

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

This is a learning experience for me, too

Brief Chat

This is my first course at UM

I taught a course on Deep Reinforcement Learning at Cambridge

This is a learning experience for me, too

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

This is a learning experience for me, too

Please provide feedback privately to me

- Email
- Chat after class

Brief Chat

This is my first course at UM

I taught a course on Deep Reinforcement Learning at Cambridge

This is a learning experience for me, too

Please provide feedback privately to me

- Email
- 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

This is a learning experience for me, too

Please provide feedback privately to me

- Email
- 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 would like to get an understanding of why you have enrolled in this course/program.

Brief Chat

I would like to get an understanding of why you have enrolled in this course/program.

It will help me alter the course towards your goals.

Brief Chat

I would like to get an understanding of why you have enrolled in this course/program.

It will help me alter the course towards your goals.

First, let me introduce myself and my interests in deep learning.

Brief Chat

I am interested in Deep Learning because I think it is the most promising approach for automating tasks that we do not want to do

Brief Chat

I am interested in Deep Learning because I think it is the most promising approach for automating tasks that we do not want to do

I am interested in **deep reinforcement learning and memory**

Brief Chat

I am interested in Deep Learning because I think it is the most promising approach for automating tasks that we do not want to do

I am interested in **deep reinforcement learning and memory**

- Reinforcement learning trains agents that know how to interact with the world

Brief Chat

I am interested in Deep Learning because I think it is the most promising approach for automating tasks that we do not want to do

I am interested in **deep reinforcement learning and memory**

- Reinforcement learning trains agents that know how to interact with the world
- Memory gives them the ability to remember your favorite food, where the agents live, etc

Brief Chat

I am interested in Deep Learning because I think it is the most promising approach for automating tasks that we do not want to do

I am interested in **deep reinforcement learning and memory**

- Reinforcement learning trains agents that know how to interact with the world
- Memory gives them the ability to remember your favorite food, where the agents live, etc

I am interested in this course because it enables me to continue my research

Brief Chat

I am interested in Deep Learning because I think it is the most promising approach for automating tasks that we do not want to do

I am interested in **deep reinforcement learning and memory**

- Reinforcement learning trains agents that know how to interact with the world
- Memory gives them the ability to remember your favorite food, where the agents live, etc

I am interested in this course because it enables me to continue my research

What are you here to learn? What interests you?

Brief Chat

I am looking for PhD students focusing on deep reinforcement learning and robotics

Brief Chat

I am looking for PhD students focusing on deep reinforcement learning and robotics

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

Brief Chat

I am looking for PhD students focusing on deep reinforcement learning and robotics

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

Strong background in robotics is desired

Brief Chat

I am looking for PhD students focusing on deep reinforcement learning and robotics

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

Strong background in robotics is desired

Overview

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

Overview

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

Prerequisites:

Prerequisites:

- Programming in python

Prerequisites:

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

Prerequisites:

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

Prerequisites:

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

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

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 $\begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \dots \end{bmatrix}^\top$

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 $\begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \dots \end{bmatrix}^\top$

Good to Know:

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 $\begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \dots \end{bmatrix}^\top$

Good to Know:

- Probability

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 $\begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \dots \end{bmatrix}^\top$

Good to Know:

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

Grading:

Grading:

- 80% assignments

Grading:

- 80% assignments
- 20% attendance and participation

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 20-45 minutes

Course Structure

3 hours is a very long time to pay attention

I will place slides titled “Relax”, every 20-45 minutes

At the Relax slides, we may take a 5-15 minute break

Course Structure

3 hours is a very long time to pay attention

I will place slides titled “Relax”, every 20-45 minutes

At the Relax slides, we may take a 5-15 minute break

You can:

Course Structure

3 hours is a very long time to pay attention

I will place slides titled “Relax”, every 20-45 minutes

At the Relax slides, we may take a 5-15 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 20-45 minutes

At the Relax slides, we may take a 5-15 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 20-45 minutes

At the Relax slides, we may take a 5-15 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 20-45 minutes

At the Relax slides, we may take a 5-15 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 20-45 minutes

At the Relax slides, we may take a 5-15 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 touching on the following topics (subject to change)

Course Structure

We will be touching on the following topics (subject to change)

- Introduction to machine learning

Course Structure

We will be touching on the following topics (subject to change)

- Introduction to machine learning
- Linear regression and classification

Course Structure

We will be touching on the following topics (subject to change)

- Introduction to machine learning
- Linear regression and classification
- Neural Networks and the Perceptron

Course Structure

We will be touching on the following topics (subject to change)

- Introduction to machine learning
- Linear regression and classification
- Neural Networks and the Perceptron
- Convolutional Neural Networks

Course Structure

We will be touching on the following topics (subject to change)

