

# **Deep Learning**

## Lecture 1: Introduction

---

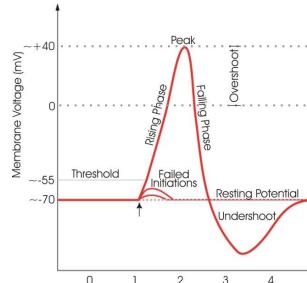
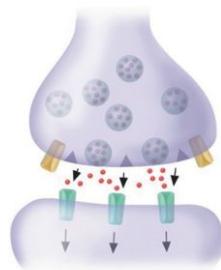
Chris G. Willcocks  
Durham University



# Lecture overview a framework for your deep learning armamentarium

## 1 Introduction

- definitions
- examples



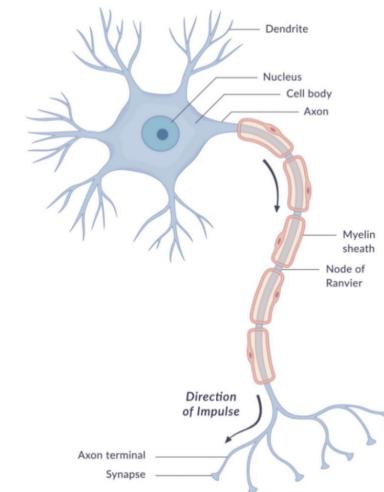
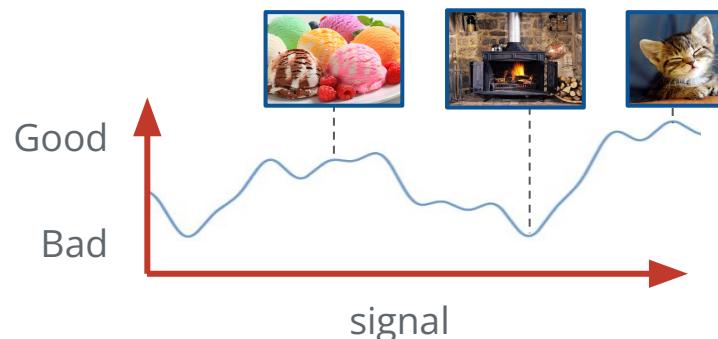
## 2 Learning in Nature

- what is learning?
- how does the brain work?
- synaptic plasticity
- Hebbian theory

## 3 A Brief History

## 4 Key Concepts

- where it fits in to ML/RL
- three spaces



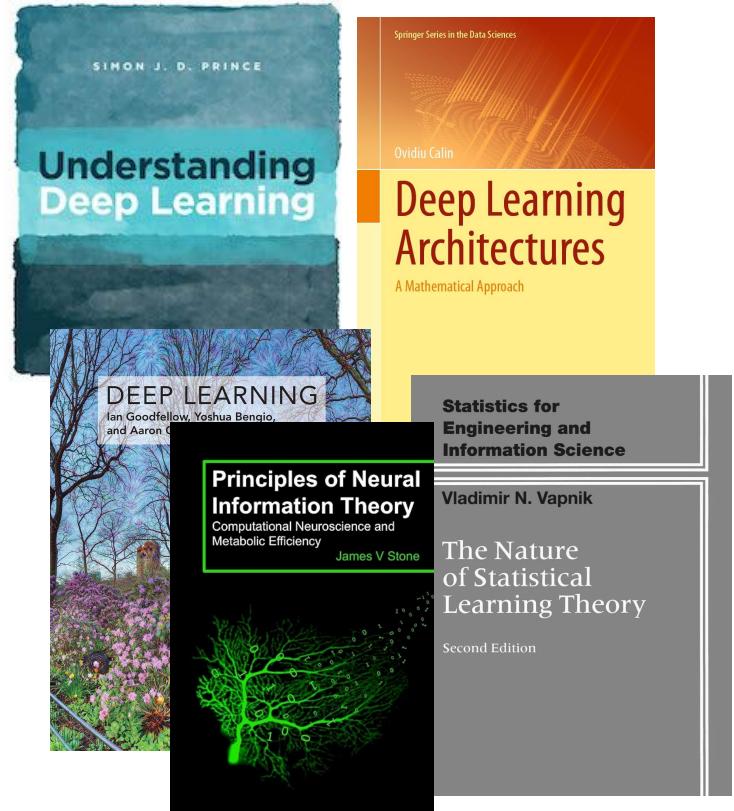


# Module aims learning outcomes

The aims of the course are:

