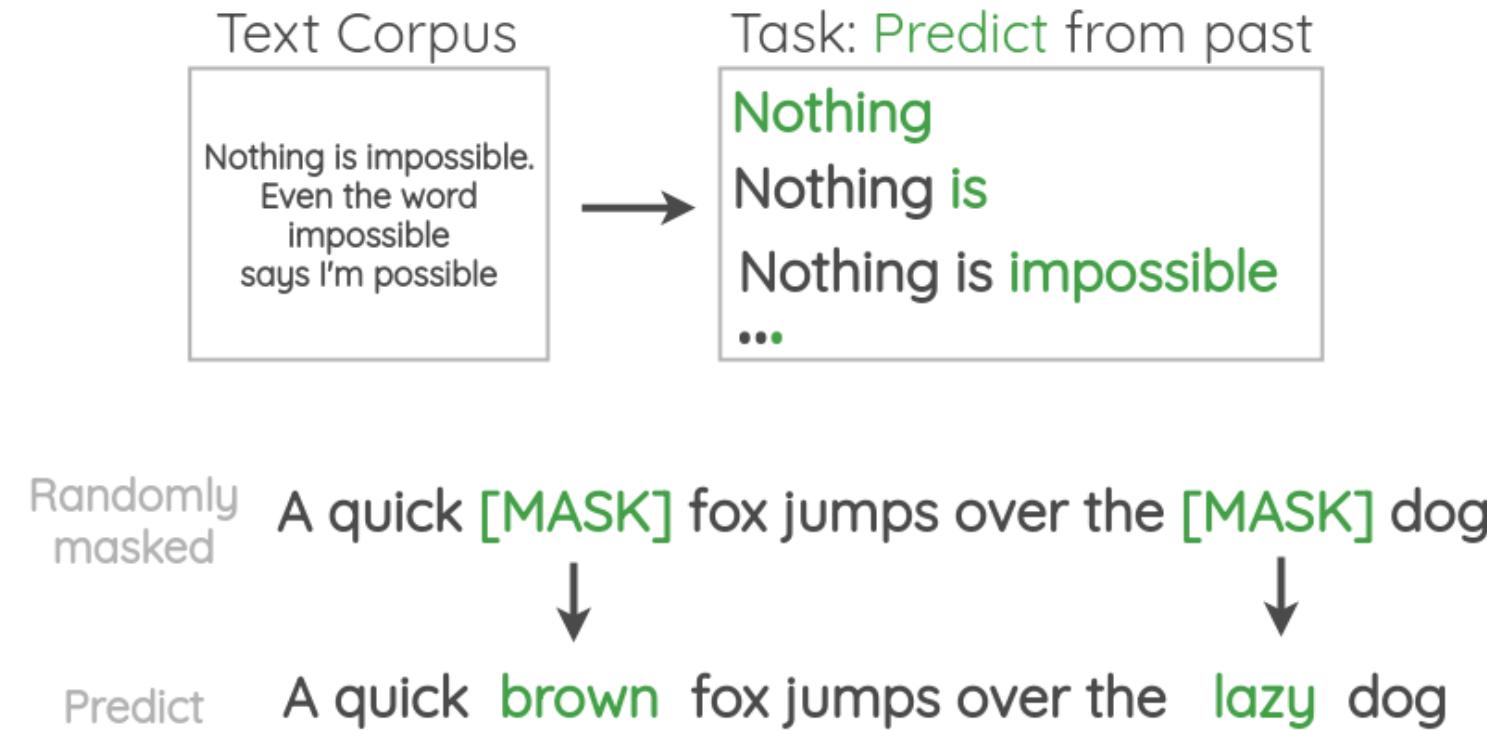
A recipe for supervised learning.



Source: Matthew Gormley (2021), [Introduction to Machine Learning Lecture Slides](#), Slide 67.

# Self-supervised learning

Data which ‘labels itself’. Example: language model.



‘Autoregressive’ (e.g. GPT) versus ‘masked’ model (e.g. BERT).



Source: Amit Chaudhary (2020), [Self Supervised Representation Learning in NLP](#).

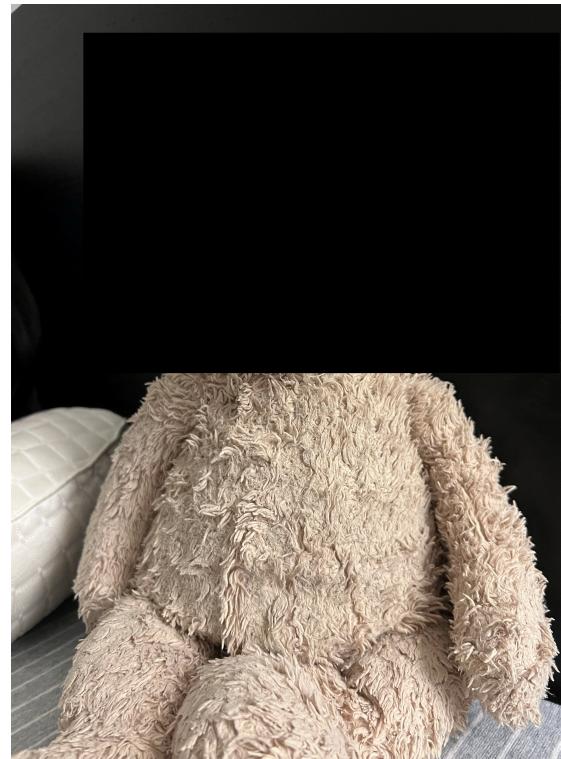


UNSW  
SYDNEY

# Example: image inpainting



Original image



Randomly remove a part



Try to fill it in from context

Other examples: image super-resolution, denoising images.