- Introduction to machine learning
- Linear regression and classification
- Neural Networks and the Perceptron
- Convolutional Neural Networks
- Recurrent Neural Networks

Course Structure

We will be touching on the following topics (subject to change)

- Introduction to machine learning
- Linear regression and classification
- Neural Networks and the Perceptron
- Convolutional Neural Networks
- Recurrent Neural Networks
- Attention and Transformers

Course Structure

We will be touching on the following topics (subject to change)

- Introduction to machine learning
- Linear regression and classification
- Neural Networks and the Perceptron
- Convolutional Neural Networks
- Recurrent Neural Networks
- Attention and Transformers
- Reinforcement Learning

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

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. Supervised and Unsupervised Learning

Overview

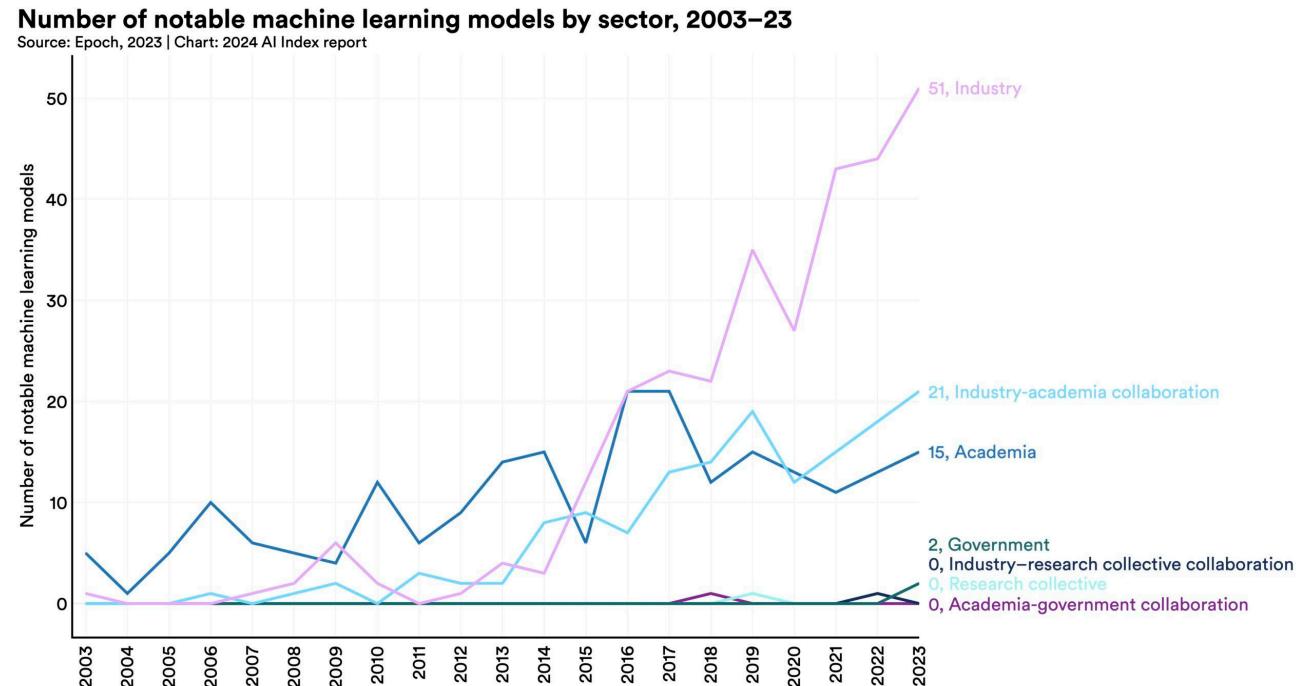
1. Deep Learning Successes
2. What is Deep Learning?
3. Differences between AI, ML, DL
4. Define Machine Learning
5. Supervised and Unsupervised Learning
6. Classification and Regression

Overview

1. Deep Learning Successes
2. What is Deep Learning?
3. Differences between AI, ML, DL
4. Define Machine Learning
5. Supervised and Unsupervised Learning
6. Classification and Regression

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



Successes

It is operating fully autonomous taxis in four cities



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
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:

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

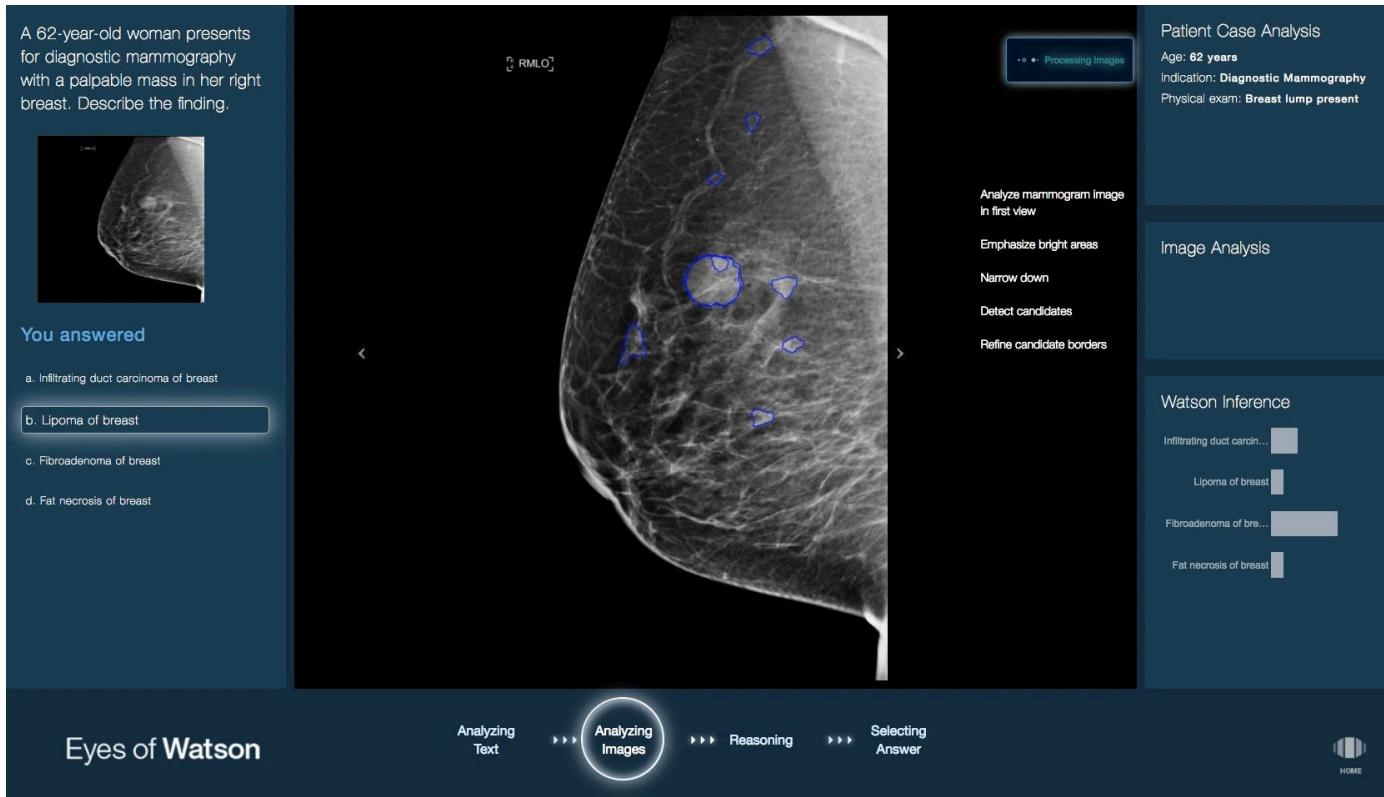
Successes

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 could 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. Supervised and Unsupervised Learning
6. Classification and Regression

Overview

1. Deep Learning Successes
2. **What is Deep Learning?**
3. Differences between AI, ML, DL
4. Define Machine Learning
5. Supervised and Unsupervised Learning
6. Classification and Regression

DL at a Glance

At a high level, how does deep learning work?

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

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

At a high level, how does deep learning work?

It consists of four parts:

1. Dataset
2. Deep neural network
3. Loss function
4. Optimizer

At a high level, how does deep learning work?

It consists of four parts:

1. Dataset
2. **Deep neural network**
3. Loss function
4. Optimizer

DL at a Glance

The deep neural network learns to map the inputs to outputs



At a high level, how does deep learning work?

It consists of four parts:

1. Dataset
2. Deep neural network
3. Loss function
4. Optimizer

At a high level, how does deep learning work?

It consists of four parts:

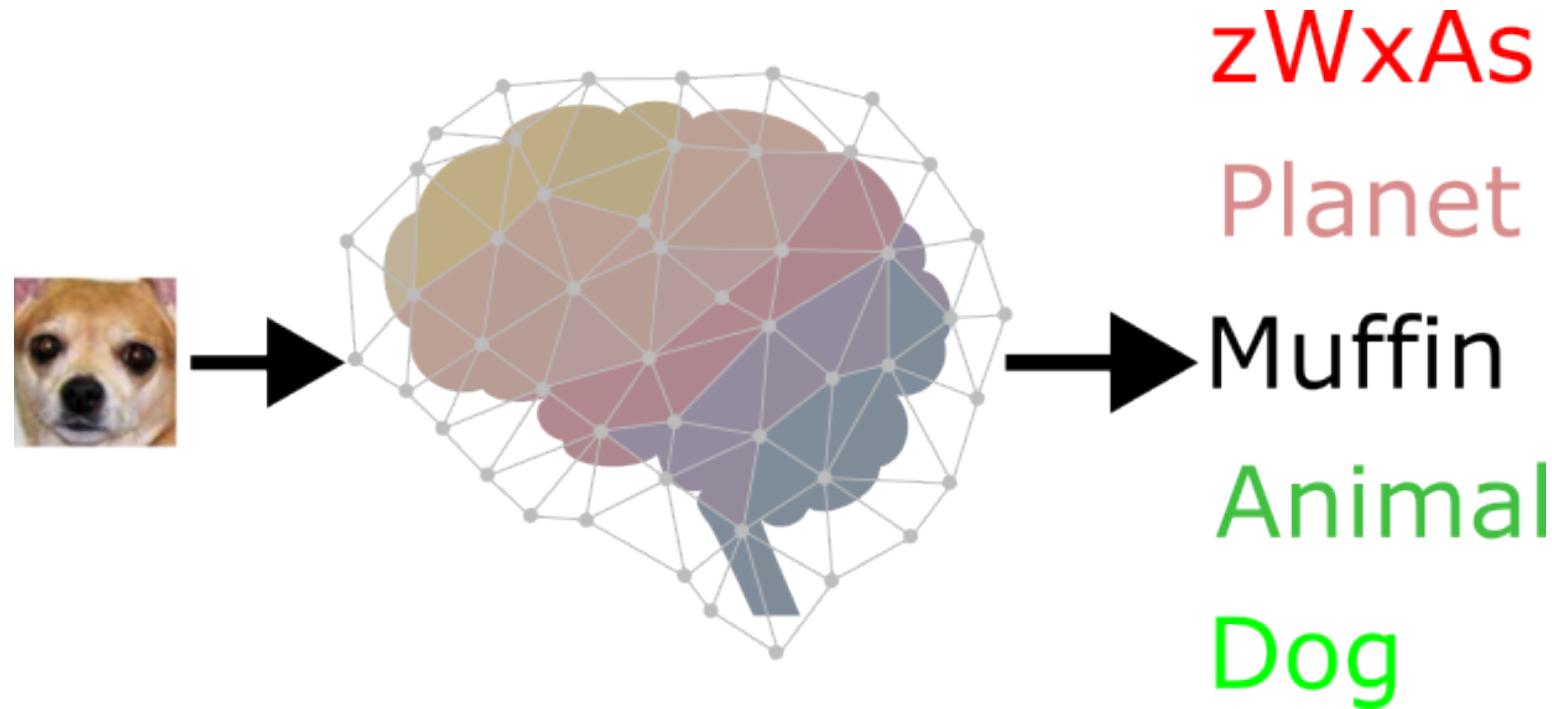
1. Dataset
2. Deep neural network
3. **Loss function**
4. Optimizer

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**.



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

At a high level, how does deep learning work?

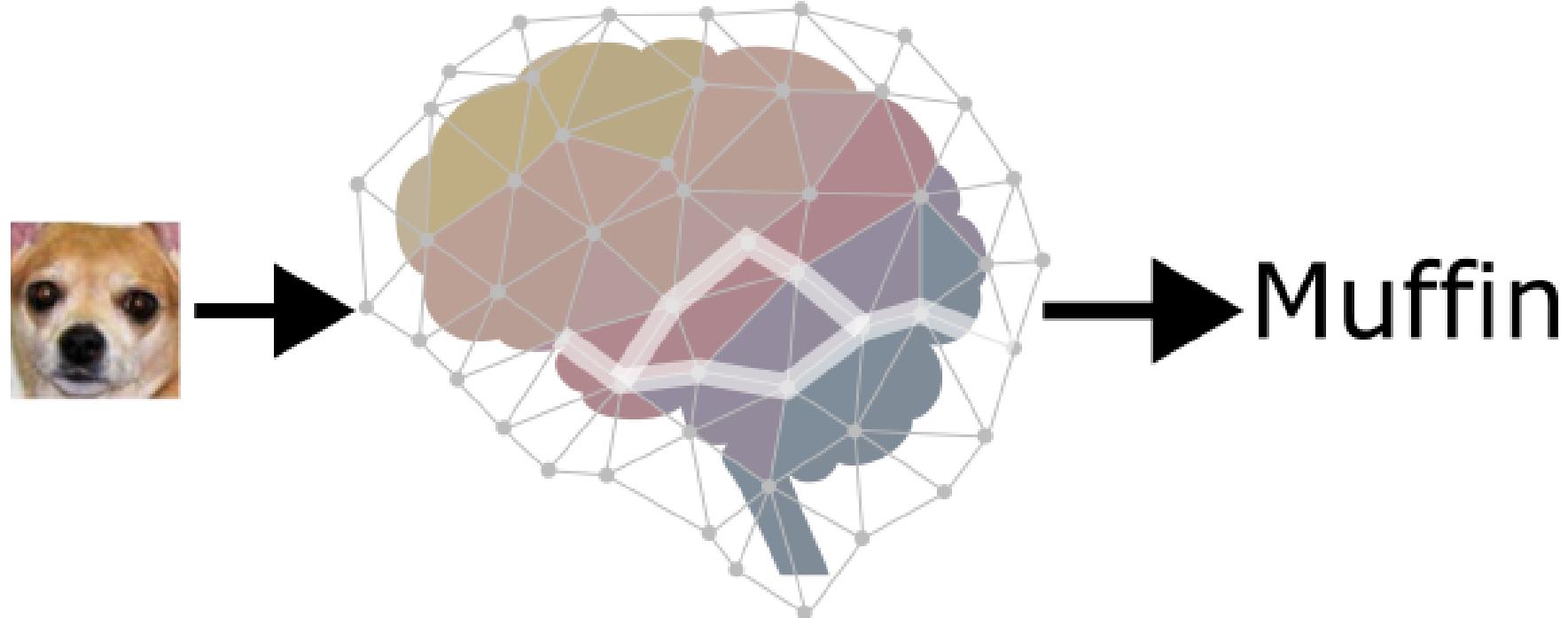
It consists of four parts:

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

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



At a high level, how does deep learning work?

It consists of four parts:

1. Dataset
2. Deep neural network
3. Loss function
4. Optimizer

At a high level, how does deep learning work?

It consists of four parts:

1. Dataset
2. Deep neural network
3. Loss function
4. Optimizer

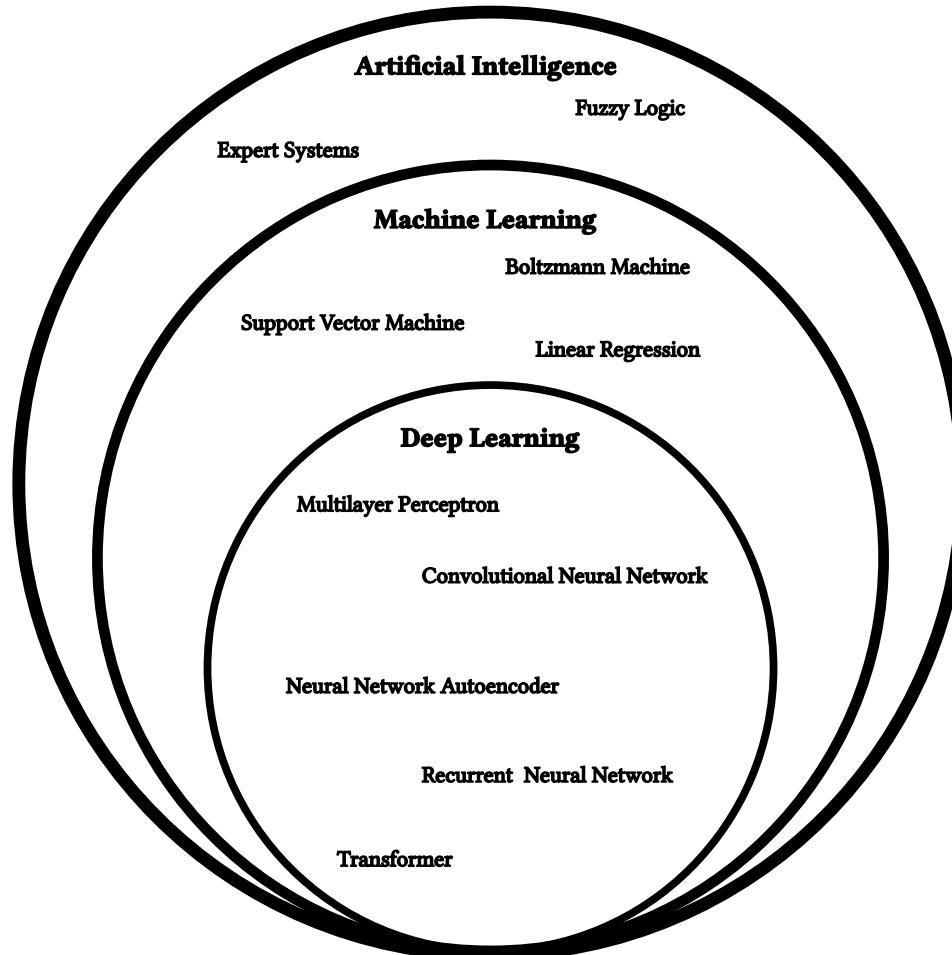
Questions so far?