1. To be able to solve complex ill-defined problems that require deep layers of learning
2. To understand natural learning, and how it connects to the theory
3. To ask the right scientific questions given a new task, and use modern deep learning libraries to effectively design, train & test

 PyTorch 



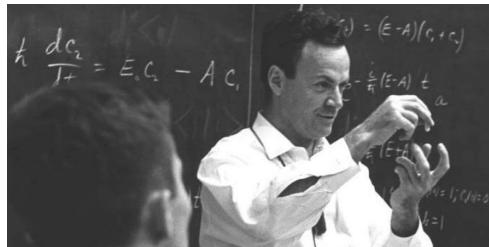


# Learning in nature definition of learning

## Definition: Learning

“We define learning as the transformative process of taking in information that—when internalized and mixed with what we have experienced—changes what we know and builds on what we do. It’s based on input, process, and reflection. It is what changes us.”

*Tony Bingham and Marcia Conner*





# Introduction deep learning definition

## Definition: Deep Learning

“Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics. Deep learning discovers intricate structure in large data sets by using the backpropagation algorithm.”

*Yann LeCun, Yoshua Bengio & Geoffrey Hinton  
Deep Learning, 2015, Nature - [Read Online](#)*

- On the Number of Linear Regions of Deep Neural Networks (Montúfar et al.)
- The Power of Depth for Feedforward Neural Networks (Eldan et al.)



# Introduction examples

## Examples: Images and Vision

Non-photorealistic interpolation:

- Photos - [video](#)
- Paintings - [video](#)
- DeepFakes - [video](#)

End-to-end self-driving:

- Wayve - [video](#)

## Examples: Multimodality

Examples from text:

- GPT-4 and **multimodality**
- **Input:** The internet
- **Output:** Any multimodal task

Examples with audio:

- OpenAI Jukebox - [video](#)
- **Input:** Artist, Genre, Lyrics
- **Output:** New music