See Liu et al. (2018), [Image Inpainting for Irregular Holes using Partial Convolutions](#).



UNSW  
SYDNEY

# Example: Deoldify images #1



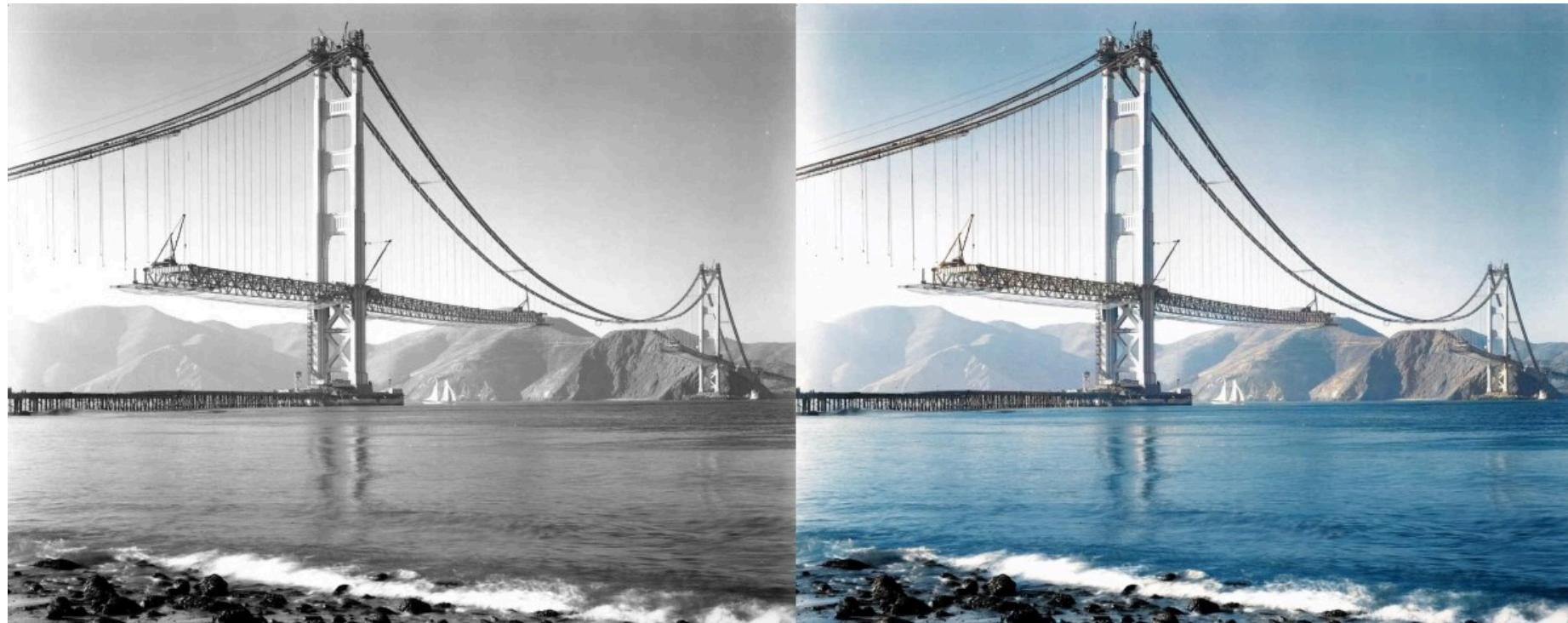
A deoldified version of the famous “Migrant Mother” photograph.

Source: [Deoldify package](#).



UNSW  
SYDNEY

# Example: Deoldify images #2



A deoldified Golden Gate Bridge under construction.



Source: [Deoldify package](#).



UNSW  
SYDNEY

# Lecture Outline

- Artificial Intelligence
- Deep Learning Successes (Images)
- Deep Learning Successes (Text)
- Classifying Machine Learning Tasks
- **Neural Networks**

