Introduction

CISC 7026: Introduction to Deep Learning

University of Macau

Overview

1. Brief chat
2. Course Information
3. Course Structure
4. Lecture

Overview

1. **Brief chat**
2. Course Information
3. Course Structure
4. Lecture

Brief Chat

This is my first course at UM

Brief Chat

This is my first course at UM

I taught a course on Deep Reinforcement Learning at Cambridge

Brief Chat

This is my first course at UM

I taught a course on Deep Reinforcement Learning at Cambridge

I am not perfect, and I am still learning how to teach effectively

Brief Chat

This is my first course at UM

I taught a course on Deep Reinforcement Learning at Cambridge

I am not perfect, and I am still learning how to teach effectively

Please provide feedback privately to me

Brief Chat

This is my first course at UM

I taught a course on Deep Reinforcement Learning at Cambridge

I am not perfect, and I am still learning how to teach effectively

Please provide feedback privately to me
- Email smorad at um.edu.mo
- Chat after class

Brief Chat

This is my first course at UM

I taught a course on Deep Reinforcement Learning at Cambridge

I am not perfect, and I am still learning how to teach effectively

Please provide feedback privately to me
- Email smorad at um.edu.mo
- Chat after class

I would like to make the class **interactive**

Brief Chat

This is my first course at UM

I taught a course on Deep Reinforcement Learning at Cambridge

I am not perfect, and I am still learning how to teach effectively

Please provide feedback privately to me
- Email smorad at um.edu.mo
- Chat after class

I would like to make the class **interactive**

The best way to learn is to **ask questions** and have **discussions**

Brief Chat

I will tell you about myself, and why I am interested in deep learning

Brief Chat

I will tell you about myself, and why I am interested in deep learning

Then, **you** will tell me why you are interested in deep learning

Brief Chat

I will tell you about myself, and why I am interested in deep learning

Then, **you** will tell me why you are interested in deep learning

It will help me alter the course towards your goals

Brief Chat

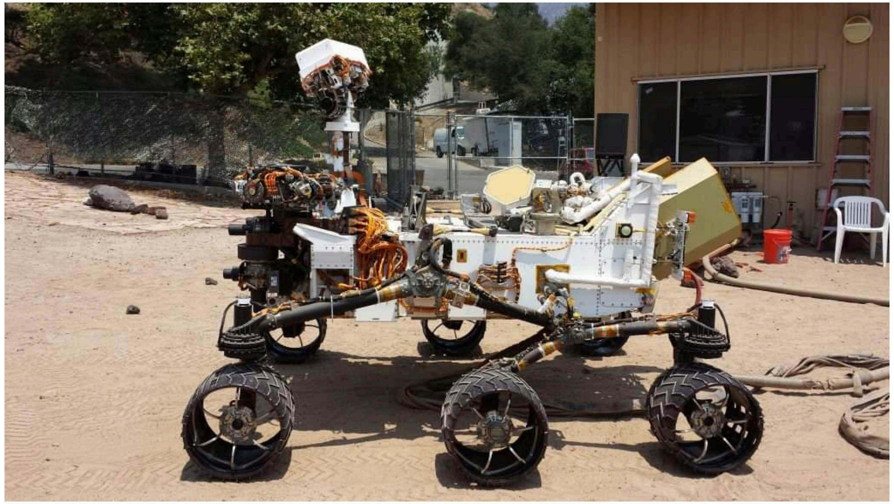
I was always interested in space and robotics

Brief Chat

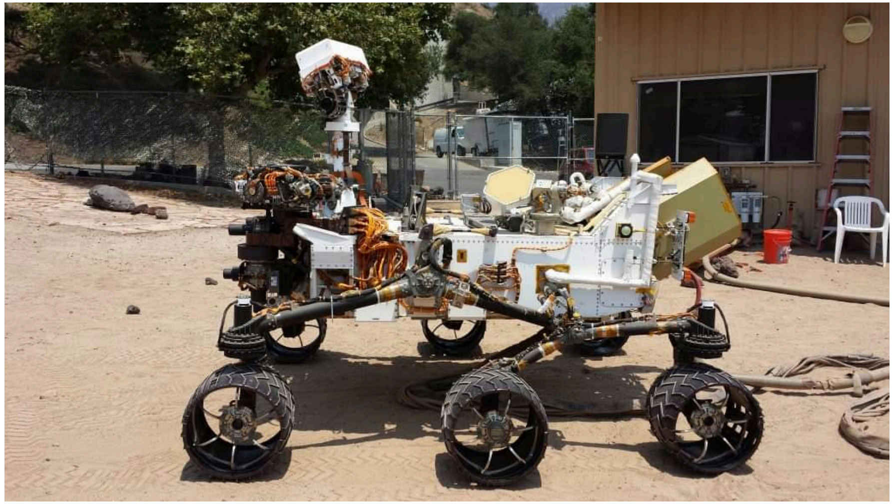
I was always interested in space and robotics

After school, I realized much of the classical robotics that we learn in school **does not work** in reality

After school, I realized much of the classical robotics that we learn in school **does not work** in reality



After school, I realized much of the classical robotics that we learn in school **does not work** in reality



Today's robots are stupid – important robots are human controlled

Since then, I have focused on creating less stupid robots

Since then, I have focused on creating less stupid robots

Since then, I have focused on creating less stupid robots

Robots that **learn** from their mistakes

I am interested in **deep learning** because I want to make smarter robots

I am interested in **deep learning** because I want to make smarter robots

There are many tasks that humans do not like to do, that robots can do

I am interested in **deep learning** because I want to make smarter robots

There are many tasks that humans do not like to do, that robots can do

What do you want to learn? Why?

Brief Chat

I am starting a lab, and looking for research students focusing on deep reinforcement learning and robotics

Brief Chat

I am starting a lab, and looking for research students focusing on deep reinforcement learning and robotics

If you finish the course and find it too easy, send me an email!

Overview

1. **Brief chat**
2. Course Information
3. Course Structure
4. Lecture

Overview

1. Brief chat
2. **Course Information**
3. Course Structure
4. Lecture

Course Info

Most communication will happen over Moodle

Course Info

Most communication will happen over Moodle

- I will try and post lecture slides after each lecture

Course Info

Most communication will happen over Moodle

- I will try and post lecture slides after each lecture
- Assignments

Course Info

Most communication will happen over Moodle

- I will try and post lecture slides after each lecture
- Assignments
- Grading

Course Info

Prerequisites:

Course Info

Prerequisites:

- Programming in python

Course Info

Prerequisites:

- Programming in python
  - Should be able to implement a stack, etc

Course Info

Prerequisites:

- Programming in python
  - Should be able to implement a stack, etc
- Linear algebra

Course Info

Prerequisites:

- Programming in python
  - Should be able to implement a stack, etc
- Linear algebra
  - Multiply matrices, invert matrices, solve systems of equations, etc

Course Info

Prerequisites:

- Programming in python
  - Should be able to implement a stack, etc
- Linear algebra
  - Multiply matrices, invert matrices, solve systems of equations, etc
- Multivariable calculus

Course Info

Prerequisites:

- Programming in python
  - Should be able to implement a stack, etc
- Linear algebra
  - Multiply matrices, invert matrices, solve systems of equations, etc
- Multivariable calculus
  - Computing gradients $\begin{pmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \dots \end{pmatrix}^\top$

Course Info

Prerequisites:

- Programming in python
  - Should be able to implement a stack, etc
- Linear algebra
  - Multiply matrices, invert matrices, solve systems of equations, etc
- Multivariable calculus
  - Computing gradients $\begin{pmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \dots \end{pmatrix}^\top$

Good to Know:

Course Info

Prerequisites:

- Programming in python
  - Should be able to implement a stack, etc
- Linear algebra
  - Multiply matrices, invert matrices, solve systems of equations, etc
- Multivariable calculus
  - Computing gradients $\begin{pmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \dots \end{pmatrix}^\top$

Good to Know:

- Probability

Course Info

Prerequisites:

- Programming in python
  - Should be able to implement a stack, etc
- Linear algebra
  - Multiply matrices, invert matrices, solve systems of equations, etc
- Multivariable calculus
  - Computing gradients $\left(\frac{\partial f}{\partial x_1} \quad \frac{\partial f}{\partial x_2} \quad \ldots\right)^\top$

Good to Know:

- Probability
  - Bayes rule, conditional probabilities $P(a \mid b) = \frac{P(b \mid a)P(a)}{P(b)}$

Course Info

Grading (subject to change):

Course Info

Grading (subject to change):

- 70% assignments

Course Info

Grading (subject to change):

- 70% assignments
- 20% quiz

Course Info

Grading (subject to change):