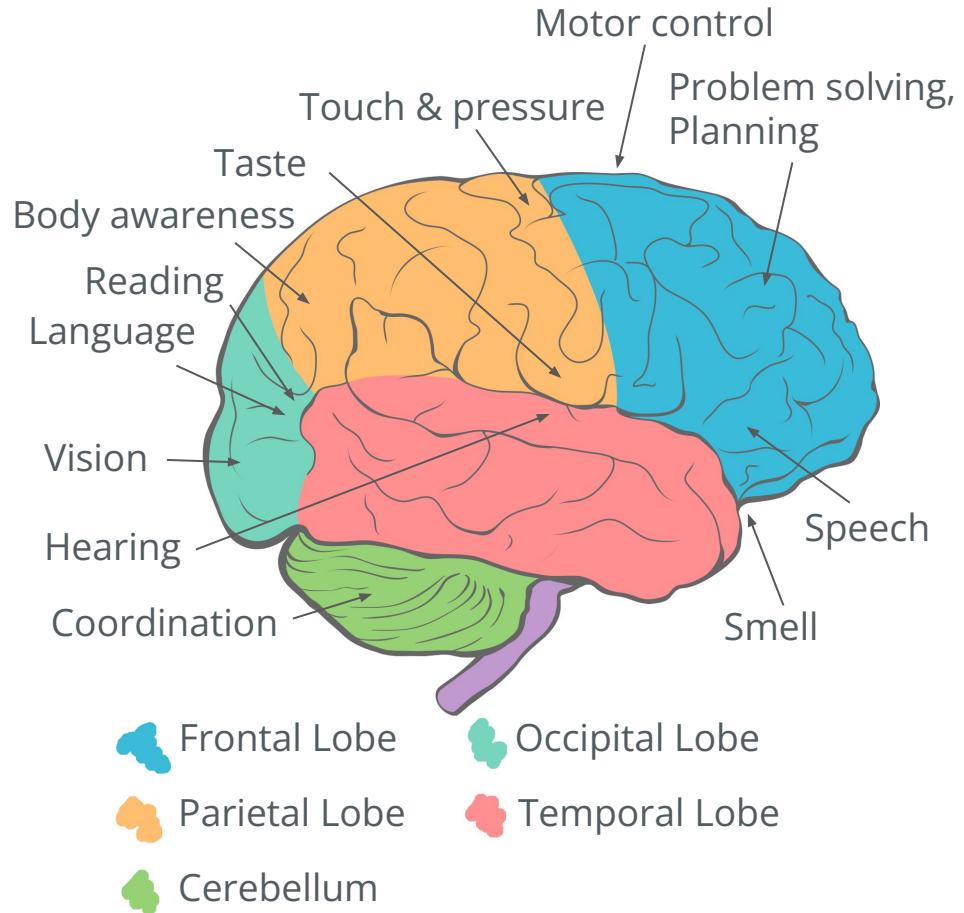


# Learning in nature how does the brain work?

## Architecture of the brain

Right side/left side (hemispheres)

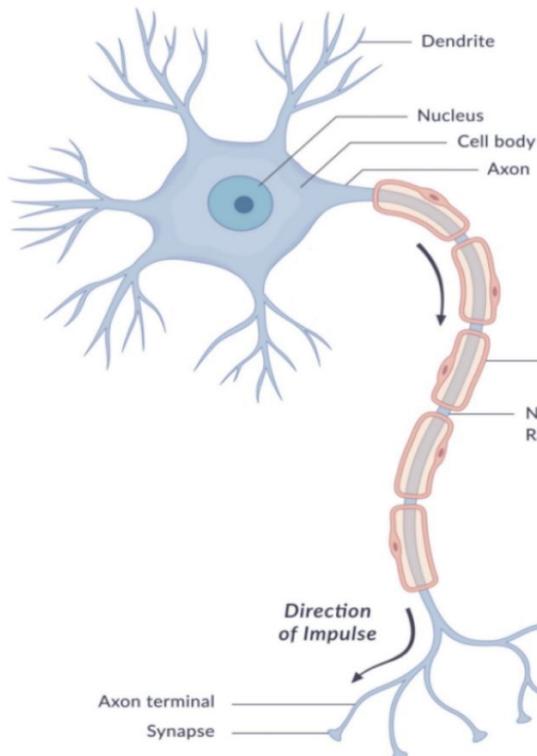
- Frontal lobe
  - executive functions, memory and planning
- Parietal lobe
  - sensation and spatial awareness
- Temporal lobe (banana shape)
  - hearing and language
- Occipital lobe at back
  - Vision from front along optic nerves



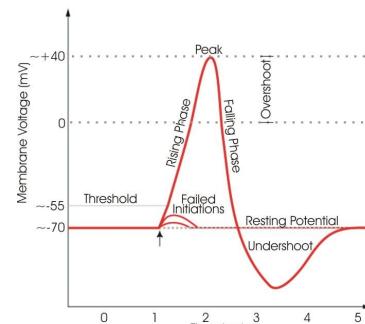


# Learning in nature how does the brain work?

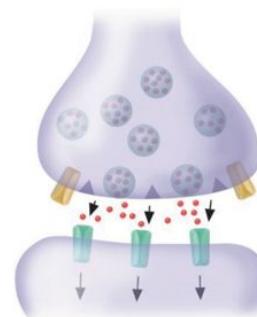
1,000's of inputs (other neurons, sensory neurons e.g. taste buds from a salt or sugar molecule...)