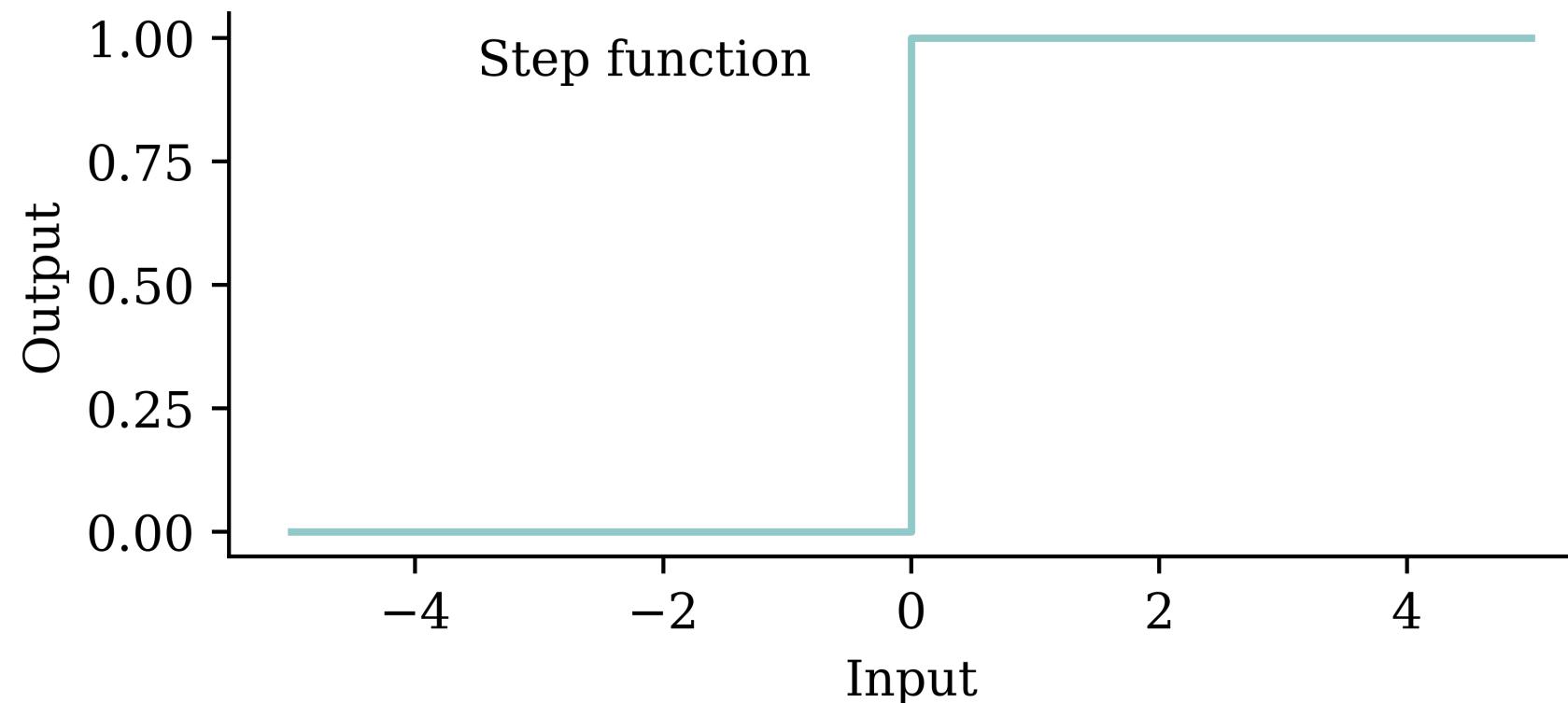


# How do real neurons work?

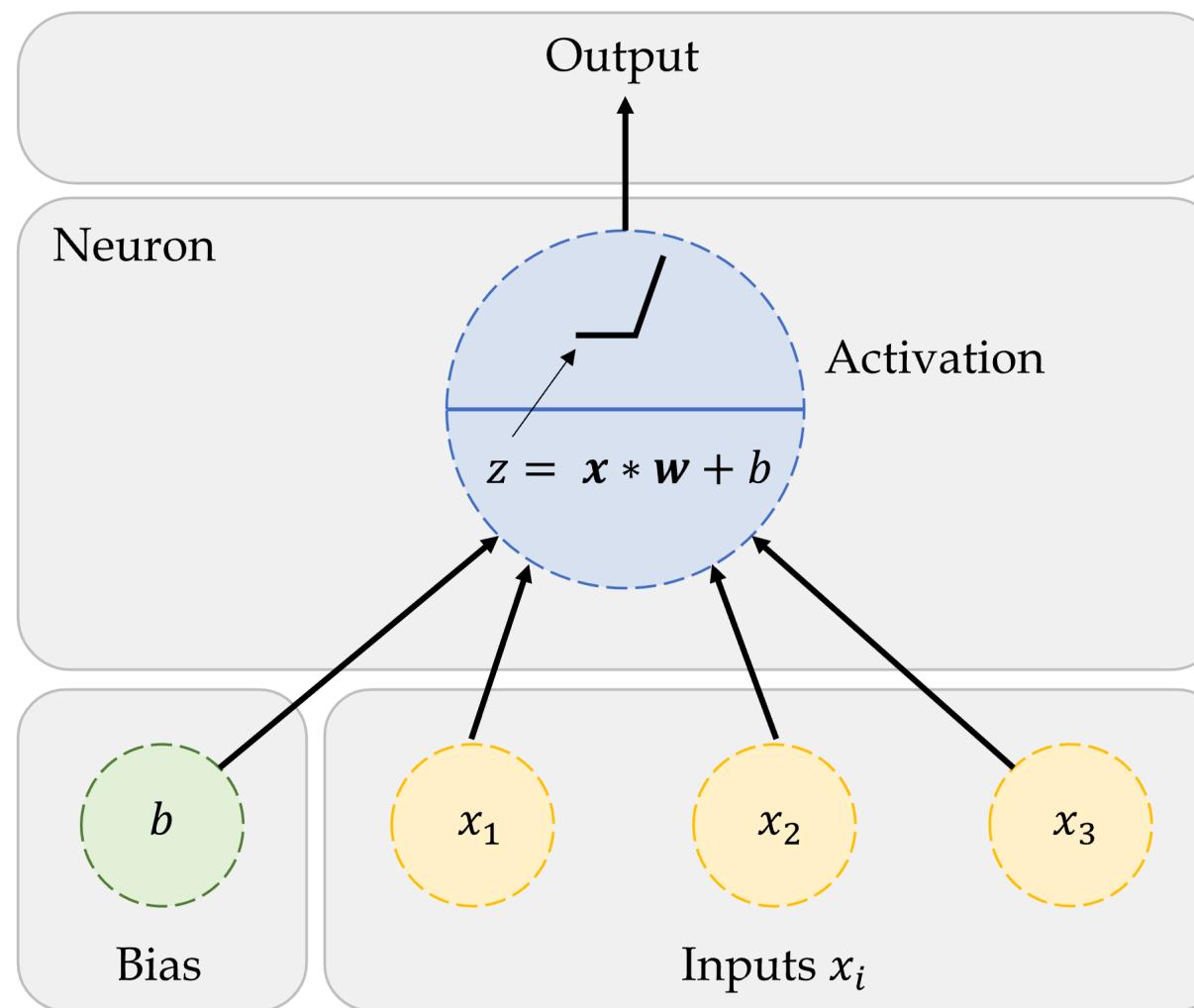
2-Minute Neuroscience: The Neuron



# A neuron ‘firing’



# An artificial neuron



A neuron in a neural network with a ReLU activation.

Source: Marcus Lautier (2022).



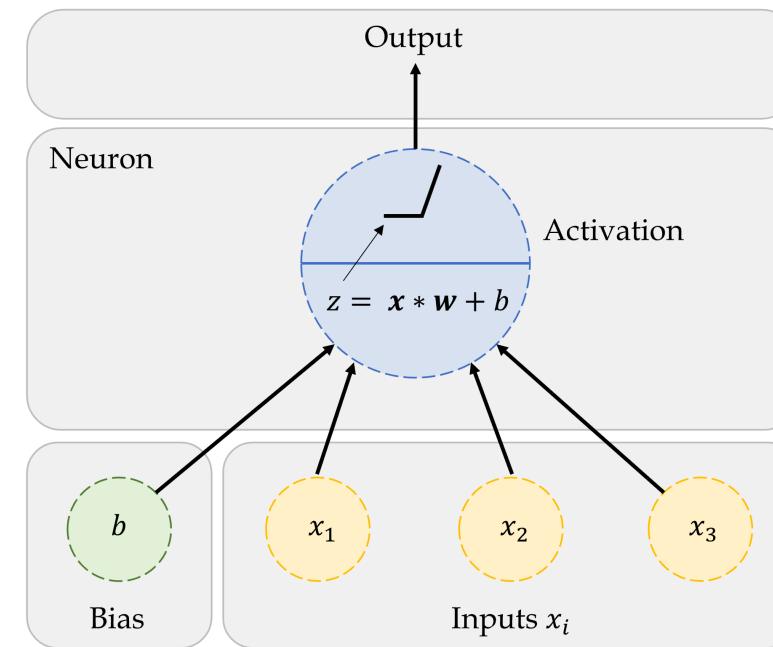
# One neuron

$$\begin{aligned} z &= x_1 \times w_1 + \\ &x_2 \times w_2 + \\ &x_3 \times w_3. \end{aligned}$$

$$a = \begin{cases} z & \text{if } z > 0 \\ 0 & \text{if } z \leq 0 \end{cases}$$

Here,  $x_1, x_2, x_3$  is just some fixed data.

The weights  $w_1, w_2, w_3$  should be ‘learned’.



A neuron in a neural network with a ReLU activation.

