

# Lecture 01: Introduction

## Introduction to Machine Learning

Sajjad Amini

Department of Electrical Engineering  
Sharif University of Technology

# Contents

## 1 Course Introduction

- State of Machine Learning

## 2 Supervised Learning

- Classification
- Regression
- Generalization
- Real World Applications

## 3 Unsupervised Learning

- Clustering
- Factors of Variation
- Real World Applications

## 4 Reinforcement Learning

- Real World Applications

# References

Except explicitly cited, the reference for the material in slides is:

- Murphy, K. P. (2022). *Probabilistic machine learning: an introduction.* MIT press.

# Color Guided Blocks

Definition Block

Result Block

Note Block

Example Block

Remember Block

## Section 1

### Course Introduction

## Subsection 1

### State of Machine Learning

# State of Machine Learning



Andrew Ng

Electricity transforms countless industries: transportation, manufacturing, healthcare, communications and more. AI (machine learning) will bring about an equally big transformation

# What is Machine Learning

## Machine Learning [1]

Consider the following three items:

- Experience  $E$
- Class of tasks  $T$
- Performance measure  $P$

Machine learning is to improve the performance measured by  $P$  of a computer program on  $T$  using  $E$

## Machine Learning

Based on *Machine Learning*, definition we have the main following major types of machine learning:

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

## Section 2

### Supervised Learning

# Supervised Learning

## Supervised Learning

Supervised Learning is:

- Task  $T$ : Finding mapping  $f : \mathbf{x} \mapsto \mathbf{y}$  ( $\mathbf{x} \in \mathcal{X} = \mathbb{R}^D$  and  $\mathbf{y} \in \mathcal{Y}$ )
  - $\mathbf{x}$ : Features, Covariates or Predictors
  - $\mathbf{y}$ : Label, Target or Response
- Experience  $E$ : Set of  $N$  input-output pairs  $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$ 
  - $\mathcal{D}$ : Dataset
  - $N$ : Sample size
- Performance measure  $P$ : Dependent on the task

## Subsection 1

### Classification

# Supervised Learning - Classification

## Classification

General Features:

- Task  $T$ : Finding mapping  $f : \mathbf{x} \mapsto y$  ( $\mathbf{x} \in \mathcal{X} = \mathbb{R}^D$  and  $y \in \mathcal{Y}$ )
- Experience  $E$ : Set of  $N$  input-output pairs  $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$

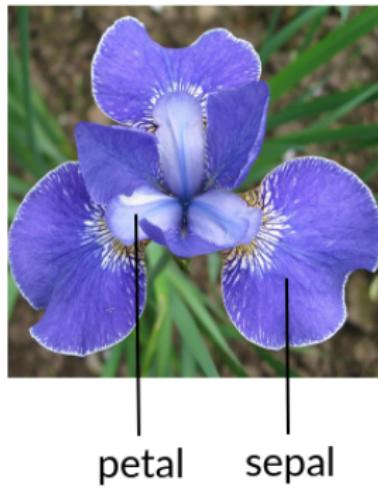
Specific Features:

- $\mathcal{Y} = \{1, 2, \dots, C\}$  (Unordered and mutually exclusive labels)
- $P = \frac{1}{N} \sum_{n=1}^N \mathbb{I}(y_n \neq f(\mathbf{x}_n))$  where:

$$\mathbb{I}(e) = \begin{cases} 1 & \text{if } e \text{ is true} \\ 0 & \text{if } e \text{ is false} \end{cases}$$

- $P$  is known as *Misclassification Error*

## Example - Iris Flower Classification



(a) Setosa ( $y = 1$ )



(b) Versicolor ( $y = 2$ )



(c) Virginica ( $y = 3$ )

Figure: Different types of Iris flower

# Example - Iris Flower Classification

$$f_1\left( \begin{array}{c} \text{Iris flower image} \end{array} \right) = \begin{cases} 1 (\text{Setosa}) \\ 2 (\text{Versicolor}) \\ 3 (\text{Virginica}) \end{cases}$$

(a) Approach 1

$$\underbrace{f_2\left(g\left(\begin{array}{c} \text{Sepal length} \\ \text{Sepal width} \\ \text{Petal length} \\ \text{Petal width} \end{array}\right)\right)}_{\text{Iris flower image}} = \begin{cases} 1 (\text{Setosa}) \\ 2 (\text{Versicolor}) \\ 3 (\text{Virginica}) \end{cases}$$