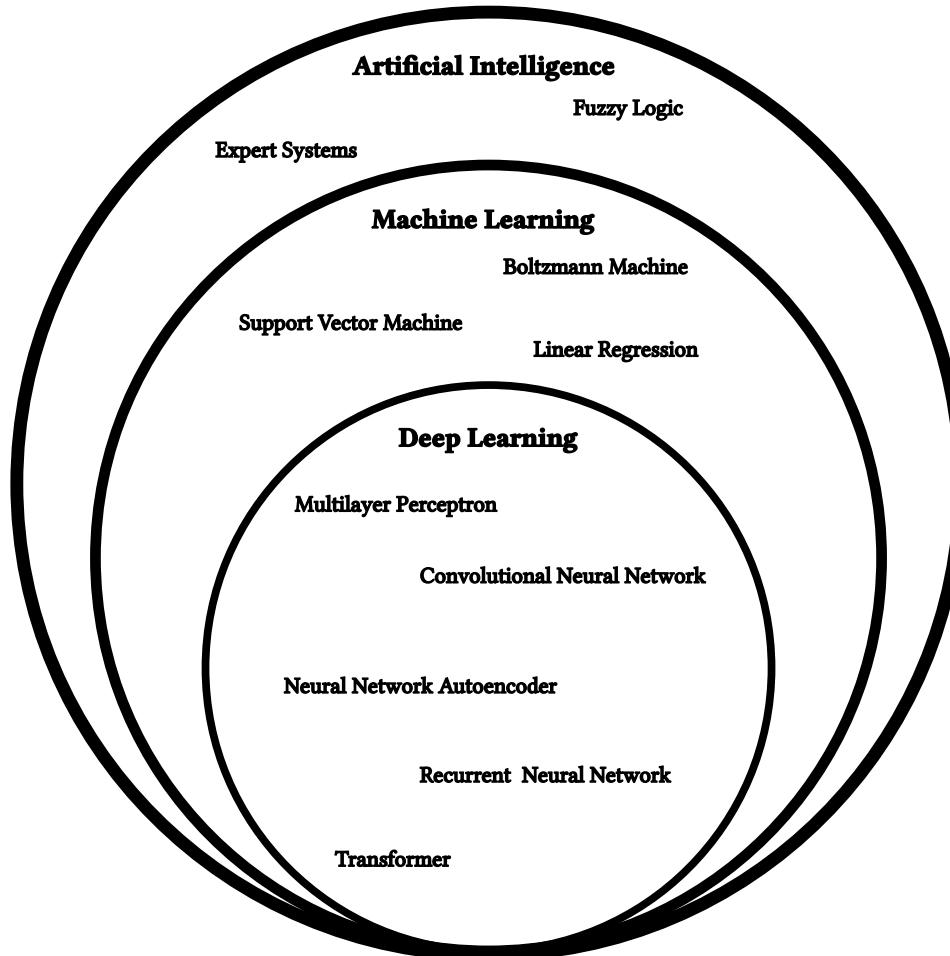
Overview

1. Deep Learning Successes
2. **What is Deep Learning?**
3. Differences between AI, ML, DL
4. Define Machine Learning
5. Supervised and Unsupervised Learning
6. Classification and Regression