- 70% assignments
- 20% quiz
- 10% attendance and participation
  - Name plates

Overview

1. Brief chat
2. **Course Information**
3. Course Structure
4. Lecture

Overview

1. Brief chat
2. Course Information
3. **Course Structure**
4. Lecture

Course Structure

This course is structured after the *Dive into Deep Learning* textbook and Prof. Dingqi Yang's slides

Course Structure

This course is structured after the *Dive into Deep Learning* textbook and Prof. Dingqi Yang's slides

The textbook is available for free online at https://d2l.ai

Course Structure

This course is structured after the *Dive into Deep Learning* textbook and Prof. Dingqi Yang's slides

The textbook is available for free online at https://d2l.ai

If you get confused by a lecture, try reading the corresponding chapter

Course Structure

This course is structured after the *Dive into Deep Learning* textbook and Prof. Dingqi Yang's slides

The textbook is available for free online at https://d2l.ai

If you get confused by a lecture, try reading the corresponding chapter

Also available in Chinese at https://zh.d2l.ai

Course Structure

This course is structured after the *Dive into Deep Learning* textbook and Prof. Dingqi Yang's slides

The textbook is available for free online at https://d2l.ai

If you get confused by a lecture, try reading the corresponding chapter

Also available in Chinese at https://zh.d2l.ai

We will start from the basics and learn the theory behind Deep Learning

Course Structure

This course is structured after the *Dive into Deep Learning* textbook and Prof. Dingqi Yang's slides

The textbook is available for free online at https://d2l.ai

If you get confused by a lecture, try reading the corresponding chapter

Also available in Chinese at https://zh.d2l.ai

We will start from the basics and learn the theory behind Deep Learning

Your assignments will teach you two popular Deep Learning libraries:

Course Structure

This course is structured after the *Dive into Deep Learning* textbook and Prof. Dingqi Yang's slides

The textbook is available for free online at https://d2l.ai

If you get confused by a lecture, try reading the corresponding chapter

Also available in Chinese at https://zh.d2l.ai

We will start from the basics and learn the theory behind Deep Learning

Your assignments will teach you two popular Deep Learning libraries:

- PyTorch

Course Structure

This course is structured after the *Dive into Deep Learning* textbook and Prof. Dingqi Yang's slides

The textbook is available for free online at https://d2l.ai

If you get confused by a lecture, try reading the corresponding chapter

Also available in Chinese at https://zh.d2l.ai

We will start from the basics and learn the theory behind Deep Learning

Your assignments will teach you two popular Deep Learning libraries:
- PyTorch
- JAX

Course Structure

3 hours is a very long time to pay attention

Course Structure

3 hours is a very long time to pay attention

I will place slides titled "Relax", every ~ 45 minutes

Course Structure

3 hours is a very long time to pay attention

I will place slides titled "Relax", every $\sim$ 45 minutes

At the Relax slides, we may take a $\sim$ 15 minute break

Course Structure

3 hours is a very long time to pay attention

I will place slides titled "Relax", every $\sim$ 45 minutes

At the Relax slides, we may take a $\sim$ 15 minute break

You can:

Course Structure

3 hours is a very long time to pay attention

I will place slides titled "Relax", every ~ 45 minutes

At the Relax slides, we may take a ~ 15 minute break

You can:

- Leave the classroom

Course Structure

3 hours is a very long time to pay attention

I will place slides titled "Relax", every ~ 45 minutes

At the Relax slides, we may take a ~ 15 minute break

You can:

- Leave the classroom
- Use the toilet

Course Structure

3 hours is a very long time to pay attention

I will place slides titled "Relax", every $\sim$ 45 minutes

At the Relax slides, we may take a $\sim$ 15 minute break

You can:

- Leave the classroom
- Use the toilet
- Walk around

Course Structure

3 hours is a very long time to pay attention

I will place slides titled "Relax", every $\sim$ 45 minutes

At the Relax slides, we may take a $\sim$ 15 minute break

You can:

- Leave the classroom
- Use the toilet
- Walk around
- Ask me questions

Course Structure

3 hours is a very long time to pay attention

I will place slides titled "Relax", every $\sim$ 45 minutes

At the Relax slides, we may take a $\sim$ 15 minute break

You can:

- Leave the classroom
- Use the toilet
- Walk around
- Ask me questions

This format is subject to change

Course Structure

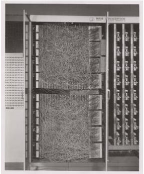
We will be following the history of machine learning

Course Structure

We will be following the history of machine learning

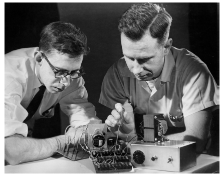
Linear Regression (1795)

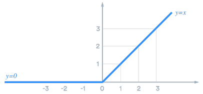
Neural Network (1943)



Hebbian Learning (1949)

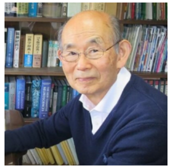
Deep Neural Network (1958)





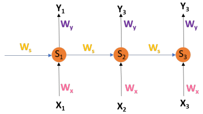
ReLU Activation Function (1969)

Recurrent Neural Network (1972)



Convolutional Neural Network (1980)

Backpropagation (1982)



Backpropagation Through Time (1990)

Alexnet (2012)

Deep Q Network (2014)

Generative Adversarial Network (2014)

Transformer (2017)

Generative Pre-Training (2018)

Diffusion Model (2020)

GPT-3 (2020)

Overview

1. Brief chat
2. Course Information
3. **Course Structure**
4. Lecture

Overview

1. Brief chat
2. Course Information
3. Course Structure
4. **Lecture**

Relax

Introduction

CISC 7026: Introduction to Deep Learning

University of Macau

Overview

1. Deep Learning Successes

Overview

1. Deep Learning Successes
2. What is Deep Learning?

Overview

1. Deep Learning Successes
2. What is Deep Learning?
3. Differences between AI, ML, DL

Overview

1. Deep Learning Successes
2. What is Deep Learning?
3. Differences between AI, ML, DL
4. Define Machine Learning

Overview

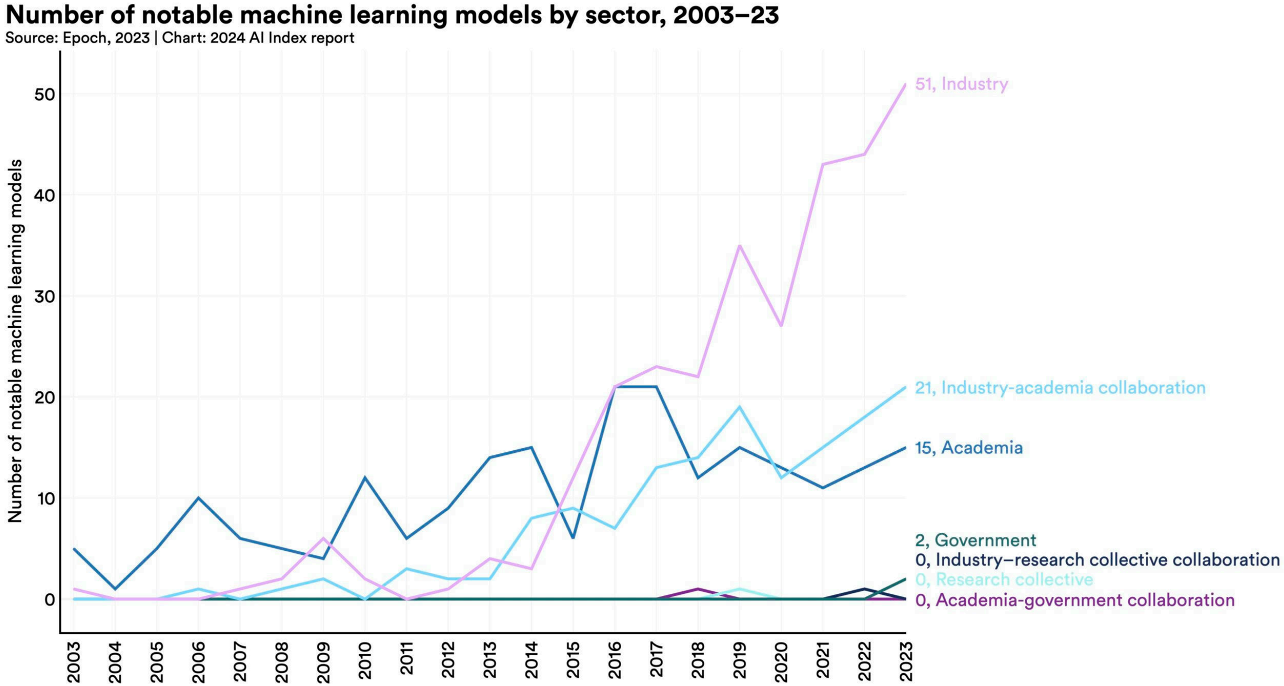
1. Deep Learning Successes
2. What is Deep Learning?
3. Differences between AI, ML, DL
4. Define Machine Learning
5. Machine Learning Libraries

Overview

1. **Deep Learning Successes**
2. What is Deep Learning?
3. Differences between AI, ML, DL
4. Define Machine Learning
5. Machine Learning Libraries

Successes

Deep learning is becoming very popular worldwide



Credit: Stanford University 2024 AI Index Report

Successes

Many things that were once considered science fiction are now possible through deep learning. It can draw pictures for us

Create an image expressing how far deep learning has come

Here is the image depicting the advancements in deep learning. It showcases a futuristic cityscape with various elements symbolizing the integration of AI and deep learning into society. Let me know if you need any adjustments or further details!