- Neurons send out branches called **dendrites**, and a large output called an **axon**
- The axon is coated in myelin that helps it conduct electrical impulses
- The places where the nerve cells make their connections with each other is called a **synapse**



## Synapse:



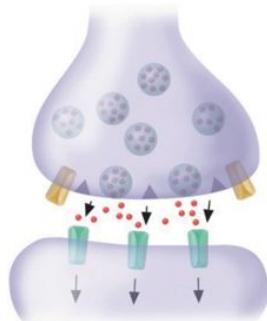
- “Synapse” (from the greek meaning “to clasp together”)
- **Signals get summed up**, and travel to the hillock (Axon neck)
  - If large enough, triggers an action potential travels down axon

1,000's of output targets (e.g. other neurons, muscle cells, gland cells, blood vessels to release hormones...)

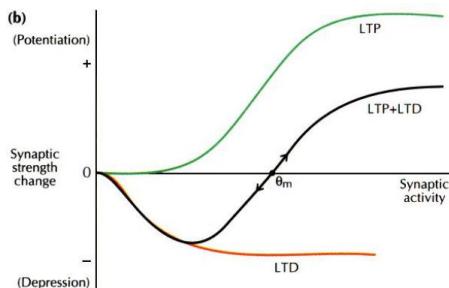


# Learning in nature synaptic plasticity

Presynaptic neuron



Postsynaptic neuron

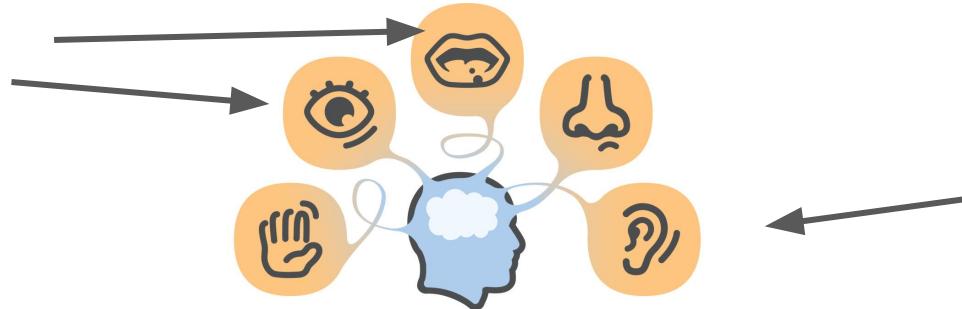


- What's very cool is that with frequent repeated stimulation, the same level of presynaptic stimulation converts into **greater** postsynaptic potential
  - In other words, as a neuron gets a lot of practice sending signals to a specific target neuron, it gets better at sending those signals (the synapse strength increases)
    - Increased strength that lasts for a long time (from minutes to many months) is called **Long Term Potentiation** (weakening is **Long Term Depression**)
  - As synapses are strengthened and retain strength, we're able to more easily recall previous experiences

Figure from: "Synaptic Plasticity: A molecular mechanism for metaplasticity", Journal of Current Biology.