(b) Approach 2

Figure: Different types of Iris flower

# Example - Iris Flower Classification

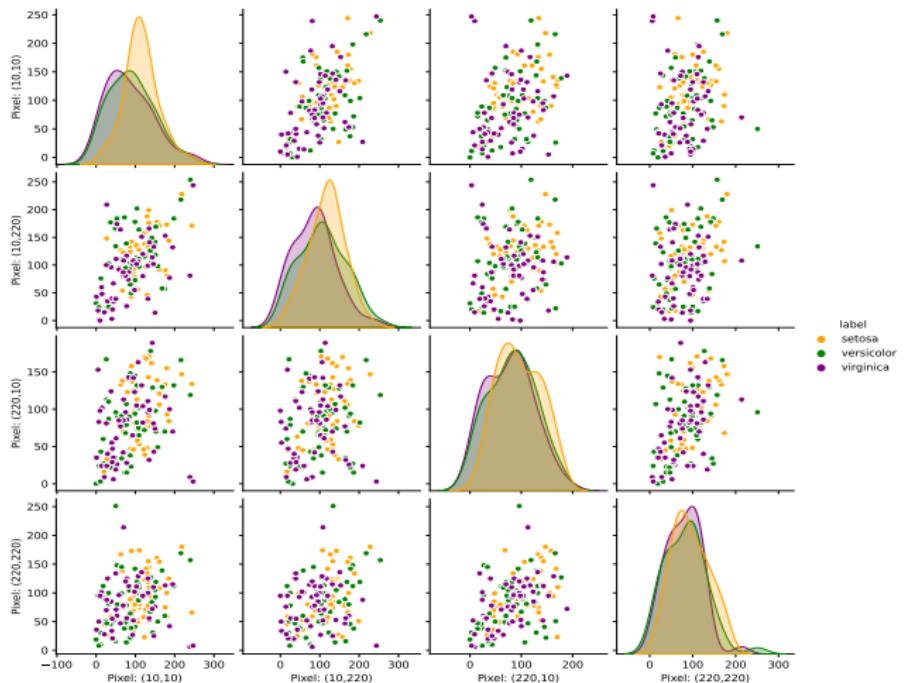


Figure: Exploratory data analysis - Approach 1

# Example - Iris Flower Classification

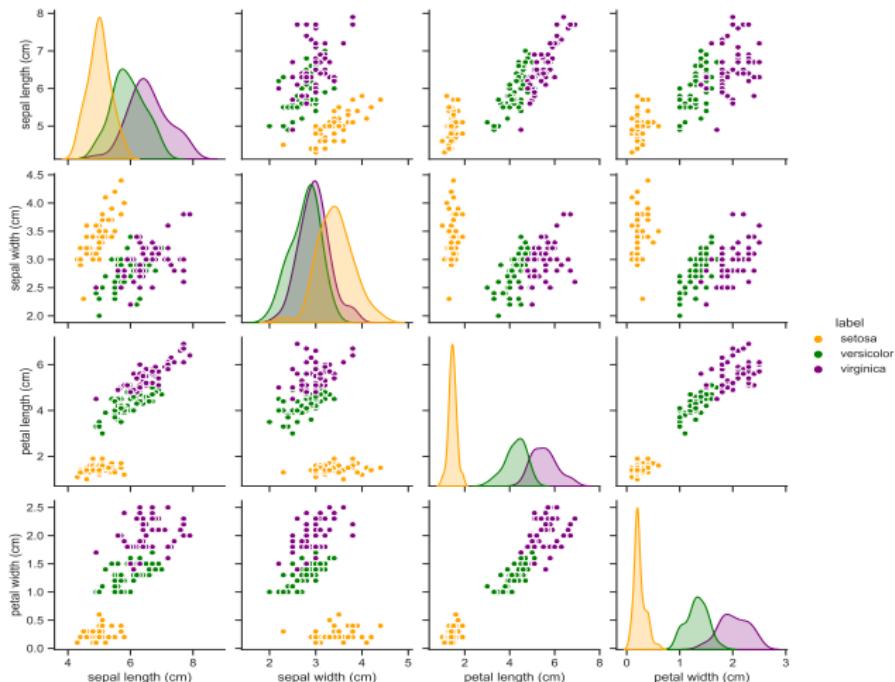
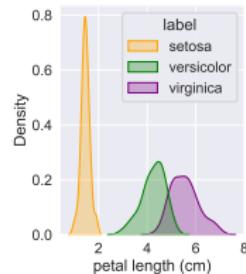
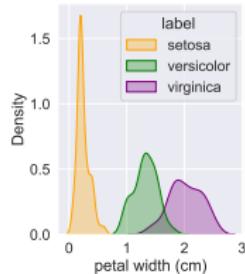