Successes

It can beat the world champions at difficult video games like DotA 2

https://www.youtube.com/watch?v=UZHTNBMAfAA

Successes

It is learning to use tools and break rules

https://youtu.be/kopoLzvh5jY?si=keH4i8noY4zUVNrP

Successes

It is operating fully autonomous taxis in four cities

https://www.youtube.com/watch?v=Zeyv1bN9v4A

Successes

Maybe it is doing your homework, then explaining itself

Write a stack implementation in python

Certainly! Below is a simple stack implementation in Python using a list. The stack will support typical operations such as push, pop, peek, and checking if the stack is empty.

python Copy code

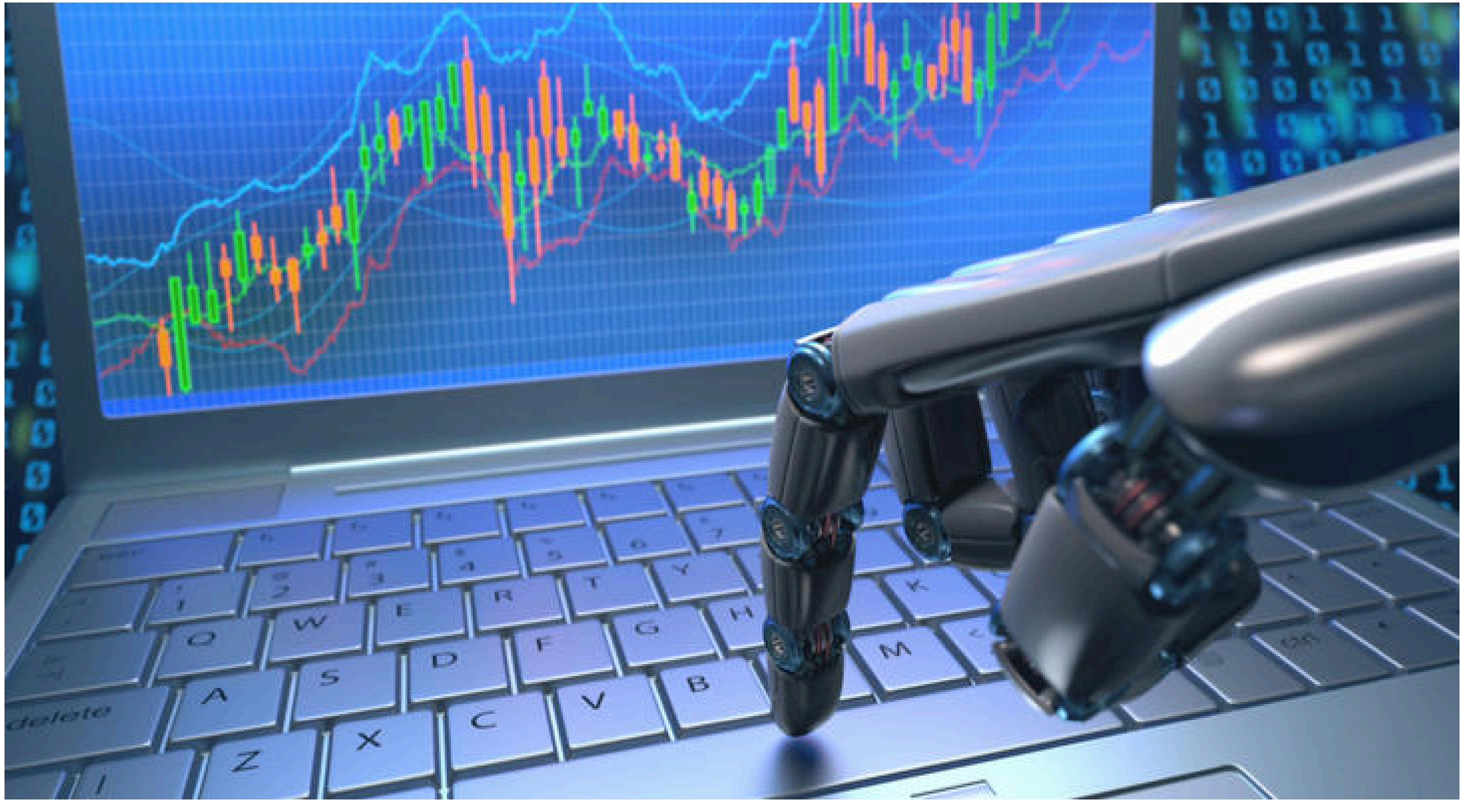
```python
class Stack:
    def __init__(self):
        self.items = []

    def is_empty(self):
        return len(self.items) == 0

    def push(self, item):
        self.items.append(item)

    def pop(self):
        if self.is_empty():
            raise IndexError("pop from empty stack")
        return self.items.pop()

    def peek(self):
        if self.is_empty():
            raise IndexError("peek from empty stack")
        return self.items[-1]

    def size(self):
        return len(self.items)

    def __str__(self):
        return str(self.items)

# Example usage:
if __name__ == "__main__":
    stack = Stack()
    stack.push(1)
    stack.push(2)
    stack.push(3)
    print("Stack after pushes:", stack)
    print("Peek:", stack.peek())
    print("Pop:", stack.pop())
    print("Stack after pop:", stack)
    print("Is stack empty?", stack.is_empty())
    print("Stack size:", stack.size())
```

Explanation:

1. **Initialization**: The stack is initialized with an empty list `self.items`.
2. **is_empty**: Checks if the stack is empty by returning `True` if the length of `self.items` is 0.