# Learning in nature Hebbian theory



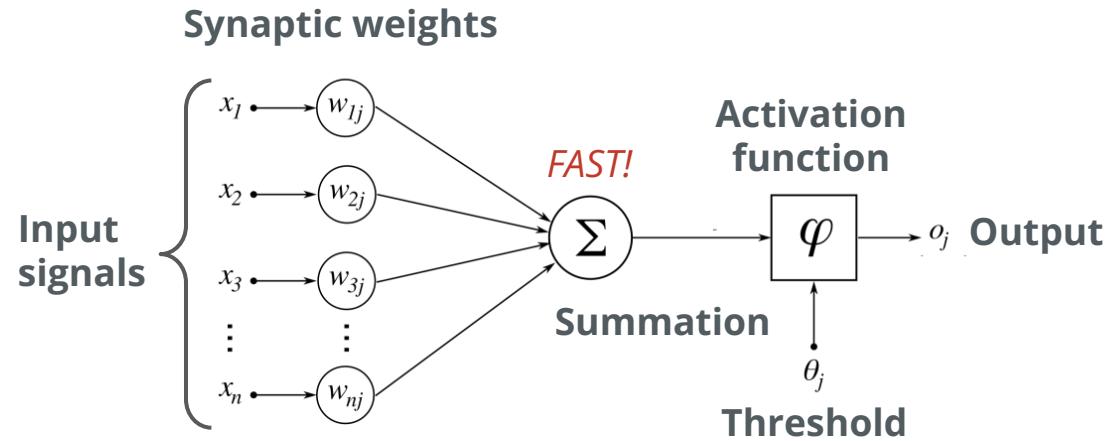
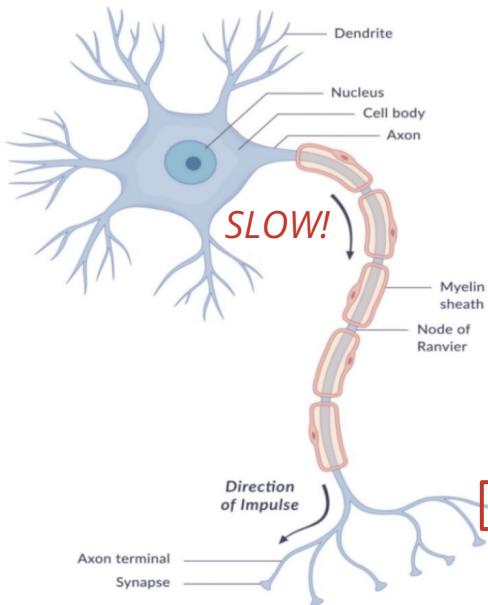
- Hebbian theory
  - If two neurons fire at the same time, the connections between them are strengthened, and thus are more likely to fire again together in the future
  - If two neurons fire in an uncoordinated manner, their connections are weakened, and they're more likely to act independently in the future
- Updated Hebbian hypothesis based on recent findings
  - If the presynaptic neuron fires within a window of 20ms before the postsynaptic neuron, the synapse will be strengthened
  - However if within a window of 20ms after, the synapse will be weakened



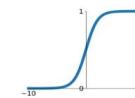
# A brief history of artificial neurons

Pitts and McCulloch, 1943

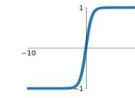
The Artificial Neuron



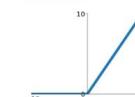
**Sigmoid**  
 $\sigma(x) = \frac{1}{1+e^{-x}}$



