APL 405: Machine Learning in Mechanics

Lecture 2: Supervised learning

by

Rajdip Nayek

Assistant Professor,
Applied Mechanics Department,
IIT Delhi

Instructor email: rajdipn@am.iitd.ac.in

Different types of machine learning

Supervised

Teacher provides answer

- Labelled data
- Direct feedback
- Predict outcome

- Classification
- Regression

Unsupervised

No teacher, find patterns!

- No labels
- No feedback
- Find hidden structure

- Clustering
- Dimensionality reduction
- Outlier detection

Semi-supervised

- Some labelled data
- A lot of unlabelled data

Reinforcement

Teacher provides rewards

- Decision process
- Rewards
- Learn series of actions

- Gaming
- Control

Example of Supervised learning

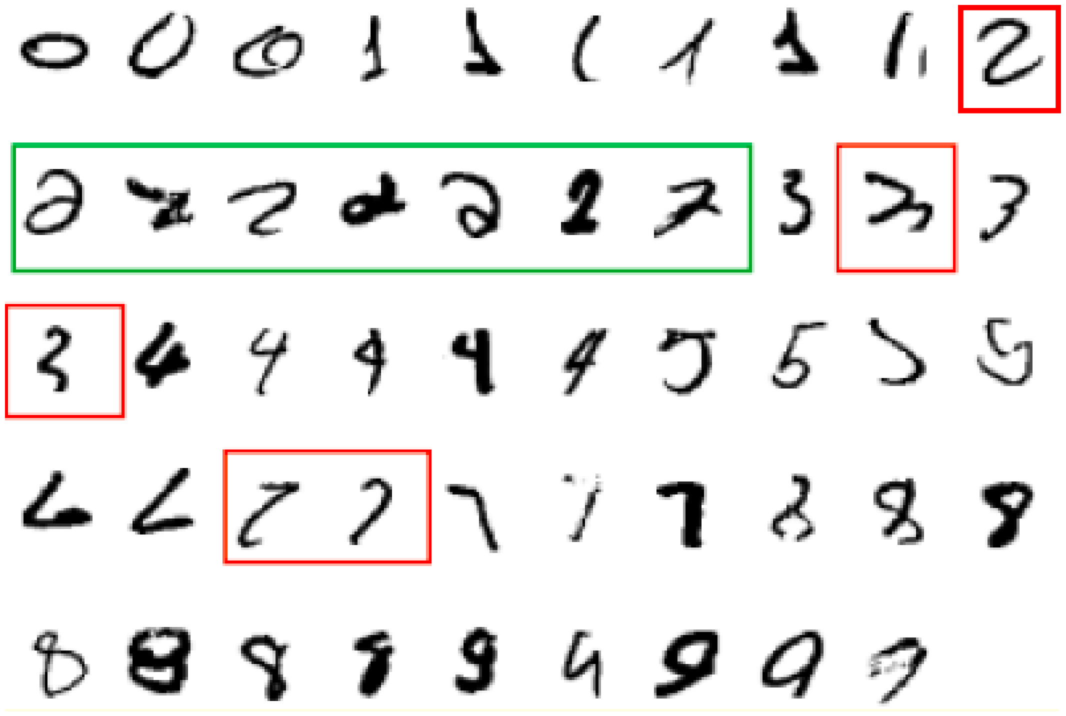
Supervised learning: have labelled examples of what is “correct”

e.g. Handwritten digit classification with the MNIST dataset

- **Task:** Given an image of a handwritten digit, predict the digit class
  - **Input:** a handwritten image of a digit
  - **Output (or target):** the digit class
- **Data:** 70,000 images of handwritten digits labelled by humans
  - **Training set:** first 60,000 images used to train the network
  - **Test set:** last 10,000 images, not used during training, used to assess performance

Example of Supervised learning

Test prediction: What type of images look like a "**2**"?