Figure: Exploratory data analysis - Approach 2

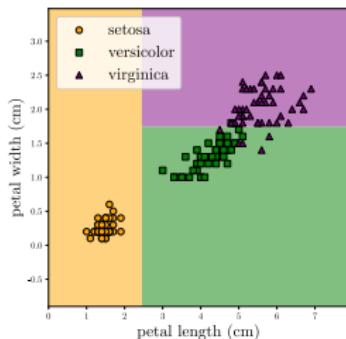
# More on Approach 2



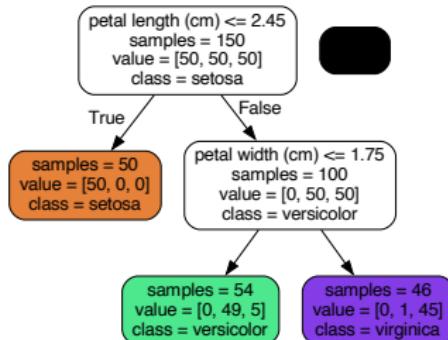
(a) Petal length



(b) Petal width



(c) Decision tree surface



(d) Decision tree

# Automate Classification

## Steps to Automate Classification

- Parameterizing mapping:  $f(\cdot) \Rightarrow f(\cdot; \theta)$
- Finding suitable  $\theta$  ( $\hat{\theta}$ ) using Experiment  $E$  (Model Fitting)
  - Enhancing performance measure  $P \Rightarrow$  decreasing misclassification error

## Note: Limitation with Misclassification Error

Inability to distinguish different errors

		$f(x; \theta)$		
		Setosa	Versicolor	Virginica (Poisonous)
$y$	Setosa	0 (0)	1 (1)	1 (1)
	Versicolor	1 (1)	0 (0)	1 (1)
	Virginica (Poisonous)	1 (10)	1 (10)	0 (0)

# Automate Classification

## Empirical Risk

Empirical Risk, the generalization of misclassification error is:

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^N l(y_n \neq f(\mathbf{x}_n; \boldsymbol{\theta}))$$

Using above definition, model fitting can be done via *Empirical Risk Minimization* (ERM) as:

$$\hat{\boldsymbol{\theta}} = \operatorname{argmin}_{\boldsymbol{\theta}} \frac{1}{N} \sum_{n=1}^N l(y_n \neq f(\mathbf{x}_n; \boldsymbol{\theta}))$$

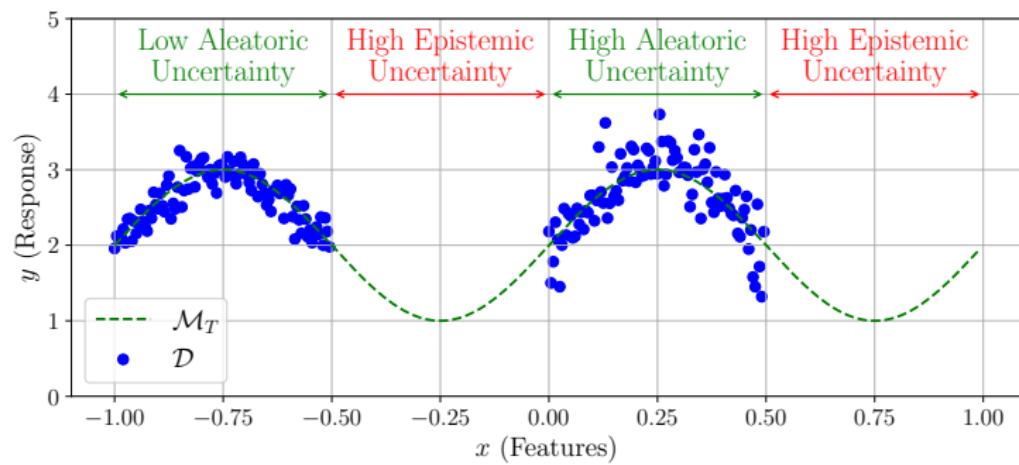
# Uncertainty [2]

## Epistemic Uncertainty (Model Uncertainty)

Uncertainty originated from lack of knowledge about true input-output mapping

## Aleatoric Uncertainty (Data Uncertainty)

Uncertainty originated from inherent randomness in experiment  $E$



# Uncertainty in Classification

## Capturing Uncertainty

To capture uncertainty, we can define *Conditional Probability Density* (CPD) as:

$$p(y = c|\mathbf{x}; \boldsymbol{\theta}) = f_c(\mathbf{x}; \boldsymbol{\theta}), \quad \begin{cases} 0 \leq f_c \leq 1 \\ \sum_{c=1}^C f_c = 1 \end{cases}$$