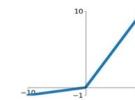
**tanh**  
 $\tanh(x)$



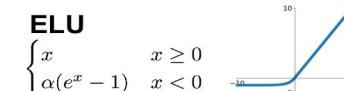
**ReLU**  
 $\max(0, x)$



**Leaky ReLU**  
 $\max(0.1x, x)$



**Maxout**  
 $\max(w_1^T x + b_1, w_2^T x + b_2)$



**ELU**

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

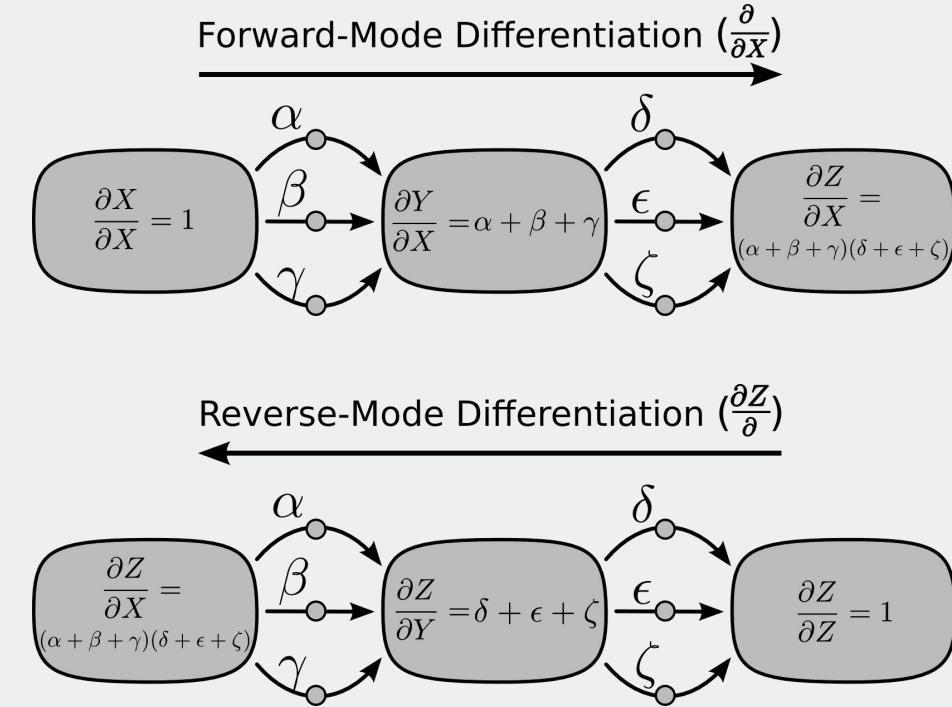
# A brief history backpropagation

## Backpropagation, 1970

The backpropagation algorithm (also called the reverse mode of automatic differentiation) was independently discovered by different researchers

Seppo Linnainmaa, 1970  
Werbos, 1974 (with neural networks)

## Reverse-mode differentiation





# A brief history recurrent neural networks and the second AI winter

Rumelhart and Hinton, 1986

Learning representations by  
back-propagating errors

Nature, 1986. David E.  
Rumelhart, Geoffrey E. Hinton &  
Ronald J. Williams

- Backpropagation
- Multiple hidden layers
- Recurrent networks

The second AI winter 1987-1993





# A brief history of convolutional neural networks

## History of CNNs

Convolutional neural networks were first introduced by Kunihiko Fukushima in 1980

- **1989** - Yann Lecun et al., trained a CNN with “Backpropagation Applied to Handwritten Zip Code Recognition”
- **1998** - Yann Lecun et al., released LeNet5 “Gradient-based learning applied to document recognition”

## Example: LeNet5

```
class LeNet(nn.Module):
    def __init__(self):
        super(LeNet, self).__init__()
        self.conv1 = nn.Conv2d(1, 6, 5, padding=2)
        self.conv2 = nn.Conv2d(6, 16, 5)
        self.fc1   = nn.Linear(16*5*5, 120)
        self.fc2   = nn.Linear(120, 84)
        self.fc3   = nn.Linear(84, 10)

    def forward(self, x):
        x = F.max_pool2d(F.relu(self.conv1(x)), (2, 2))
        x = F.max_pool2d(F.relu(self.conv2(x)), (2, 2))
        x = flatten(x)
        x = F.relu(self.fc1(x))
        x = F.relu(self.fc2(x))
        x = self.fc3(x)
        return x
```



# A brief history the current AI spring - GPUs and large benchmarks

## The current AI spring



## History of NNs on GPUs

- **2004** - first GPU implementation of a neural network
- **2006** - first GPU implementation of a CNN (just 4 times faster)
- **2012** - AlexNet - won state-of-the-art by significant margin with 60 million parameters
- **2015** - ImageNet state-of-the-art by a residual network with over 100 layers