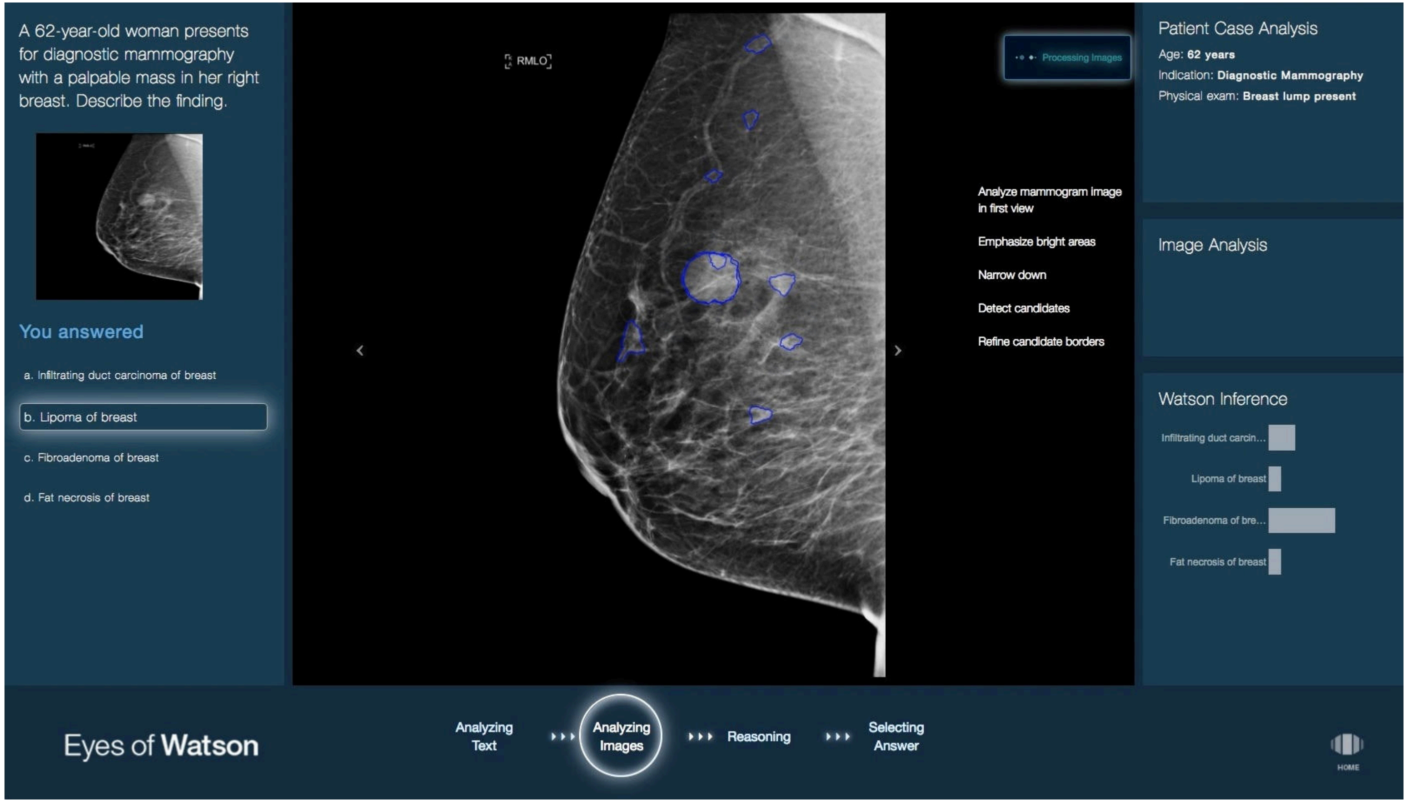
Successes

It is making you lose money in the stock market



Successes

It is telling your doctor if you have cancer



Successes

We are solving more and more problems using deep learning

Successes

We are solving more and more problems using deep learning

Deep learning is creeping into our daily lives

Successes

We are solving more and more problems using deep learning

Deep learning is creeping into our daily lives

In many cases, deep models outperform humans

Successes

We are solving more and more problems using deep learning

Deep learning is creeping into our daily lives

In many cases, deep models outperform humans

Our deep models keep improving as we get more data

Successes

We are solving more and more problems using deep learning

Deep learning is creeping into our daily lives

In many cases, deep models outperform humans

Our deep models keep improving as we get more data

Opinion: In the next 10-20 years, our lives will look very different

Successes

Throughout this course, you will be training your own deep models

Successes

Throughout this course, you will be training your own deep models

After the course, you will be experts at deep learning

Successes

Throughout this course, you will be training your own deep models

After the course, you will be experts at deep learning

Deep learning is a powerful tool

Successes

Throughout this course, you will be training your own deep models

After the course, you will be experts at deep learning

Deep learning is a powerful tool

Like all powerful tools, deep learning can be used for evil

Successes

Throughout this course, you will be training your own deep models

After the course, you will be experts at deep learning

Deep learning is a powerful tool

Like all powerful tools, deep learning can be used for evil

Request: Before you train a deep model, ask yourself whether it is good or bad for the world

Overview

1. **Deep Learning Successes**
2. What is Deep Learning?
3. Differences between AI, ML, DL
4. Define Machine Learning
5. Machine Learning Libraries

Overview

1. Deep Learning Successes
2. **What is Deep Learning?**
3. Differences between AI, ML, DL
4. Define Machine Learning
5. Machine Learning Libraries

DL at a Glance

At a high level, how does deep learning work?

DL at a Glance

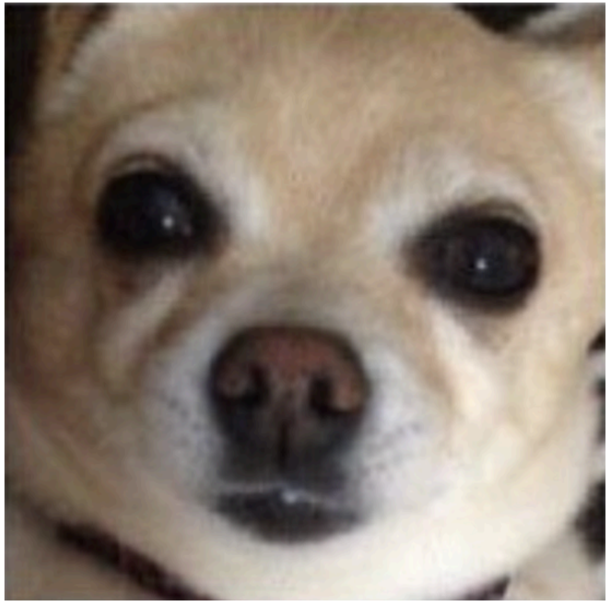
At a high level, how does deep learning work?

It consists of four parts:

1. Dataset
2. Deep neural network
3. Loss function
4. Optimization procedure

DL at a Glance

The dataset provides a set inputs and associated outputs



Dog

Muffin

DL at a Glance

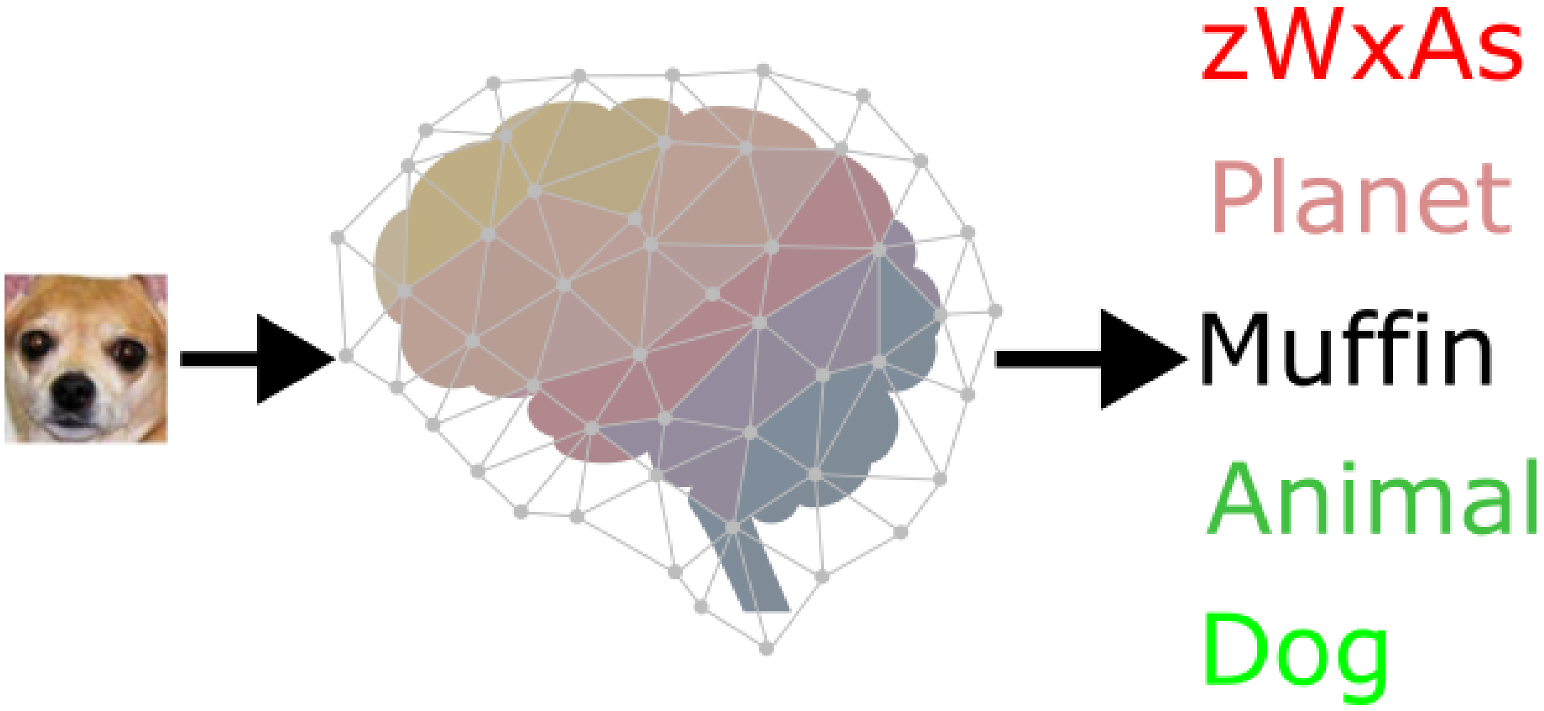
The **neural network** learns to map the inputs to outputs

DL at a Glance

The **loss function** describes how “wrong” the neural network is. We call this “wrongness” the **loss**.

DL at a Glance

The **loss function** describes how "wrong" the neural network is. We call this "wrongness" the **loss**.