# One neuron with bias

$$z = x_1 \times w_1 +$$

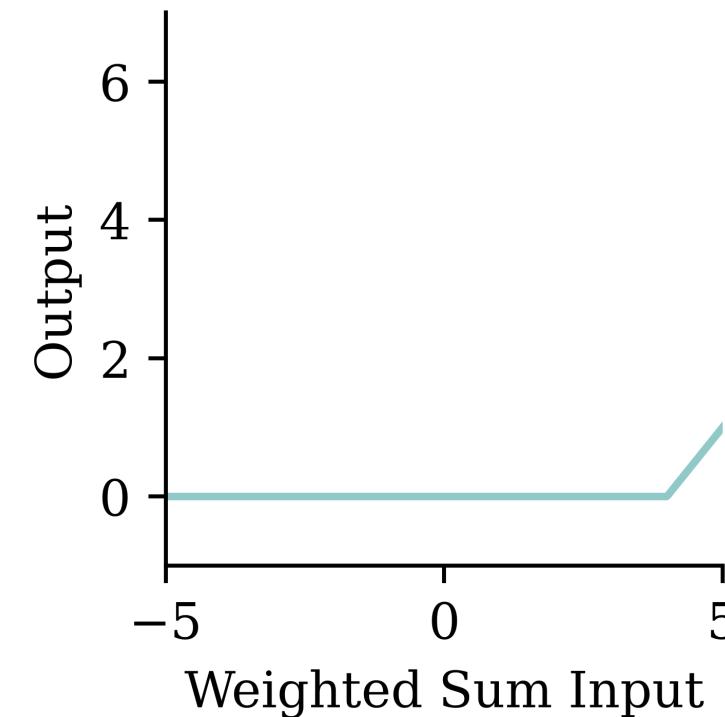
$$x_2 \times w_2 +$$

$$x_3 \times w_3 + b.$$

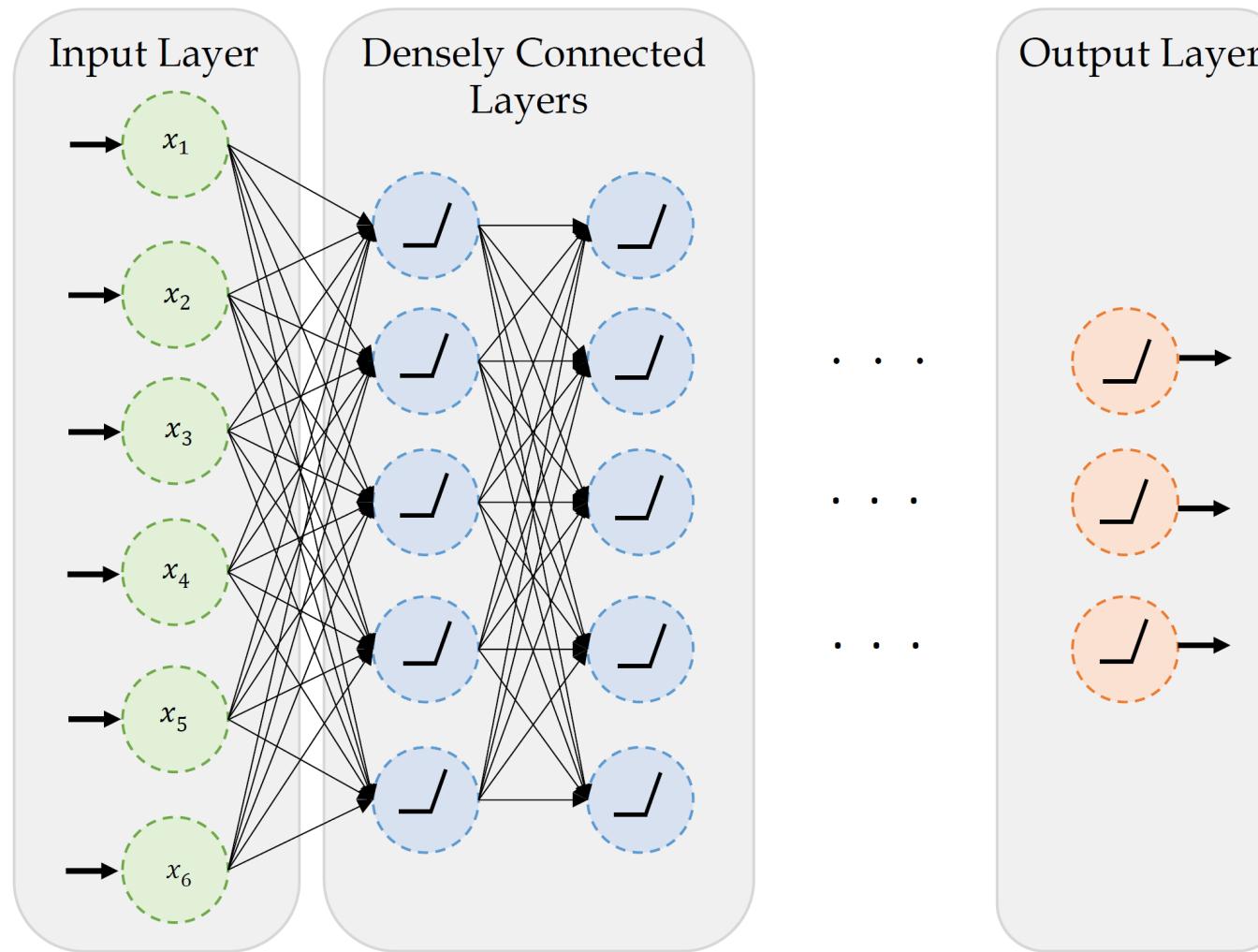
$$a = \begin{cases} z & \text{if } z > 0 \\ 0 & \text{if } z \leq 0 \end{cases}$$

The weights  $w_1, w_2, w_3$  and bias  $b$  should be ‘learned’.

Bias = -4	0	4
-----------	---	---



# A basic neural network



A basic fully-connected/dense network.



Source: Marcus Lautier (2022).