# A brief history of cloud computing - GANs, transformers, and beyond

## Cloud Computing - GANs & GPT-4

Fake faces generated by StyleGAN2



## Unsupervised Generative Models

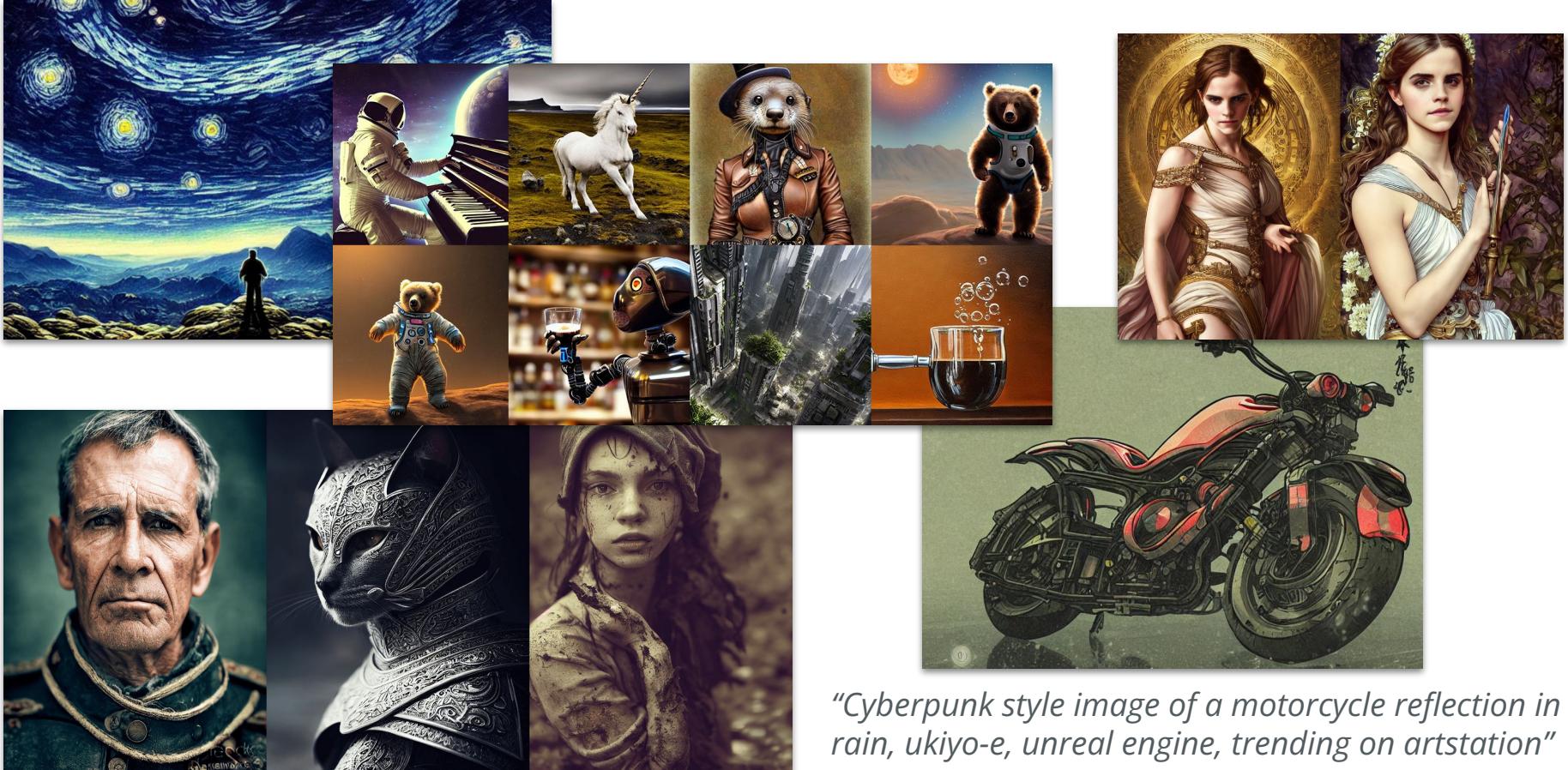
Progressively larger models are being trained on GPU cloud services

- **2014** - Ian Goodfellow releases “Generative Adversarial Nets”
- **2017** - GoogleBrain releases “Attention is all you need”
- **2019** - OpenAI GPT-2 “Language Models are Unsupervised Multitask Learners”
- **2020+** OpenAI GPT-3 “Language Models are Few-Shot Learners”



2023

GPT-4 multimodality, stable diffusion, plugins...