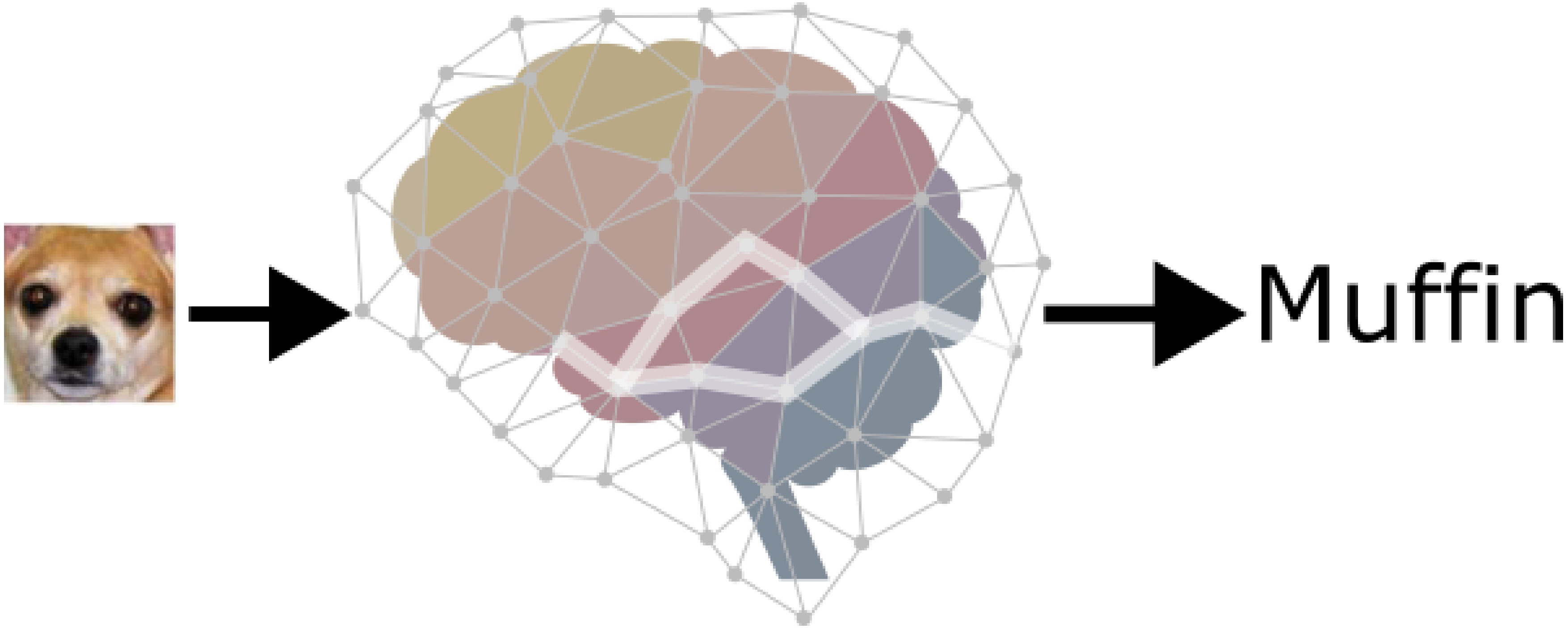


DL at a Glance

The **optimization procedure** changes the neural network to reduce the loss

DL at a Glance

The **optimization procedure** changes the neural network to reduce the loss



DL at a Glance

At a high level, how does deep learning work?

It consists of four parts:

1. Dataset
2. Deep neural network
3. Loss function
4. Optimization procedure

Overview

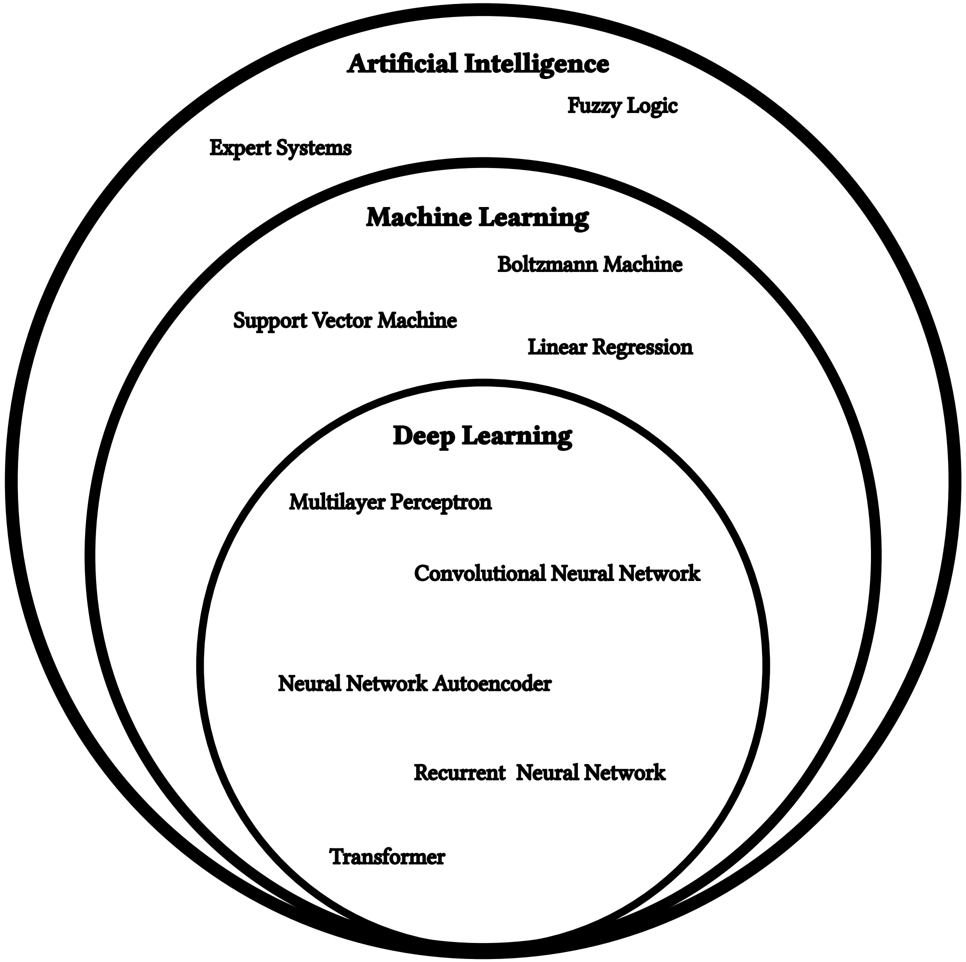
1. Deep Learning Successes
2. **What is Deep Learning?**
3. Differences between AI, ML, DL
4. Define Machine Learning
5. Machine Learning Libraries

Relax

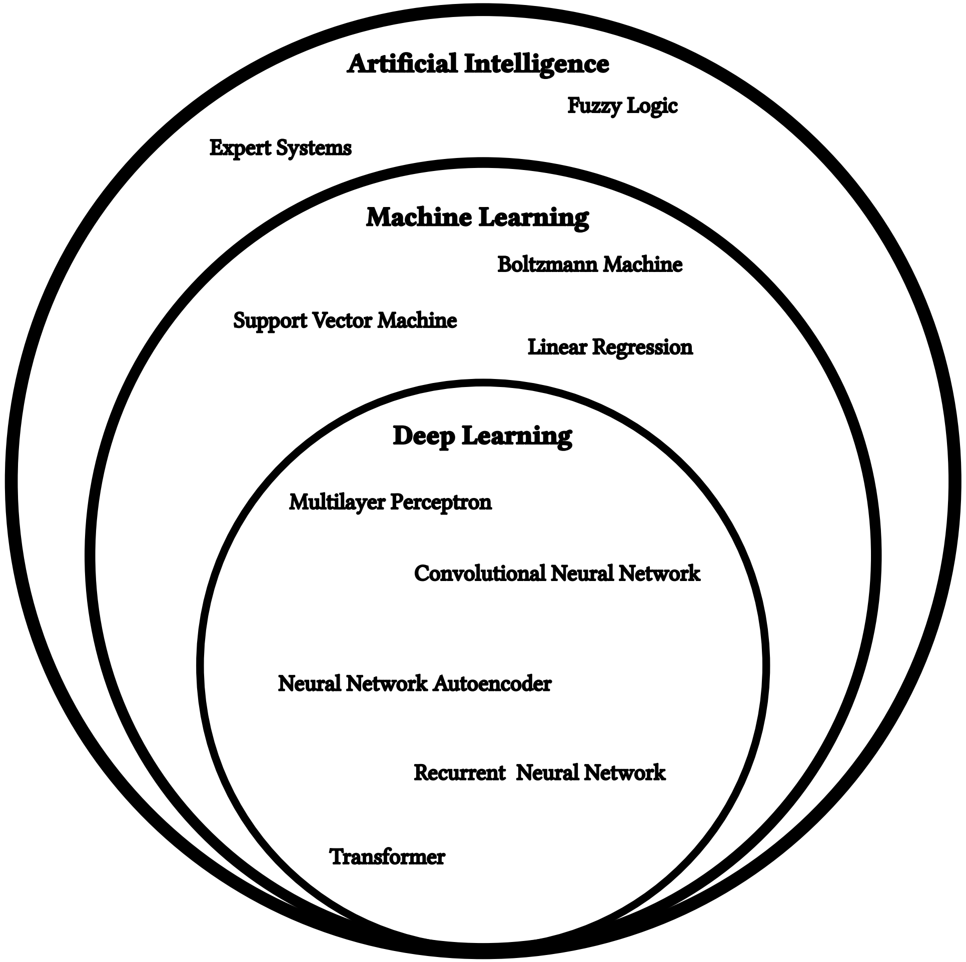
Overview

1. Deep Learning Successes
2. What is Deep Learning?
3. **Differences between AI, ML, DL**
4. Define Machine Learning
5. Machine Learning Libraries