# Step-function activation

## Perceptrons

Brains and computers are binary, so make a perceptron with binary data. Seemed reasonable, impossible to train.

## Modern neural network

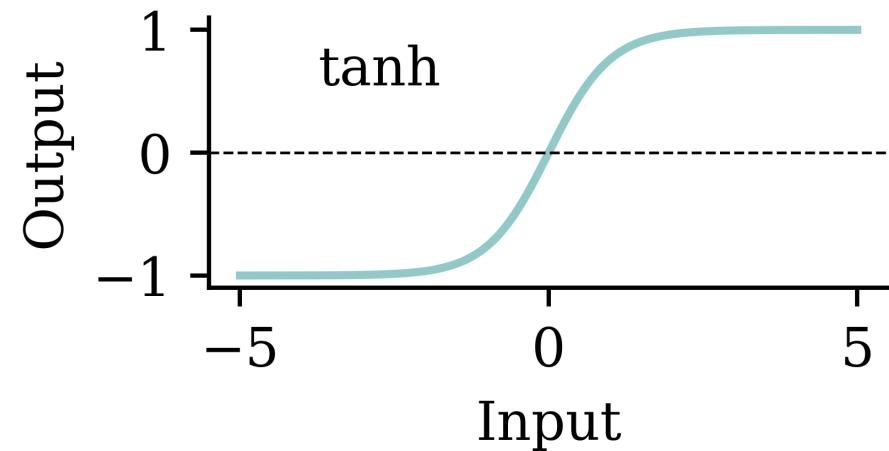
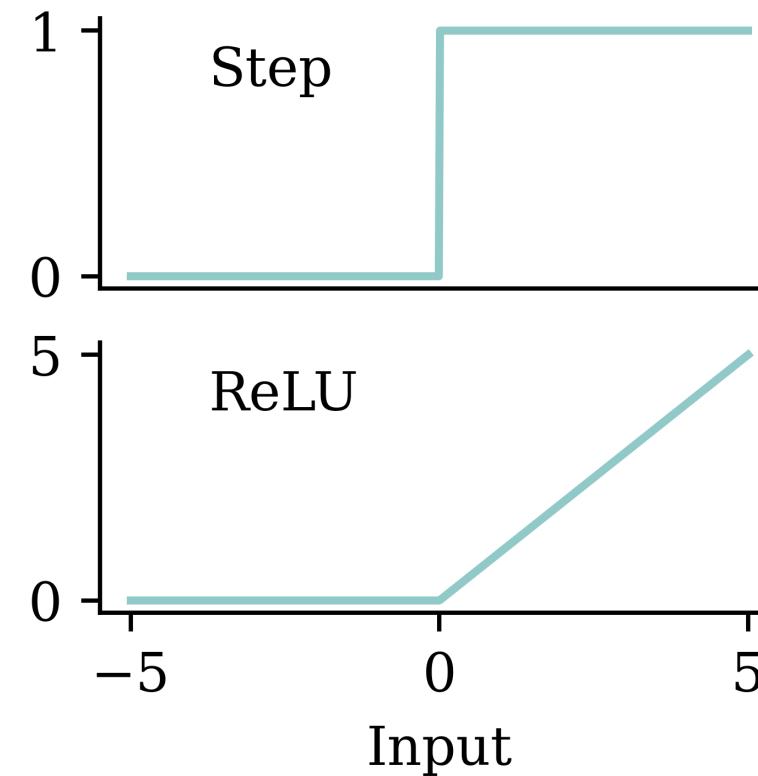
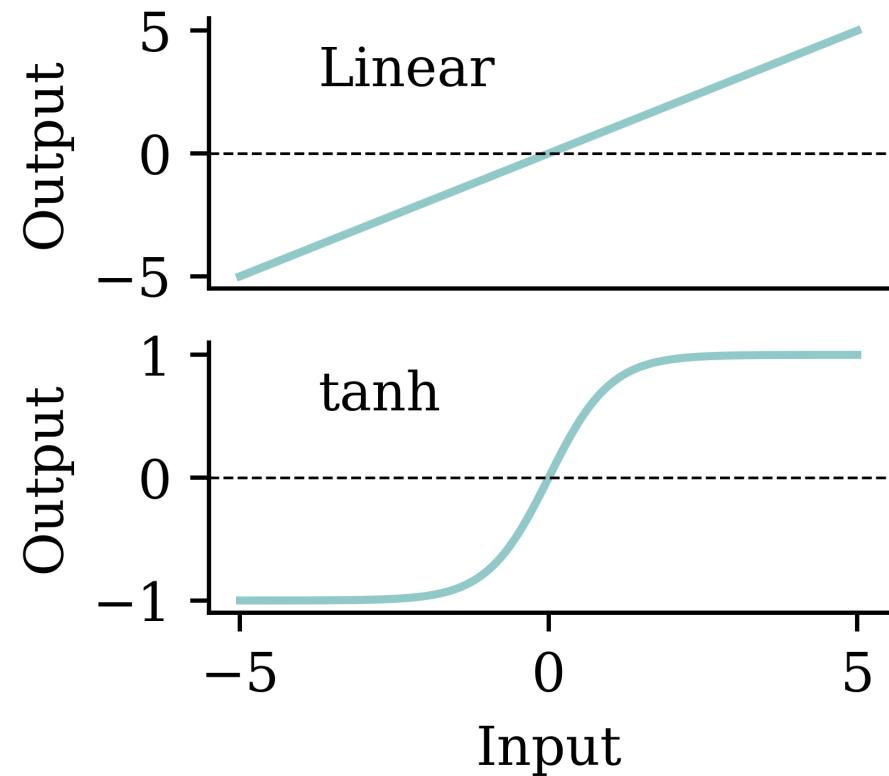
Replace binary state with continuous state. Still rather slow to train.