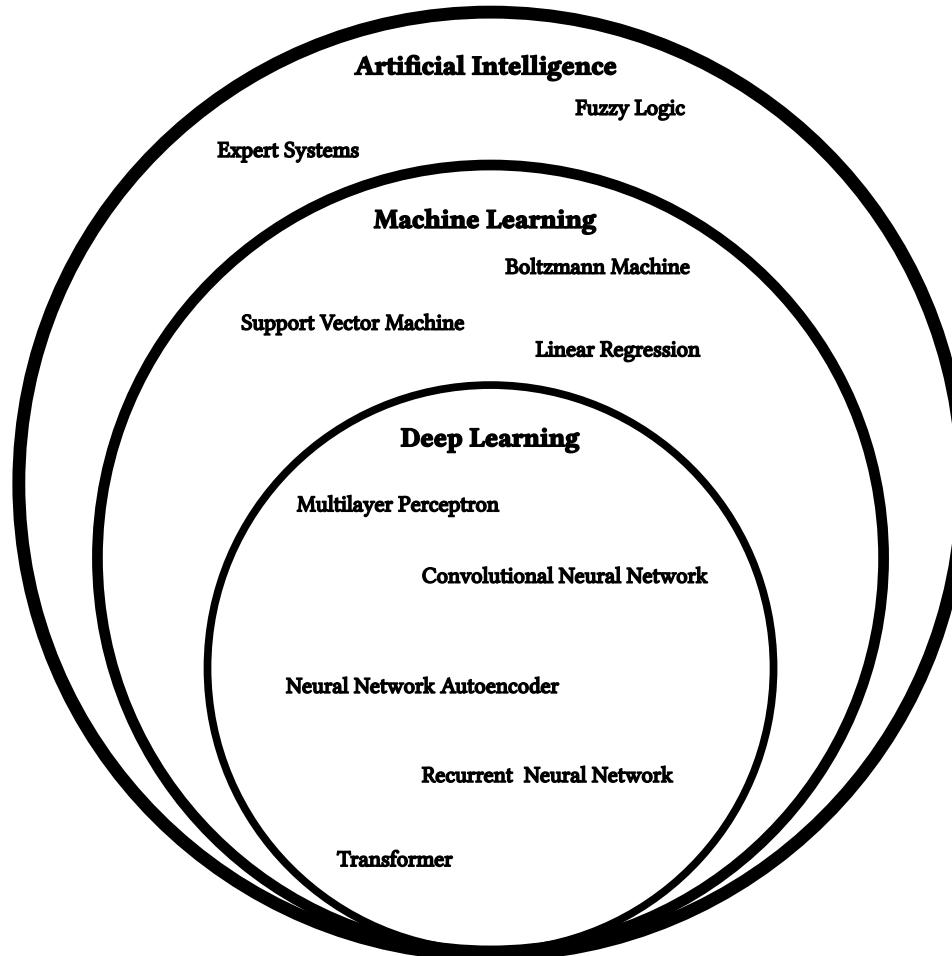
Overview

1. Deep Learning Successes
2. What is Deep Learning?
3. **Differences between AI, ML, DL**
4. Define Machine Learning
5. Supervised and Unsupervised Learning
6. Classification and Regression