AI, ML, DL

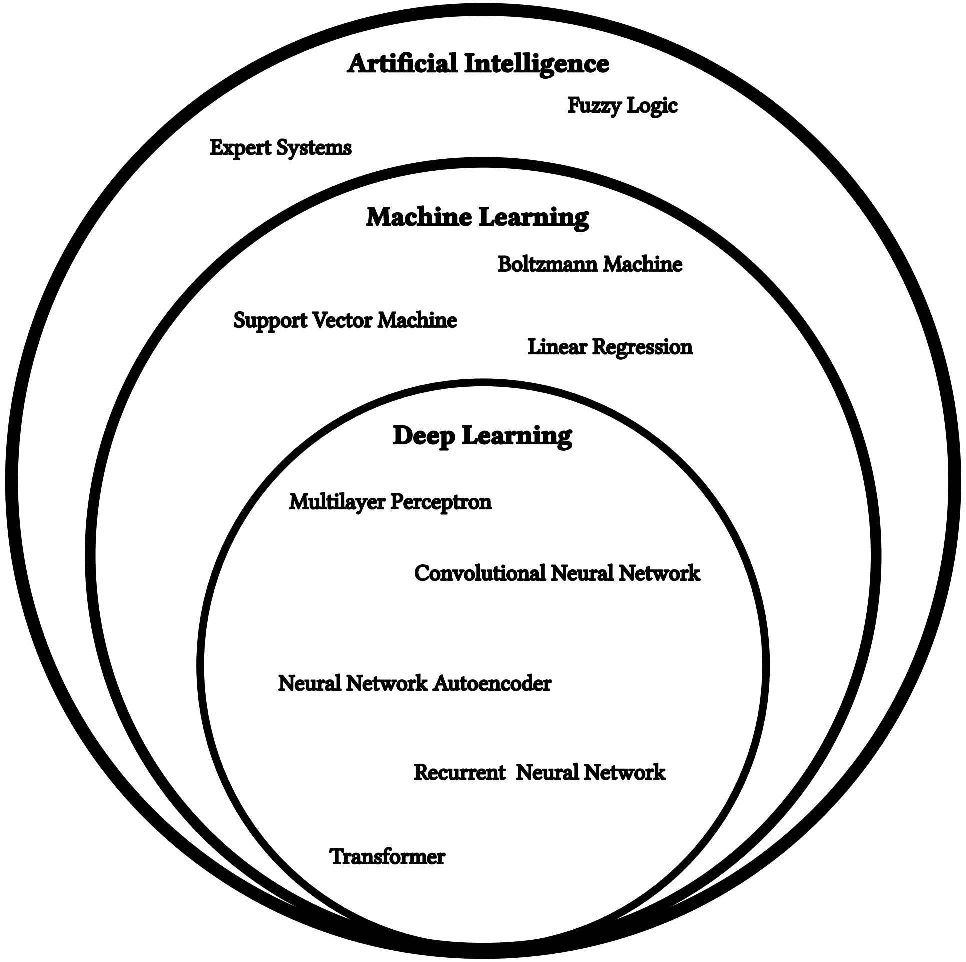

Artificial Intelligence
Fuzzy Logic
Expert Systems
Machine Learning
Boltzmann Machine
Support Vector Machine
Linear Regression
Deep Learning
Multilayer Perceptron
Convolutional Neural Network
Neural Network Autoencoder
Recurrent Neural Network
Transformer