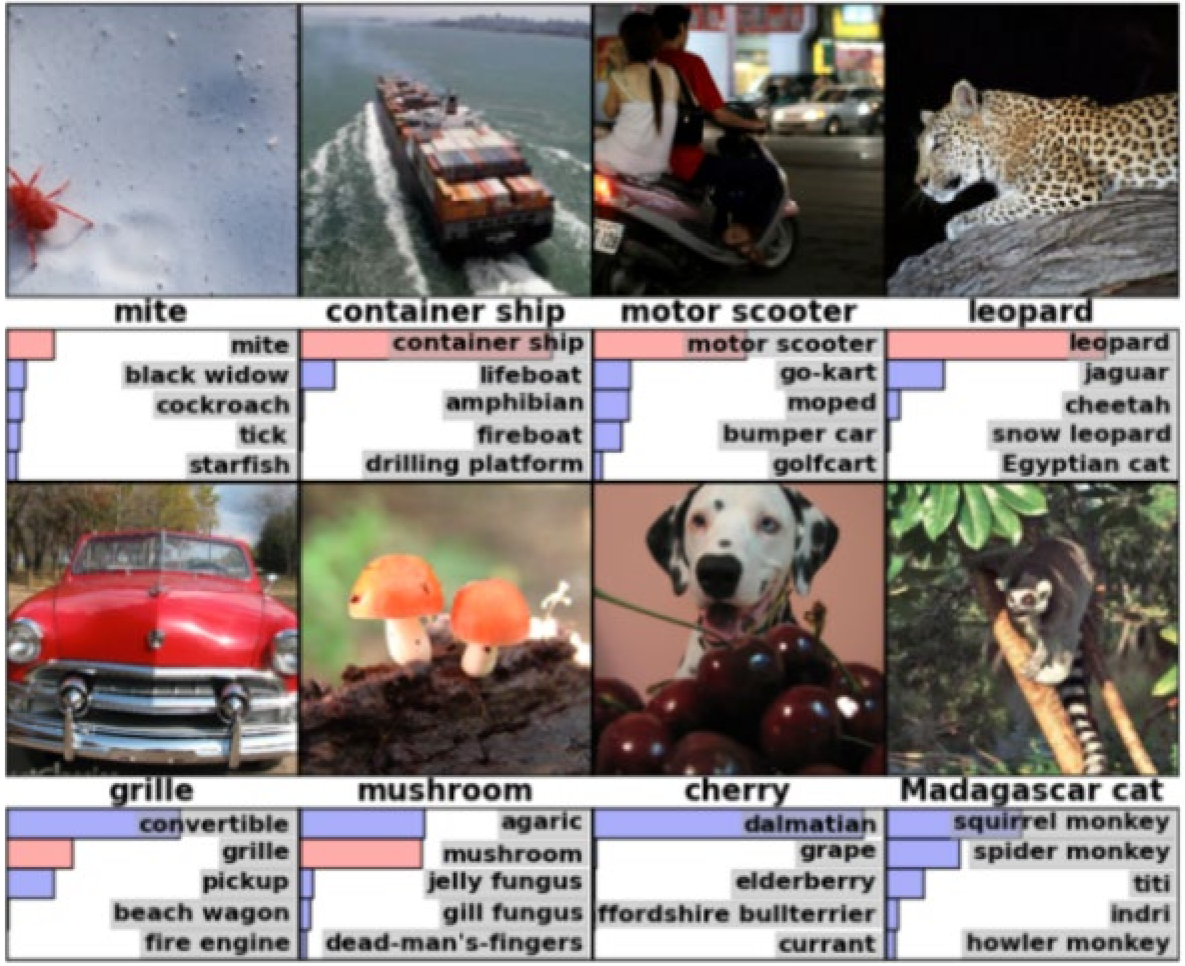


Misclassification

Example of Supervised learning

Object Recognition: Detect the class of the object

- ImageNet
- 1.2 million labelled images
- 1000 classes
- Lots of variability in lighting, viewpoint, etc.
- Deep neural networks reduced error rates from 26% to under 4%



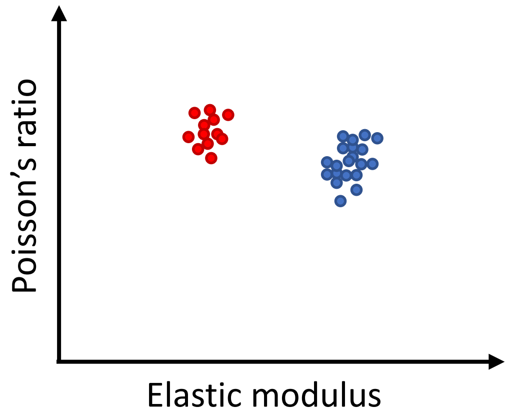
Example of Unsupervised learning

Unsupervised learning: no labelled examples, you only have input data. You are looking for interesting patterns in the data