# Softmax Function

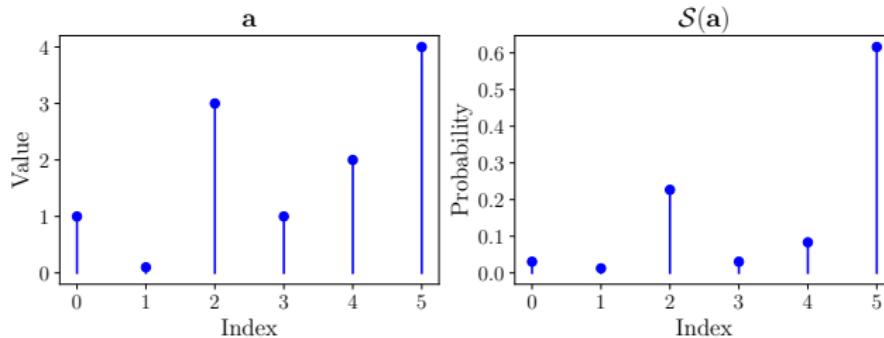
## Softmax Function

Consider the *logits* vector defined as:

$$\mathbf{a} \triangleq [a_1, \dots, a_C] = [f_1(\mathbf{x}; \boldsymbol{\theta}), \dots, f_C(\mathbf{x}; \boldsymbol{\theta})] = \mathbf{f}(\mathbf{x}; \boldsymbol{\theta})$$

The softmax function for this vector is defined as:

$$\mathcal{S} \triangleq \left[ \frac{e^{a_1}}{\sum_{c'=1}^C e^{a_{c'}}}, \dots, \frac{e^{a_C}}{\sum_{c'=1}^C e^{a_{c'}}} \right]$$



# Application of Softmax Function in Classification

## Capturing Uncertainty Using Softmax Function

Previously we define:

$$p(y = c|\mathbf{x}; \boldsymbol{\theta}) = f_c(\mathbf{x}; \boldsymbol{\theta}), \quad \begin{cases} 0 \leq f_c \leq 1 \\ \sum_{c=1}^C f_c = 1 \end{cases}$$

Now using softmax we have:

$$p(y = c|\mathbf{x}; \boldsymbol{\theta}) = \mathcal{S}_c(\mathbf{f}(\mathbf{x}; \boldsymbol{\theta}))$$

where the following constraints are met:

$$0 \leq \mathcal{S}_c(\mathbf{f}(\mathbf{x}; \boldsymbol{\theta})) \leq 1, \quad c = 1, 2, \dots, C$$

$$\sum_{c=1}^C \mathcal{S}_c(\mathbf{f}(\mathbf{x}; \boldsymbol{\theta})) = 1$$

# Fitting Probabilistic Model

## Maximum Likelihood Estimation

One approach to fit probabilistic models is *Maximum Likelihood Estimation* (MLE). We can equivalently define loss function as:

$$l(y, f(\mathbf{x}; \boldsymbol{\theta})) = -\log p(y|f(\mathbf{x}; \boldsymbol{\theta}))$$

Using above loss, the *Negative Log Likelihood* (NLL) over training set is:

$$\text{NLL}(\boldsymbol{\theta}) = -\frac{1}{N} \sum_{n=1}^N \log p(y_n|f(\mathbf{x}_n; \boldsymbol{\theta}))$$

Then the MLE for model parameters is:

$$\hat{\boldsymbol{\theta}}_{mle} = \operatorname{argmin}_{\boldsymbol{\theta}} \text{NLL}(\boldsymbol{\theta})$$

## Subsection 2

### Regression

# Supervised Learning - Regression

## Regression

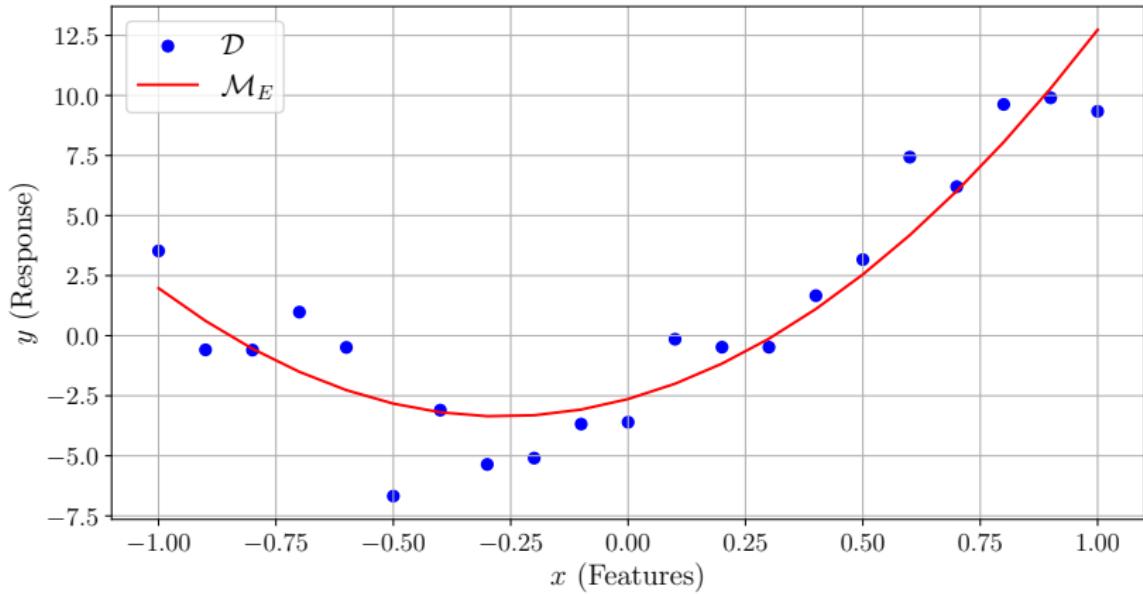
General Features:

- Task  $T$ : Finding mapping  $f : \mathbf{x} \mapsto \mathbf{y}$  ( $\mathbf{x} \in \mathcal{X} = \mathbb{R}^D$  and  $y \in \mathcal{Y}$ )
- Experience  $E$ : Set of  $N$  input-output pairs  $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$

Specific Features:

- $\mathcal{Y} = \mathbb{R}$
- $P = \frac{1}{N} \sum_{n=1}^N (y_n - f(\mathbf{x}_n; \boldsymbol{\theta}))^2$ 
  - $P$  is known as *Mean Square Error* (MSE)

# Example - One Dimensional Curve Fitting



# Model Fitting in Regression

## Model Fitting via ERM

Similar to classification, model parameters for regression problem can be found via ERM as:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \underbrace{\frac{1}{N} \sum_{n=1}^N (y_n - f(\mathbf{x}_n; \boldsymbol{\theta}))^2}_{\text{MSE}(\boldsymbol{\theta})}$$

# Uncertainty in Regression

## Capturing Uncertainty in Regression

To capture uncertainty, we assume the output distribution to be Gaussian (Normal) as:

$$\mathcal{N}(y|\mu, \sigma^2) \triangleq \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(y-\mu)^2}$$

We make the mean depend on the inputs by defining  $\mu \triangleq f(\mathbf{x}_n, \boldsymbol{\theta})$ . Then we have the following CPD:

$$p(y|\mathbf{x}; \boldsymbol{\theta}) = \mathcal{N}(y|f(\mathbf{x}; \boldsymbol{\theta}), \sigma^2)$$

### Subsection 3

#### Generalization

# Overfitting and Generalization

## Population Risk

Consider  $p^*(\mathbf{x}, y)$  to be the true generating distribution of training set. Then population risk is defined as:

$$\mathcal{L}(\boldsymbol{\theta}; p^*) \triangleq \mathbb{E}_{p^*(\mathbf{x}, y)}[l(y, f(\mathbf{x}; \boldsymbol{\theta}))]$$

## Generalization Gap

The difference  $\mathcal{L}(\boldsymbol{\theta}; p^*) - \mathcal{L}(\boldsymbol{\theta}; \mathcal{D}_{train})$  is called generalization gap where  $\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}_{train})$  is ERM defiend as:

$$\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}_{train}) = \frac{1}{|\mathcal{D}_{train}|} \sum_{(\mathbf{x}_n, y_n) \in \mathcal{D}_{train}} l(y_n, f(\mathbf{x}_n; \boldsymbol{\theta}))$$

## Overfitting

Overfitting occure when the generalization gap is large.

# Population Risk

## Population Risk Estimation

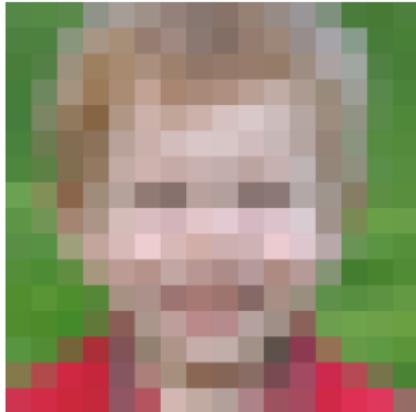
- In practice, we don't know  $p^*(\mathbf{x}, y)$ .
- We partition the data into two subsets, known as the training set and test set.
- We use test set to estimate population risk as:

$$\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}_{test}) = \frac{1}{|\mathcal{D}_{test}|} \sum_{(\mathbf{x}_n, y_n) \in \mathcal{D}_{test}} l(y_n, f(\mathbf{x}_n; \boldsymbol{\theta}))$$

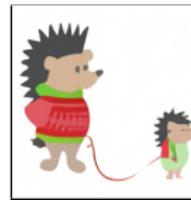
## Subsection 4

### Real World Applications

# Real World Classification: Super resolution [3]



# Real World Classification: Image Generation [4]



(a) a tapir made of accordion.  
a tapir with the texture of an  
accordion.