AI, ML, DL



Deep Learning is a type of Machine Learning

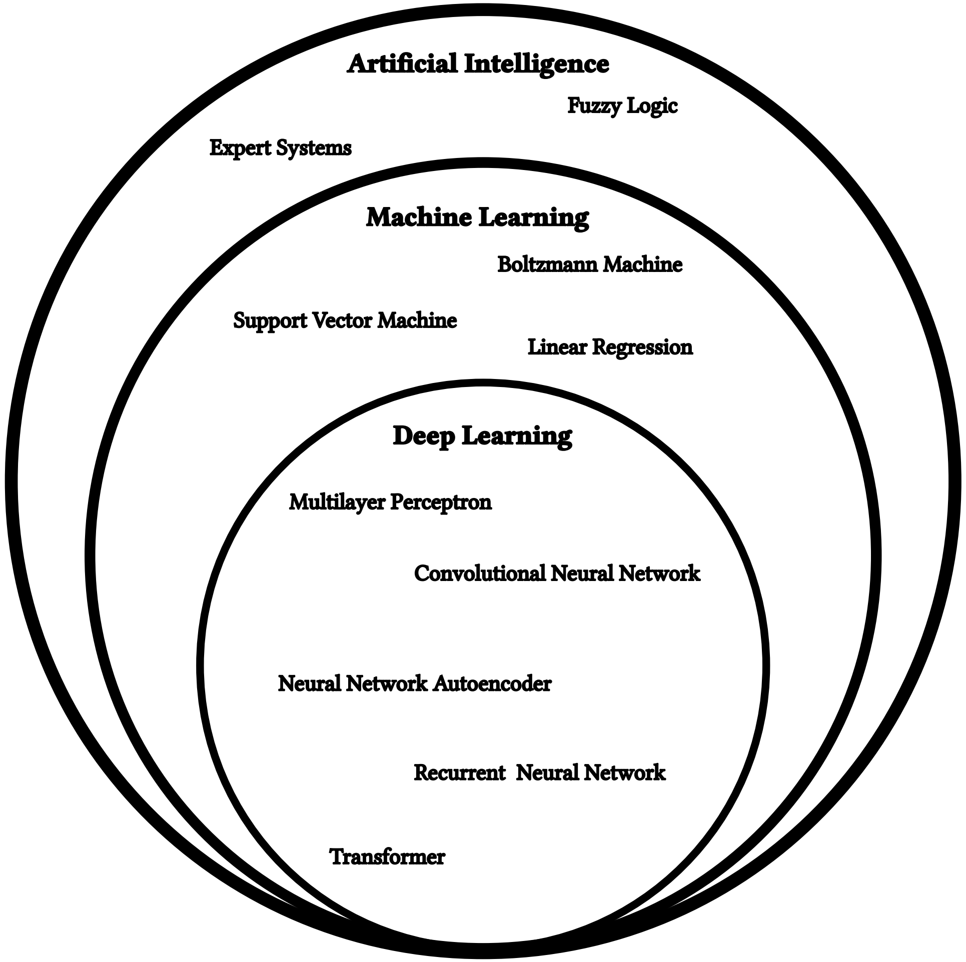
AI, ML, DL



Deep Learning is a type of Machine Learning

Machine learning defines **what** we are trying to do

AI, ML, DL

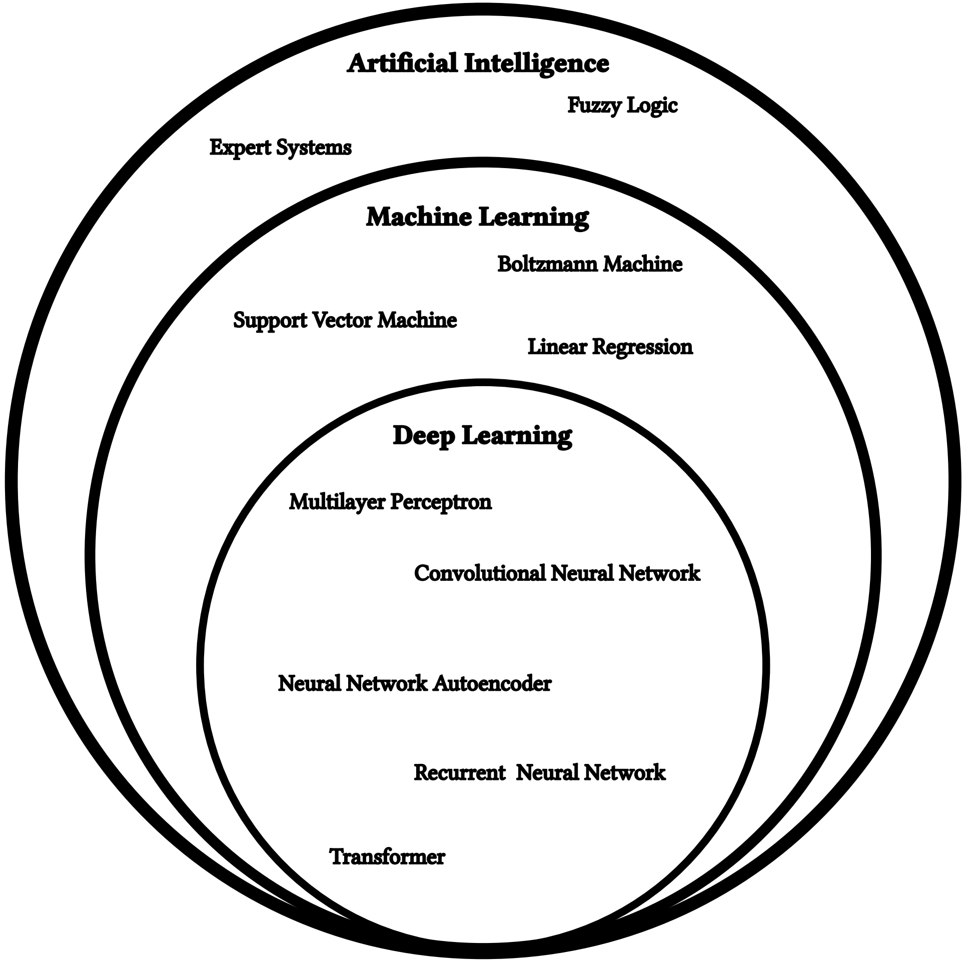


Deep Learning is a type of Machine Learning

Machine learning defines **what** we are trying to do

Deep learning defines **how** we do it

AI, ML, DL



Deep Learning is a type of Machine Learning

Machine learning defines **what** we are trying to do

Deep learning defines **how** we do it

Before we can understand deep learning, we must become familiar with machine learning

Overview

Overview

1. Deep Learning Successes
2. What is Deep Learning?
3. Differences between AI, ML, DL
4. **Define Machine Learning**
5. Machine Learning Libraries

ML

Task: Write a program to determine if a picture contains a dog.

Task: Write a program to determine if a picture contains a dog.

Question: How would you program this?

Task: Write a program to determine if a picture contains a dog.

Question: How would you program this?

Task: Write a program to determine if a picture contains a dog.

Question: How would you program this?

Would your method still work?

Task: Write a program to determine if a picture contains a dog.

Question: How would you program this?

Would your method still work?

We often know **what** we want, but we do not know **how**

We give machine learning the **what**

ML

We give machine learning the **what**

And it tells us the **how**

ML

We give machine learning the **what**

And it tells us the **how**

In other words, we tell an ML model **what** we want it to do

ML

We give machine learning the **what**

And it tells us the **how**

In other words, we tell an ML model **what** we want it to do

And it learns **how** to do it

We often know **what** we want, but we do not know **how**

ML

We often know **what** we want, but we do not know **how**

We have many pictures of either dogs or muffins $x \in X$

ML

We often know **what** we want, but we do not know **how**

We have many pictures of either dogs or muffins $x \in X$

We want to know if the picture is [dog | muffin] $y \in Y$

ML

We often know **what** we want, but we do not know **how**

We have many pictures of either dogs or muffins $x \in X$

We want to know if the picture is [dog | muffin] $y \in Y$

We learn a function or mapping from X to Y

ML

We often know **what** we want, but we do not know **how**

We have many pictures of either dogs or muffins $x \in X$

We want to know if the picture is [dog | muffin] $y \in Y$

We learn a function or mapping from X to Y

$$f : X \mapsto Y$$

ML

We often know **what** we want, but we do not know **how**

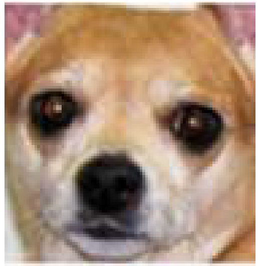
We have many pictures of either dogs or muffins $x \in X$

We want to know if the picture is [dog | muffin] $y \in Y$

We learn a function or mapping from X to Y

$$f : X \mapsto Y$$

Machine learning tells us how to find f

$f($  $) = \text{dog}$

$f($  $) = \text{muffin}$

Consider some more interesting functions

Consider some more interesting functions

$$f : \text{image} \mapsto \text{caption}$$

Consider some more interesting functions

$$f : \text{image} \mapsto \text{caption}$$

$$f : \text{caption} \mapsto \text{image}$$

Consider some more interesting functions

$$f : \text{image} \mapsto \text{caption}$$

$$f : \text{caption} \mapsto \text{image}$$

$$f : \text{English} \mapsto \text{Chinese}$$

Consider some more interesting functions

$$f : \text{image} \mapsto \text{caption}$$

$$f : \text{caption} \mapsto \text{image}$$

$$f : \text{English} \mapsto \text{Chinese}$$

$$f : \text{law} \mapsto \text{change in climate}$$

ML

Consider some more interesting functions

$$f : \text{image} \mapsto \text{caption}$$

$$f : \text{caption} \mapsto \text{image}$$

$$f : \text{English} \mapsto \text{Chinese}$$

$$f : \text{law} \mapsto \text{change in climate}$$

$$f : \text{voice command} \mapsto \text{robot action}$$

ML

Consider some more interesting functions

$$f : \text{image} \mapsto \text{caption}$$

$$f : \text{caption} \mapsto \text{image}$$

$$f : \text{English} \mapsto \text{Chinese}$$

$$f : \text{law} \mapsto \text{change in climate}$$

$$f : \text{voice command} \mapsto \text{robot action}$$

Question: Can anyone suggest other interesting functions?

ML

Why do we call it machine **learning**?

Why do we call it machine **learning**?

We learn the function f from the **data** $x \in X, y \in Y$

Why do we call it machine **learning**?

We learn the function f from the **data** $x \in X, y \in Y$

More specifically, we learn function **parameters** $\boldsymbol{\Theta}$

ML

Why do we call it machine **learning**?

We learn the function f from the **data** $x \in X, y \in Y$

More specifically, we learn function **parameters** Θ

$$f : X, \Theta \mapsto Y$$

More specifically, we learn function **parameters** Θ

$$f : X, \Theta \mapsto Y$$

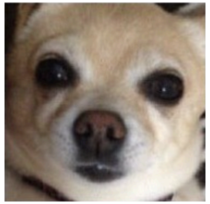
More specifically, we learn function **parameters** Θ

$$f : X, \Theta \mapsto Y$$

$$f\left(\text{你好吗}, \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \end{pmatrix}\right) = \text{You good?}$$

More specifically, we learn function **parameters** Θ

$$f : X, \Theta \mapsto Y$$

$f \Bigg($  $, \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \end{pmatrix} \Bigg) = \text{Dog}$

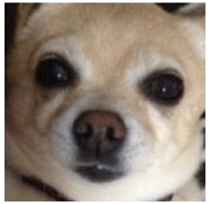
More specifically, we learn function **parameters** Θ

$$f : X, \Theta \mapsto Y$$

$$f\left(\square, \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \end{pmatrix} \right) = \text{Muffin}$$

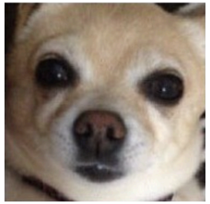
More specifically, we learn function **parameters** Θ

$$f : X, \Theta \mapsto Y$$

$$f\left(\text{Dog}, \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \end{pmatrix} \right) =$$ 

More specifically, we learn function **parameters** Θ

$$f : X, \Theta \mapsto Y$$

$$f\left(\text{Dog}, \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \end{pmatrix}\right) =$$ 

Machine learning learns the parameters that solve difficult problems

Summary:

Summary:

1. Certain problems are difficult to solve with programming

Summary:

1. Certain problems are difficult to solve with programming
   - Dog or muffin?

Summary:

1. Certain problems are difficult to solve with programming
   - Dog or muffin?
2. Machine learning provides a framework to solve difficult problems

Summary:

1. Certain problems are difficult to solve with programming
   - Dog or muffin?
2. Machine learning provides a framework to solve difficult problems
   - We learn the parameters θ for some function $f(x, \theta) = y$

Relax

You will use Python with machine learning libraries in this course

ML

You will use Python with machine learning libraries in this course

We will specifically focus on:

ML

You will use Python with machine learning libraries in this course

We will specifically focus on:

- JAX

ML

You will use Python with machine learning libraries in this course

We will specifically focus on:

- JAX
- PyTorch

ML

You will use Python with machine learning libraries in this course

We will specifically focus on:
- JAX
- PyTorch

You should become comfortable using these libraries

ML

You will use Python with machine learning libraries in this course

We will specifically focus on:

- JAX
- PyTorch

You should become comfortable using these libraries

- Read tutorials online

ML

You will use Python with machine learning libraries in this course

We will specifically focus on:
- JAX
- PyTorch

You should become comfortable using these libraries
- Read tutorials online
- Play with the libraries

Both JAX and PyTorch are libraries for the Python language

Both JAX and PyTorch are libraries for the Python language

They are both based on `numpy`, which itself is based on `MATLAB`

Both JAX and PyTorch are libraries for the Python language

They are both based on `numpy`, which itself is based on `MATLAB`

Both libraries are **tensor processing** libraries

Both JAX and PyTorch are libraries for the Python language

They are both based on `numpy`, which itself is based on `MATLAB`

Both libraries are **tensor processing** libraries

- Designed for linear algebra and taking derivatives

Both JAX and PyTorch are libraries for the Python language

They are both based on `numpy`, which itself is based on `MATLAB`

Both libraries are **tensor processing** libraries

- Designed for linear algebra and taking derivatives

To install, use `pip`

Both JAX and PyTorch are libraries for the Python language

They are both based on `numpy`, which itself is based on `MATLAB`

Both libraries are **tensor processing** libraries

- Designed for linear algebra and taking derivatives

To install, use `pip`

- `pip install torch`

Both JAX and PyTorch are libraries for the Python language

They are both based on `numpy`, which itself is based on `MATLAB`

Both libraries are **tensor processing** libraries

- Designed for linear algebra and taking derivatives

To install, use `pip`

- `pip install torch`
- `pip install jax jaxlib`

Create vectors, matrices, or tensors in `jax`

```
import jax.numpy as jnp
a = jnp.array(1) # Scalar
b = jnp.array([1, 2]) # Vector
C = jnp.array([[1,2], [3,4]]) # 2x2 Matrix
D = jnp.ones((3,3,3)) # 3x3x3 Tensor
```

You can determine the dimensions of a variable using `shape`

```
b.shape # Prints (2,)
C.shape # Prints (2,2)
D.shape # prints (3,3,3)
```

Create vectors, matrices, or tensors in `pytorch`

```
import torch
a = torch.tensor(1) # Scalar
b = torch.tensor([1, 2]) # Vector
C = torch.tensor([[1,2], [3,4]]) # 2x2 Matrix
D = torch.ones((3,3,3)) # 3x3x3 Tensor
```

You can determine the dimensions of a variable using `shape`

```
b.shape # Prints (2,)
C.shape # Prints (2,2)
D.shape # prints (3,3,3)
```

Most operations in `jax` and `pytorch` are **vectorized**

Most operations in `jax` and `pytorch` are **vectorized**

- Executed in parallel, very fast

Most operations in `jax` and `pytorch` are **vectorized**

- Executed in parallel, very fast

```
import jax.numpy as jnp

s = 5 * jnp.array([1, 2])
print(s) # jnp.array(5, 10)
x = jnp.array([1, 2]) + jnp.array([3, 4])
print(x) # jnp.array([4, 6])
y = jnp.array([1, 2]) * jnp.array([3, 4]) # Careful!
print(y) # jnp.array([3, 8])
z = jnp.array([[1], [2]]) @ jnp.array([[3, 4]])
print(z) # A^t B (dot product), jnp.array([[11]])
```

`pytorch` is very similar to `jax`

```python
import torch

s = 5 * torch.tensor([1, 2])
print(s) # torch.tensor(5, 10)
x = torch.tensor([1, 2]) + torch.tensor([3, 4])
print(x) # torch.tensor([4, 6])
y = torch.tensor([1, 2]) * torch.tensor([3, 4]) # Careful!
print(y) # torch.tensor([3, 8])
z = torch.tensor([[1], [2]]) @ torch.tensor([[3, 4]])
print(z) # A^t B (dot product), torch.tensor([[11]])
```

You can also call various methods on arrays/tensors

```
import jax.numpy as jnp

x = jnp.array([[1, 2], [3, 4]]).sum(axis=0)
print(x) # Sum across leading axis, array([4, 6])
y = jnp.array([[1, 2], [3, 4]]).mean()
print(y) # Mean across all axes, array(2.5)
z = jnp.array([[1, 2], [3, 4]]).reshape(4, 1)
print(z) # jnp.array([[1], [2], [3], [4]])
```

Same thing for `pytorch`

```python
import torch

x = torch.tensor([[1, 2], [3, 4]]).sum(axis=0)
print(x) # Sum across leading axis, array([4, 6])
y = torch.tensor([[1, 2], [3, 4]]).mean()
print(y) # Mean across all axes, array(2.5)
z = torch.tensor([[1, 2], [3, 4]]).reshape(4, 1)
print(z) # torch.tensor([[1], [2], [3], [4]])
```

These libraries can produce tricky error messages!

These libraries can produce tricky error messages!

```
  >>> jnp.array([[1,2]]) @ jnp.array([[3,4]]) # (1, 2) x (1, 2)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/local/scratch/sm2558/miniconda3/envs/jax/lib/python3.11/site-
packages/jax/_src/numpy/array_methods.py", line 256, in deferring_binary_op
    return binary_op(*args)
           ^^^^^^^^^^^^^^^^

  File "/local/scratch/sm2558/miniconda3/envs/jax/lib/python3.11/site-
packages/jax/_src/numpy/lax_numpy.py", line 3192, in matmul
    out = lax.dot_general(
          ^^^^^^^^^^^^^^^^

TypeError: dot_general requires contracting dimensions to have the same
shape, got (2,) and (1,).
```

Let us do a Google Colab tutorial!

Let us set up a local `conda` environment

Homework:

Homework:

- Review linear algebra if you are not familiar

Homework:

- Review linear algebra if you are not familiar
- Install `pytorch` and `jax` onto your computer, or use Google Colab

Homework:

- Review linear algebra if you are not familiar
- Install `pytorch` and `jax` onto your computer, or use Google Colab
- Play around with them

Homework:

- Review linear algebra if you are not familiar
- Install `pytorch` and `jax` onto your computer, or use Google Colab
- Play around with them
  - What does `@` do if you have a `(3, 3, 3)` tensor?

Homework:

- Review linear algebra if you are not familiar
- Install `pytorch` and `jax` onto your computer, or use Google Colab
- Play around with them
  - What does `@` do if you have a `(3, 3, 3)` tensor?
  - What does `*` do if one tensor has fewer dimensions?

Homework:

- Review linear algebra if you are not familiar
- Install `pytorch` and `jax` onto your computer, or use Google Colab
- Play around with them
  - What does `@` do if you have a `(3, 3, 3)` tensor?
  - What does `*` do if one tensor has fewer dimensions?
  - How do you invert a matrix?

Homework:

- Review linear algebra if you are not familiar
- Install `pytorch` and `jax` onto your computer, or use Google Colab
- Play around with them
  - What does `@` do if you have a `(3, 3, 3)` tensor?
  - What does `*` do if one tensor has fewer dimensions?
  - How do you invert a matrix?
  - What does `jnp.arange, jnp.zeros, jnp.full` do?

Homework:

- Review linear algebra if you are not familiar
- Install `pytorch` and `jax` onto your computer, or use Google Colab
- Play around with them
  - What does `@` do if you have a `(3, 3, 3)` tensor?
  - What does `*` do if one tensor has fewer dimensions?
  - How do you invert a matrix?
  - What does `jnp.arange, jnp.zeros, jnp.full` do?
    - `torch.arange, torch.zeros, torch.full`

Homework:

- Review linear algebra if you are not familiar
- Install `pytorch` and `jax` onto your computer, or use Google Colab
- Play around with them
  - What does `@` do if you have a `(3, 3, 3)` tensor?
  - What does `*` do if one tensor has fewer dimensions?
  - How do you invert a matrix?
  - What does `jnp.arange, jnp.zeros, jnp.full` do?
    - `torch.arange, torch.zeros, torch.full`
- Read the documentation

There might be a quiz on `jax` and `pytorch` operations next lecture