- To find clusters in data
- To find a compressed representation
- To find a generative model that could be used to generate more data

E.g. Clustering – Group the input data into separate classes

Elastic mod.	Poisson's ratio
210 GPa	0.279
70 GPa	0.325
⋮	⋮
190 GPa	0.267

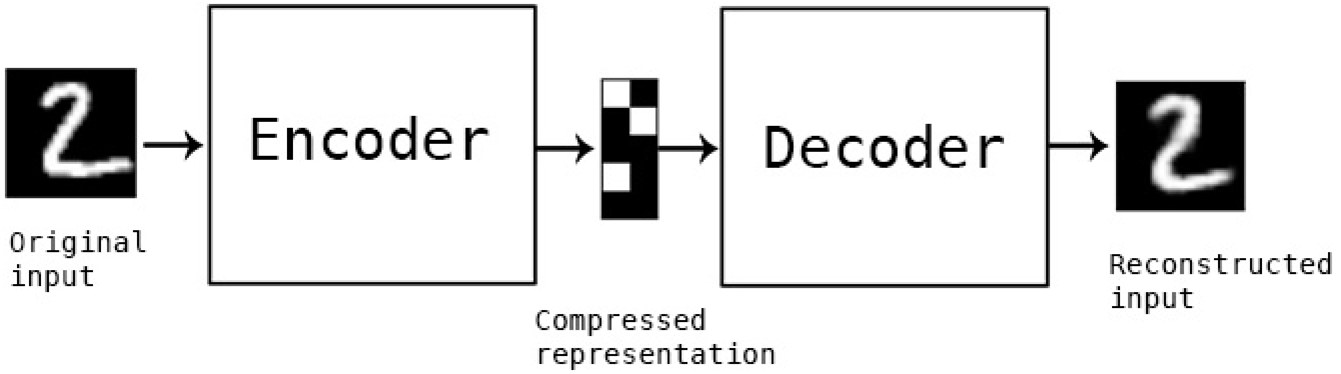


Example of Unsupervised learning

Unsupervised learning: no labelled examples, you only have input data. You are looking for interesting patterns in the data

- To find clusters in data
- To find a compressed representation
- To find a generative model that could be used to generate more data

E.g. Compressed representation – Find a reduced dimension of the input



Example of Unsupervised learning

Unsupervised learning: no labelled examples, you only have input data. You are looking for interesting patterns in the data

- In **generative modeling**, we want to learn a distribution over some dataset, such as natural images. We can then sample from the generative model and see if it looks like the data.

Generated faces
(not true faces)

Example of reinforcement learning

Agent

State

Reward

Action

Environment

Computer playing a game

Goal: Learn to choose actions that maximize rewards

- An **agent** (e.g. player) interacts with an **environment** (e.g. enemy-killing game)
- In each time step
  - the agent receives observations of the **state** (e.g. how many enemies remaining)
  - the agent picks an **action** (e.g. moving to safe location, or killing an enemy)
- The agent will periodically receive some **rewards** (e.g. health, ammunition, scores)

Idea of RL is based on animal psychology

- Reinforcements are used to train animals

- Negative reinforcements
  - Hunger
  - Pain

- Positive reinforcements
  - Food
  - Pleasure

- This psychology is applied here to computers
  - Rewards: numbers or numerical signals indicating how good the agent performed

  - Example of rewards: Win/loss in games, points earned, etc.

Supervised Learning

Supervised Learning: Background

- We start with **Supervised Learning**; it is most common type of machine learning (will span most of this course)

- The task is to **learn the function** f that best maps certain **input** $(\mathbf{x})$ to **output** (y)

		A	B	C
	1	x1	x2	y
Example 1	2	2.1	1.5	3
Example 2	3	2.3	-0.6	1
Example 3	4	3.1	0.9	3.2
Example 4	5	2.4	-0.1	1.2

	A	B	C
1	x1	x2	y
2	2.1	1.5	Cat
3	2.3	-0.6	Dog
4	3.1	0.9	Cat
5	2.4	-0.1	Dog

$$y = f(\mathbf{x})$$

Supervised Learning: What do we do in supervised ML?