(b) an illustration of a baby  
hedgehog in a christmas  
sweater walking a dog

(c) a neon sign that reads  
“backprop”. a neon sign that  
reads “backprop”. backprop  
neon sign

# Real World Classification: Image Captioning [5]

A person riding a motorcycle on a dirt road.



A group of young people playing a game of frisbee.



A herd of elephants walking across a dry grass field.



Two dogs play in the grass.



Two hockey players are fighting over the puck.



A skateboarder does a trick on a ramp.



A dog is jumping to catch a frisbee.



A refrigerator filled with lots of food and drinks.



A little girl in a pink hat is blowing bubbles.



A red motorcycle parked on the side of the road.

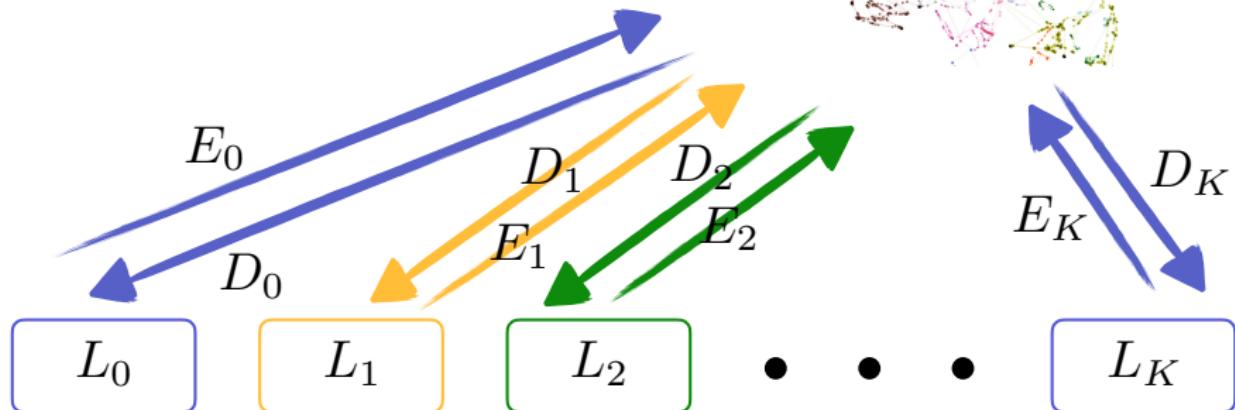


A yellow school bus parked in a parking lot.



# Real World Classification: Machine Translation [6]

Representation Space  $\mathcal{Z} =$



## Section 3

### Unsupervised Learning

# Unsupervised Learning

## Unsupervised Learning

Unsupervised Learning is:

- Task  $T$ : Dependent on the task
- Experience  $E$ : Set of  $N$  samples  $\mathcal{D} = \{\mathbf{x}_n\}_{n=1}^N$
- Performance measure  $P$ : Dependent on the task

## Subsection 1

### Clustering

## Clustering

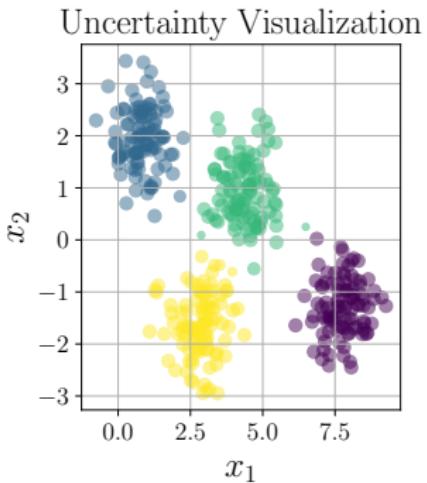
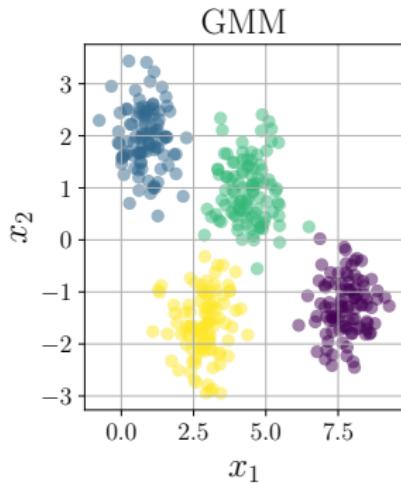
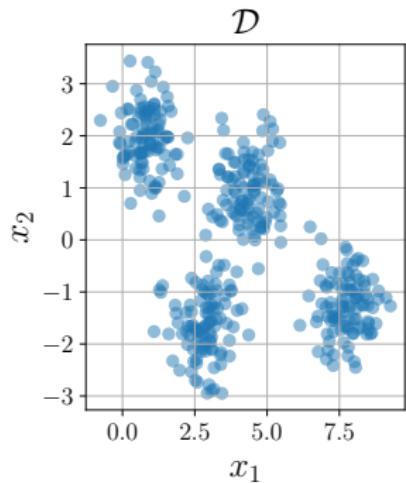
General Features:

- Experience  $E$ : Set of  $N$  samples  $\mathcal{D} = \{\mathbf{x}_n\}_{n=1}^N$

Specific Features:

- Task  $T$ : Partition the input into regions that contains *similar* points.
- Performance measure in *Compression*: Compression loss

# Clustering: Gaussian Mixture Model [7]



## Subsection 2

### Factors of Variation

# Unsupervised Learning - Factors of Variation

## Factors of Variation

General Features:

- Experience  $E$ : Set of  $N$  samples  $\mathcal{D} = \{\mathbf{x}_n\}_{n=1}^N$

Specific Features:

- Task  $T$ : Projecting data into low dimensional subspace which captures its main aspects
- Performance measure: Performance of low dimensional data in various downstream tasks

# Principle Component Analysis [8]



(a) Original  $64 \times 64$  pixels

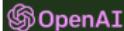


(b) Reconstructed from  $8 \times 8$  pixels

### Subsection 3

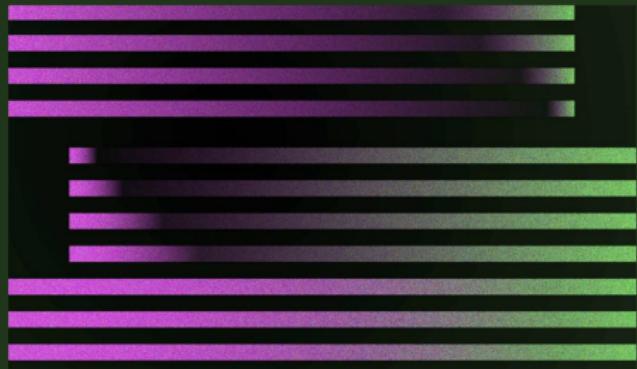
#### Real World Applications

# ChatGPT [9]

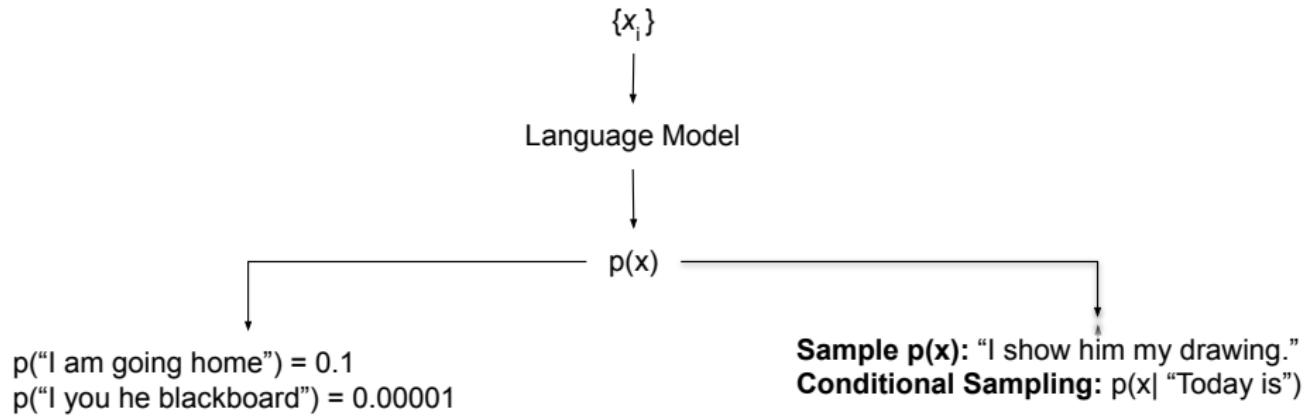
[API](#)[RESEARCH](#)[BLOG](#)[ABOUT](#)

## ChatGPT: Optimizing Language Models for Dialogue

We've trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests. ChatGPT is a sibling model to InstructGPT, which is trained to follow an instruction in a prompt and provide a detailed response.