 Note

It's a **neural** network made of **neurons**, not a “neuron network”.



# Try different activation functions



# Flexible

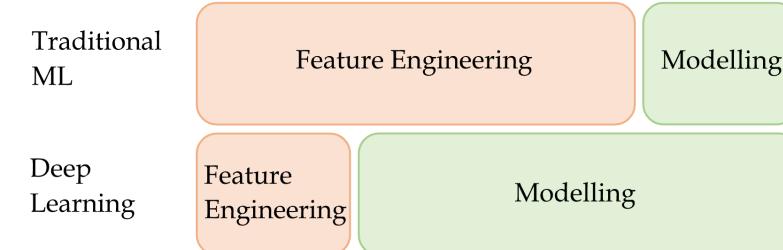
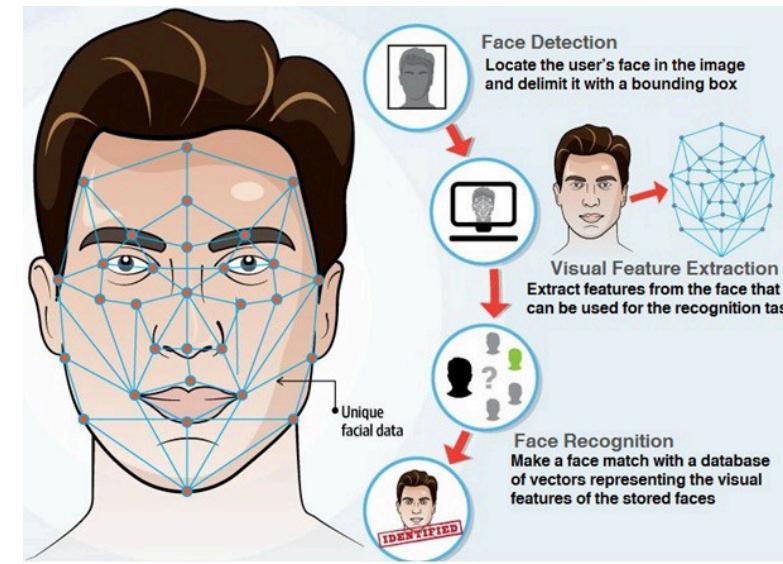
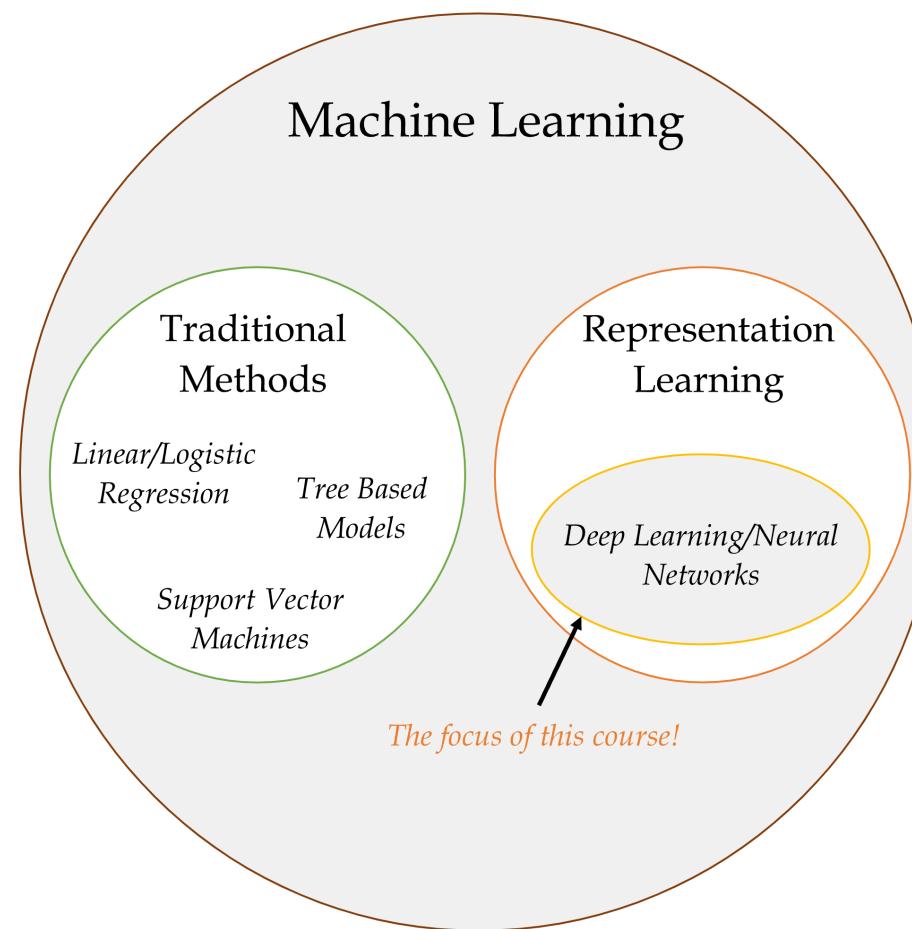
One can show that an MLP is a **universal approximator**, meaning it can model any suitably smooth function, given enough hidden units, to any desired level of accuracy (Hornik 1991). One can either make the model be “wide” or “deep”; the latter has some advantages...