- The task in supervised ML is to **learn the function** f that best maps certain **input** ($\mathbf{x}$) to **output** (y)

$$y = f(\mathbf{x})$$

- We don't know what the function (f) looks like or its form
  - If we knew the form, we would use it directly and we would not need to learn it from data

- Moreover, the output y is often observed with some errors e that is independent of the input $\mathbf{x}$
  - The error could be due to measurement instrument errors
  - The error could be due to not including enough input features to sufficient characterize the mapping from $\mathbf{x}$ to y

$$y = f(\mathbf{x}) + e$$

- In supervised ML, we use some *labeled training data* (input-output pairs) that contains examples of how some *input* $\mathbf{x}$ relates to *output* y to learn the input-output mapping
  - Say, N examples of labelled training data: $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)$
  - Labelled data: Each input $\mathbf{x}_i$ is accompanied by an associated label y_i, which be jointly recorded or labelled later by some domain expert

Supervised Learning: What is reason for learning a function?

$$y = f(\mathbf{x}) + e$$

- The most common type of ML is to learn the mapping $y = f(\mathbf{x})$ to **make good predictions** of the output for new examples of input (say,) → to **generalize** well beyond training data
  - This is called **predictive modeling** or **predictive analytics** and our goal is to make the most accurate predictions possible
  - Hence, we are not really interested in the form of the function (f) that we are learning, only that it makes accurate predictions

- We could learn the mapping of $y = f(\mathbf{x})$ to **understand more about the relationship** in the data → **statistical inference**
  - If this were the goal, we would use simpler methods and give more importance to understanding the learned form of f than making accurate predictions
  - E.g. "Does eating seafood increase life expectancy?" requires careful reasoning about the function that was learned

Supervised Learning: Learning a function

$$y = f(\mathbf{x}) + e$$

- Learning a function f means estimating its form and its parameters from the noisy data that is available with us
  - The estimate will have errors and will not be exactly same as the underlying true mapping from $\mathbf{x} \to y$
  - Much time in applied machine learning is spent attempting to improve the estimate of the underlying function and in turn improve the performance of the predictions made by the model

- Supervised ML algorithms are techniques for estimating the target function f to predict the output y given input $\mathbf{x}$

- Different ML algorithms make different assumptions about the form of the function being learned
  - Linear vs nonlinear models
  - Parametric vs non-parametric models

Parametric vs Non-parametric algorithms

- Assumptions about the unknown f can greatly simplify the learning process, but can also limit what can be learned

Parametric models

1. Simplify the unknown function f to a known explicit form
2. Summarizes the model using a fixed number of model parameters (independent of the number of training examples)

Pros

1. Often **simpler** and **faster**
2. May require **less training data**

Cons

1. **Constrained**: Functional form is fixed
2. **Poor fit:** Unlikely to match the underlying true function

Non-Parametric models

1. Don't make strong assumptions about the form of f
2. Summarizes the model using number of model parameters that depend upon the number of training examples

Pros

1. **Flexible:** Can fit any type of function
2. **Powerful**: Can result in better prediction

Cons

1. **More data**: Require a lot more training data
2. **Over fitting:** Harder to explain certain predictions made

Supervised learning: Two types of data

The variables contained in the data (input $\mathbf{x}$ as well as output y) can be of two different types:

- **Numerical (quantitative)**
  - Has a **natural ordering**, i.e., a numerical variable maybe larger or smaller than another one
  - Can be **continuous** or **discrete**

- **Categorical (qualitative)**
  - Lacks a natural ordering
  - Is **always discrete**

- The notion of categorical vs. numerical applies to both the output variable y and to the p elements x_j of the input vector variable $\mathbf{x} = [x_1 \quad x_2 \quad \cdots \quad x_p]^T$

- Also, the p components of the input vector do not have to be of the same type and can be a mix of categorical and numerical input