# Language Model



## Section 4

### Reinforcement Learning

# Reinforcement Learning

## Reinforcement Learning

Reinforcement Learning is:

- Task  $T$ : Learning an agent to take action in different environmental conditions.
- Experience  $E$ : Set of  $N$  condition-action-reward triplet
- Performance measure  $P$ : Average reward

# Reinforcement Learning

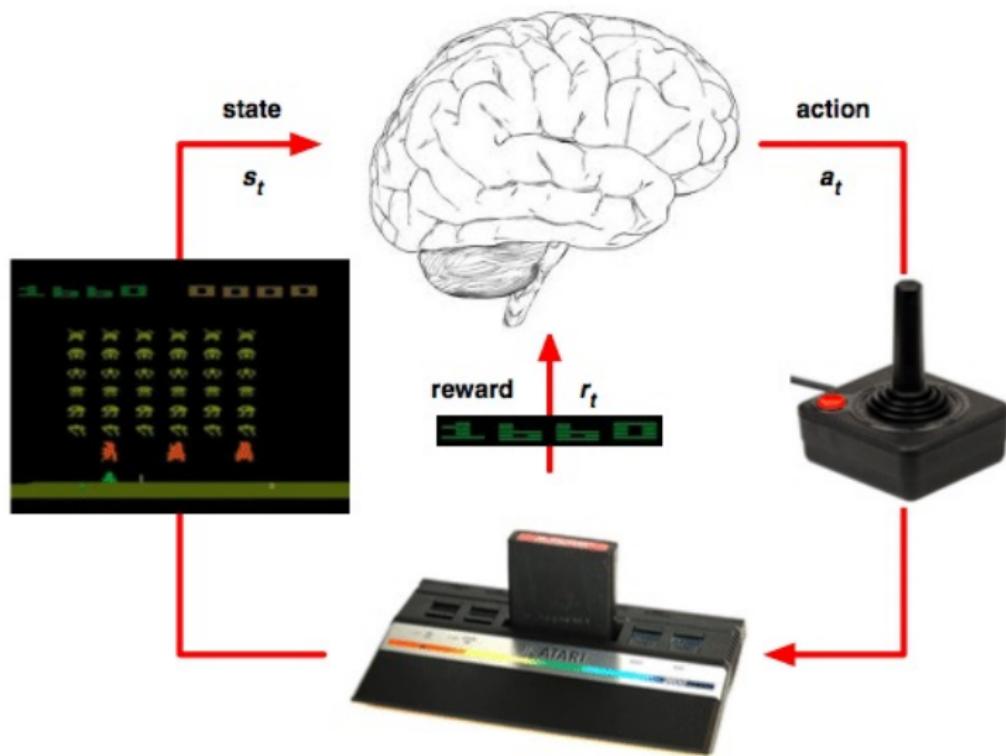
## Playing an Atari game

- Task T: learning policy mapping  $\mathbf{a} = \pi(\mathbf{x})$  where:
  - $\mathbf{a}$ : Action
  - $\mathbf{x}$ : Environmental conditions
- Experience  $E$ : Set of  $N$  triplet  $\{(\mathbf{x}_n, \mathbf{a}_n, r_n)\}_{n=1}^N$
- Performance measure  $P$ : Maximizing the reward

## Subsection 1

Real World Applications

# Reinforcement Learning



# References I

-  Tom M Mitchell and Tom M Mitchell,  
*Machine learningg*, vol. 1,  
McGraw-hill New York, 1997.
-  “Ensemble distribution distillation,”  
<https://campusai.github.io/papers/ensemble-distribution-distillation>,  
Accessed: 2022-09-13.
-  Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and  
Mohammad Norouzi,  
“Image super-resolution via iterative refinement,”  
*IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.
-  Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark  
Chen, and Ilya Sutskever,  
“Zero-shot text-to-image generation,”  
in *International Conference on Machine Learning*. PMLR, 2021, pp. 8821–8831.
-  Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan,  
“Show and tell: A neural image caption generator,”  
in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015,  
pp. 3156–3164.
-  Han Zhao, Junjie Hu, and Andrej Risteski,  
“On learning language-invariant representations for universal machine translation,”  
in *International Conference on Machine Learning*. PMLR, 2020, pp. 11352–11364.

# References II



"In depth: Gaussian mixture models,"

<https://jakevdp.github.io/PythonDataScienceHandbook/05.12-gaussian-mixtures.html>,  
Accessed: 2022-09-13.



"Facial image compression and reconstruction with pca,"

<https://shankarmsy.github.io/posts/pca-sklearn.html#>,  
Accessed: 2022-09-13.



"Chatgpt: Optimizing language models for dialogue," <https://openai.com/blog/chatgpt/>,  
Accessed: 2022-09-13.