Source: Murphy (2012), Machine Learning: A Probabilistic Perspective, 1st Ed, p. 566.



# Feature engineering



Doesn't mean deep learning is always the best option!



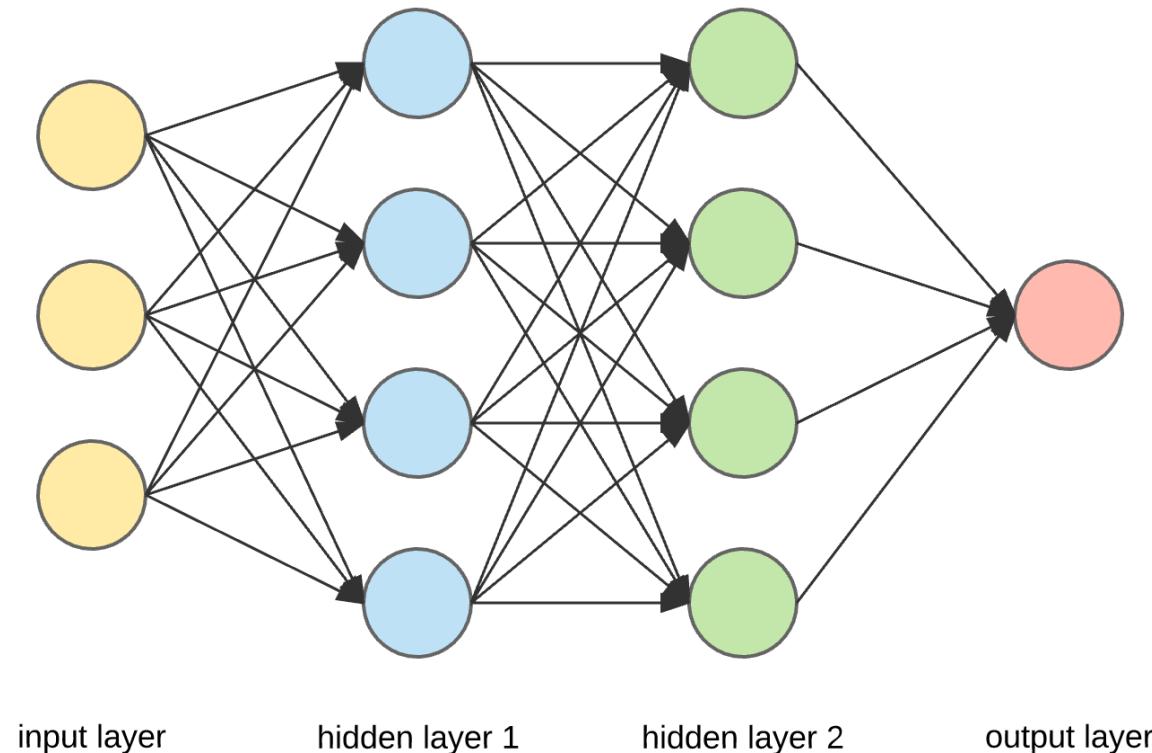
Sources: Marcus Lautier (2022) & Fenjiro (2019), *Face Id: Deep Learning for Face Recognition*, Medium.

# Quiz

In this ANN, how many of the following are there:

- features,
- targets,
- weights,
- biases, and
- parameters?

What is the depth?



An artificial neural network.



Source: Dertat (2017), *Applied Deep Learning - Part 1: Artificial Neural Networks*, Medium.



# Package Versions

```
1 from watermark import watermark  
2 print(watermark(python=True, packages="keras,matplotlib,numpy,pandas,seaborn,scipy,torch"))
```

Python implementation: CPython

Python version : 3.11.11

IPython version : 8.32.0

keras : 3.8.0

matplotlib: 3.10.0

numpy : 1.26.4

pandas : 2.2.3

seaborn : 0.13.2

scipy : 1.13.1

torch : 2.5.1+cu124

tensorflow: 2.18.0

tf\_keras : 2.18.0



# Glossary

- activations, activation function
- artificial neural network
- biases (in neurons)
- classification problem
- deep network, network depth
- dense or fully-connected layer
- feed-forward neural network
- labelled/unlabelled data
- machine learning
- minimax algorithm
- neural network architecture
- perceptron
- ReLU
- representation learning
- sigmoid activation function
- targets
- weights (in a neuron)