- However, the output y is either categorical or numerical

Numerical vs Categorical data: Examples

Data Type	Example	Handled as
Number (continuous)	15.58 km/h, 11.50 km/h	**Numerical**
Number (discrete) with natural ordering	0 bikes, 1 bikes, 2 bikes	**Numerical**
Number (discrete) *without* natural ordering	1 = Argentina, 2 = Brazil, 3 = India	**Categorical**
Text String	Hello, Bye, Welcome	**Categorical**
?	$3.4 + 5.6i, -6.2 + 0.1i$	**?**

- The distinction between numerical and categorical is sometimes arbitrary
- For example, having no bike is qualitatively different from having bikes, and we can use the categorical variable 'bikes: yes/no' instead of the numerical '0, 1 or 2 bikes'
- Therefore, the **decision lies the ML engineer** whether a certain variable is to be considered as numerical or categorical

Supervised ML: Regression vs Classification

- Output variable y? → **categorical** → **Classification**
- Output variable y?→ **numerical** → **Regression**
- Note that the $p-$ dimensional input vector variable $\mathbf{x} = [x_1 \quad x_2 \quad \cdots \quad x_p]^T$ can be either numerical or categorical for both regression and classification problems
- It is **only the type of the output** that determines whether a problem is a regression or a classification problem
- Classification: **Binary** vs **Multi-class**
  - Output is categorical
  - Output can take values in a finite set
  - Binary classification, if only two set of values. E.g. True or False
  - Multi-class classification: if more than two set of values. E.g. “Sweden”, “Norway”, “Finland”, “Denmark”

Examples of classification and regression

Problem	Input	Output	Classification or Regression?
Spam detection	Text (set of words)		
Stock price prediction	Time-series of prices		
Speech recognition	Audio signal		

Examples of classification and regression

Problem	Input	Output	Classification or Regression?
Digit recognition	Images of digits		
Housing valuation	House features		
Weather prediction	Sensor data (images, wind speed)		

Bias-Variance Trade-Off

- The goal of any supervised machine learning algorithm is to best estimate the mapping function f for the output variable y given the input data $\mathbf{x} = [x_1 \quad x_2 \quad \cdots \quad x_p]^T$

- The prediction error for any machine learning algorithm results from three things:
  - Bias
  - Variance
  - Irreducible error, that cannot be reduced regardless of the algorithm used; caused by factors like partially known inputs

- **Bias** - the simplifying assumptions made by a model to make the target function easier to learn
  - They make algorithms easier to understand but are generally less flexible
    - Low bias: Suggests less assumptions about the function f
    - High bias: Suggests more assumptions about the function f

- **Variance** - amount by which the <u>estimated function</u> (say $\hat{f}$) will change if a different training data was used to obtain the estimate
  - Machine learning algorithms that have a high variance are strongly influenced by the specifics of the training data
    - Low variance: Suggests small changes to the estimated function with changes to the training dataset
    - High variance: Suggests large changes to the estimated function with changes to the training dataset

The goal of any supervised machine learning algorithm **is to achieve low bias and low variance**

Underfit vs Overfit

- The goal of any supervised machine learning algorithm is to best estimate the mapping function f for the output variable y given the input data $\mathbf{x} = [x_1 \quad x_2 \quad \cdots \quad x_p]^T$

- The prediction error for any machine learning algorithm results from three things:
  - Bias
  - Variance
  - Irreducible Error, that cannot be reduced regardless of the algorithm used; caused by factors like partially known inputs

- **Bias** - the simplifying assumptions made by a model to make the target function easier to learn
  - They make algorithms easier to understand but are generally less flexible
    - Low bias: Suggests less assumptions about the function f
    - High bias: Suggests more assumptions about the function f

- **Variance** - amount by which the estimated function (say $\hat{f}$) will change if a different training data was used to obtain the estimate
  - Machine learning algorithms that have a high variance are strongly influenced by the specifics of the training data
    - Low variance: Suggests small changes to the estimated function with changes to the training dataset
    - High variance: Suggests large changes to the estimated function with changes to the training dataset

The goal of any supervised machine learning algorithm **is to achieve low bias and low variance**