# Key concepts overlap with other areas of machine learning

## Machine Learning

Shallow Learning  
often hand-engineered

## Reinforcement Learning

bad gradients  
dynamic not IID  
optimising future reward

## Deep Learning

multiple layers  
gradients  
mostly IID

## Unsupervised Learning

Generative Models  
learning the data  
distribution

## Meta Learning

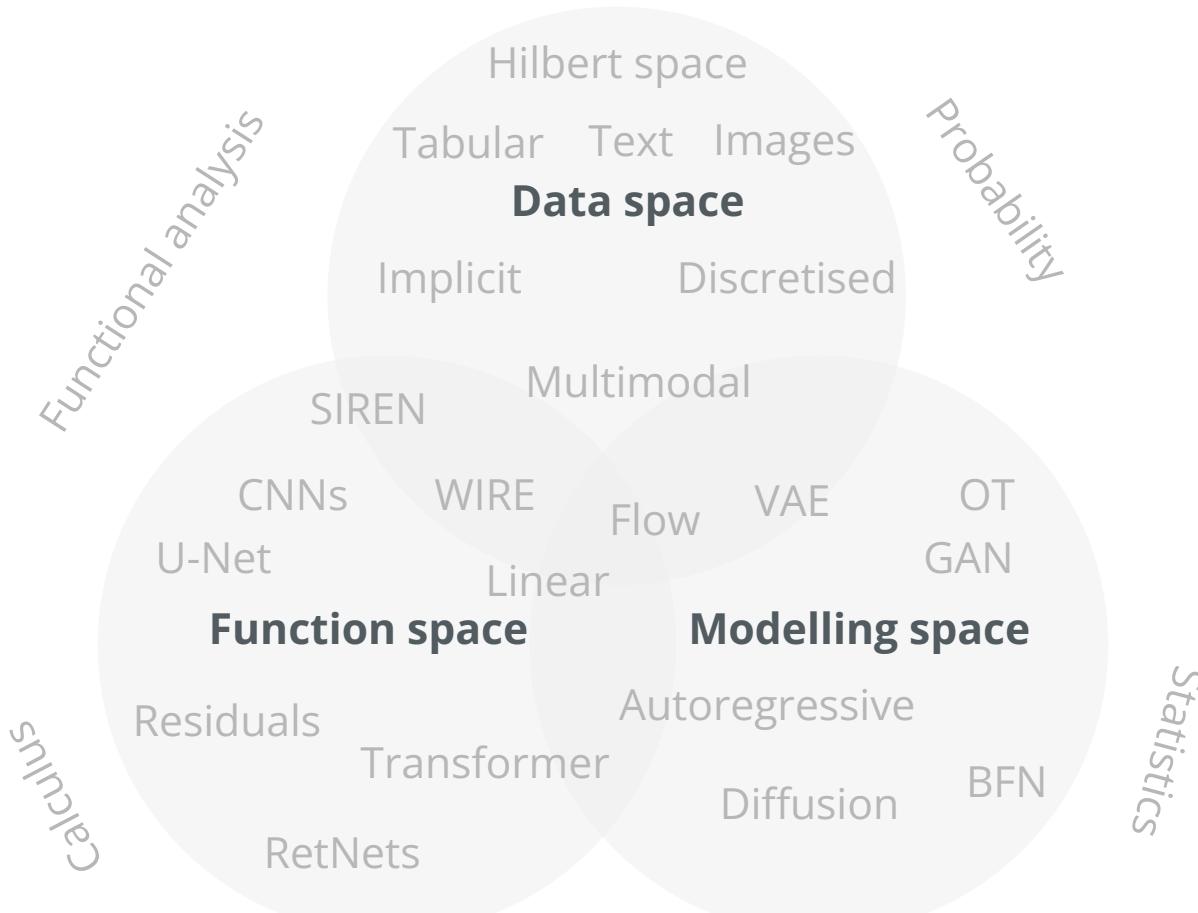
learning the task  
distribution

Supervised Learning  
learning one task

Discriminative Models  
classifying data  
regression



# Key concepts deep learning contributions from three spaces





# Take away points

## Summary

In summary, deep learning:

- has overlap with many areas
- achieves state-of-the-art in ill-defined tasks and scales with...
  - very high-dimensional datasets
  - huge datasets
  - parallelizable hardware
- fields of maths: probability, calculus, statistics, functional analysis...
- is rapidly growing and evolving