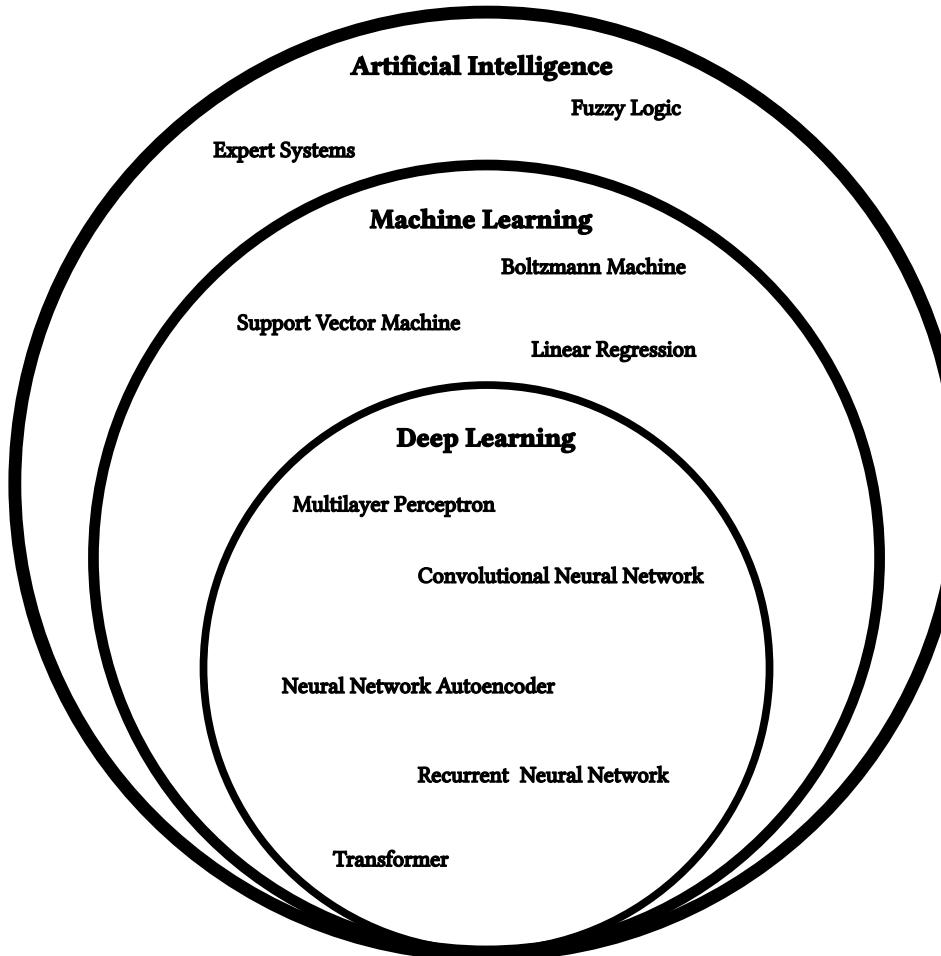


Deep Learning is a type of
Machine Learning



Deep Learning is a type of Machine Learning

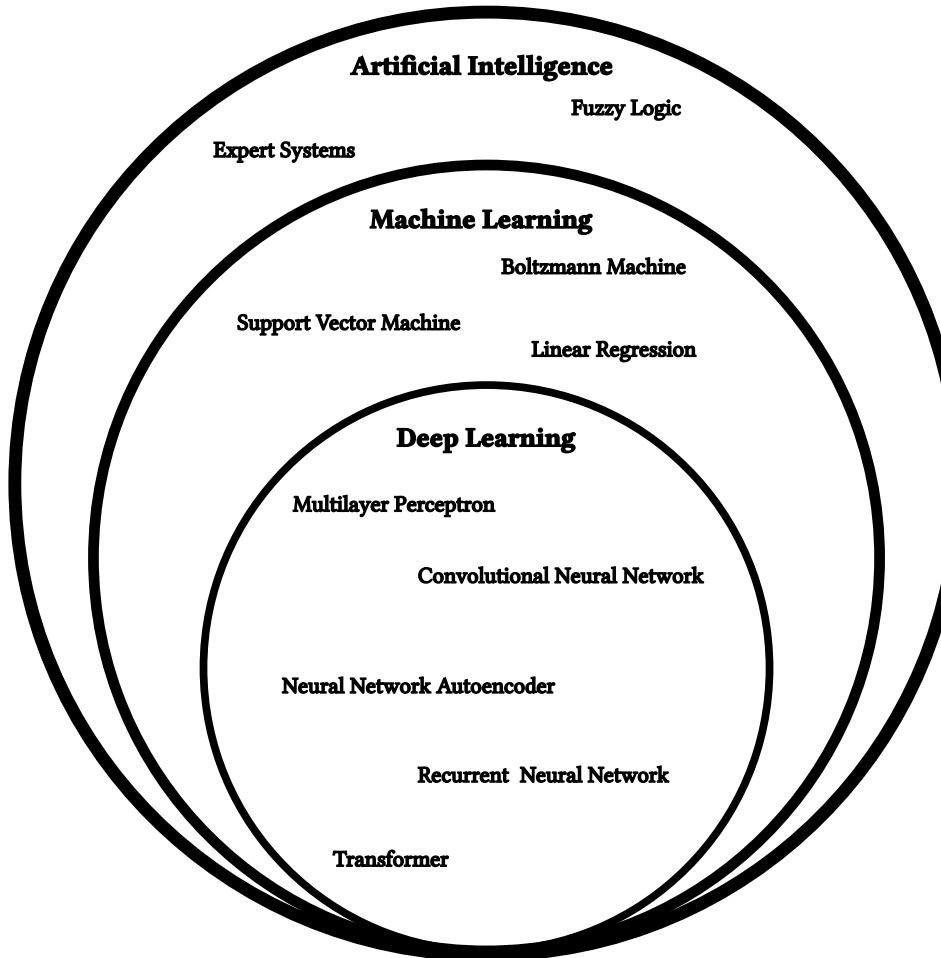
Machine learning defines **what** we are trying to do



Deep Learning is a type of Machine Learning

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

Deep learning defines **how** we do it



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

1. Deep Learning Successes
2. What is Deep Learning?
3. **Differences between AI, ML, DL**
4. Define Machine Learning
5. Supervised and Unsupervised Learning
6. Classification and Regression

Relax

Overview

1. Deep Learning Successes
2. What is Deep Learning?
3. Differences between AI, ML, DL
4. **Define Machine Learning**
5. Supervised and Unsupervised Learning
6. Classification and Regression

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**

We give machine learning the **what**

And it tells us the **how**

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

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**

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 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 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

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

$$f : X \mapsto Y$$

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

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

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

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

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

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

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

Consider some more interesting functions

$$f : \text{image} \mapsto \text{caption}$$
$$f : \text{caption} \mapsto \text{image}$$
$$f : \text{English} \mapsto \text{Chinese}$$
$$f : \text{law} \mapsto \text{change in climate}$$
$$f : \text{voice command} \mapsto \text{robot action}$$

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?

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** Θ

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$f\left(\text{Dog}, \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \end{bmatrix}\right) = \img[alt="A small brown dog's face." data-bbox="568 368 662 525]$$

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

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

$$f\left(\text{Dog}, \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \end{bmatrix}\right) = \img[alt="A small brown dog's face." data-bbox="568 368 662 525]$$

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 tells us how to solve difficult problems

Summary:

1. Certain problems are difficult to solve with programming
 - Dog or muffin?
2. Machine learning tells us how 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

You will use Python with machine learning libraries in this course

We will specifically focus on:

You will use Python with machine learning libraries in this course

We will specifically focus on:

- JAX

You will use Python with machine learning libraries in this course

We will specifically focus on:

- JAX
- PyTorch

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

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

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

1. JAX vs PyTorch
 1. Based on numpy and matlab
2. Matrix operations

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]) # 2D 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]) # 2D 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

```
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

```
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

```
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,).
```

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
 - ▶ Bonus: einsum