Introduction to Machine Learning

NYU K12 STEM Education: Machine Learning

Department of Electrical and Computer Engineering,
NYU Tandon School of Engineering
Brooklyn, New York

Course Details

- Course Website
- Instructors:

Rugved Mhatre
rugved.mhatre@nyu.edu

Akshath Mahajan
akshathmahajan@nyu.edu

NYU

Outline

Supervised Learning

- This is exactly what we've been doing so far

Supervised Learning

- This is exactly what we've been doing so far
- In a supervised setting, we have inputs and corresponding their outputs

Supervised Learning

- This is exactly what we've been doing so far
- In a supervised setting, we have inputs and corresponding their outputs
- Let's look at a few advanced supervised problems later!

Outline

Unsupervised Learning

- What if we don't have labelled data for the given task?

Unsupervised Learning

- What if we don't have labelled data for the given task?
- The dataset still holds structure, we just don't have access to it

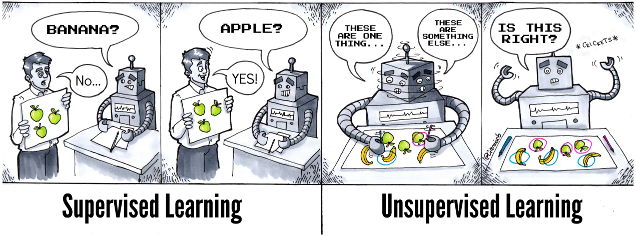


Figure 1: Supervised vs Unsupervised Learning

Unsupervised Learning

- What if we don't have labelled data for the given task?
- The dataset still holds structure, we just don't have access to it
- Or what if there is a need to create data?
- Example - Clustering, Generative AI, etc.

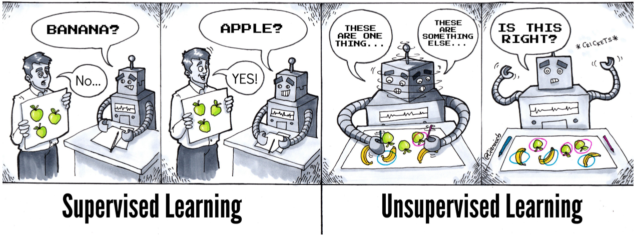


Figure 1: Supervised vs Unsupervised Learning

Unsupervised Learning

- What if we don't have labelled data for the given task?
- The dataset still holds structure, we just don't have access to it
- Or what if there is a need to create data?
- Example - Clustering, Generative AI, etc.
- Let's look at some unsupervised models

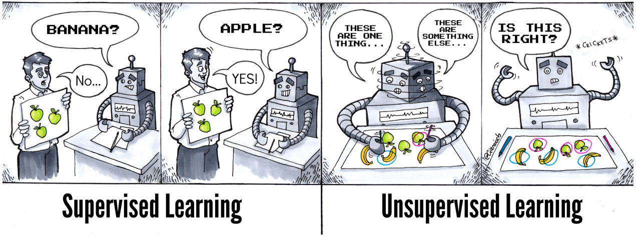


Figure 1: Supervised vs Unsupervised Learning

Clustering

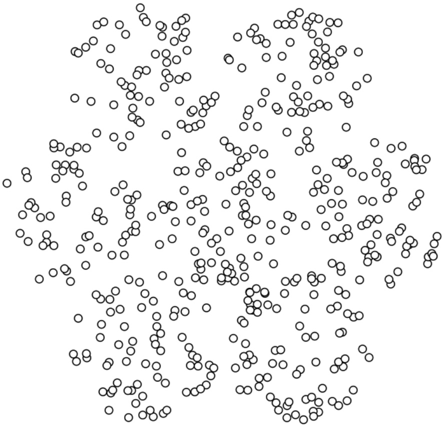


Figure 2: Problem Statement

KMeans Clustering

- Works by selecting ’k’ arbitrary centroids for clusters

KMeans Clustering

- Works by selecting ’k’ arbitrary centroids for clusters
- Euclidian Distance is used to assign points to a cluster
- (We can use other measures as well)

KMeans Clustering

- Works by selecting 'k' arbitrary centroids for clusters
- Euclidian Distance is used to assign points to a cluster
- (We can use other measures as well)
- Centroids are updated and points are reassigned till convergence

KMeans Clustering

- Works by selecting 'k' arbitrary centroids for clusters
- Euclidian Distance is used to assign points to a cluster
- (We can use other measures as well)
- Centroids are updated and points are reassigned till convergence
- Let's see a step-by-step Visualization!

KMeans - Drawbacks

- What drawbacks can it have?

KMeans - Drawbacks

- What drawbacks can it have?
  - k matters a lot!

KMeans - Drawbacks

- What drawbacks can it have?
  - k matters a lot!
  - The algorithm depends heavily on the initial centroids

KMeans - Drawbacks

- What drawbacks can it have?
  - k matters a lot!
  - The algorithm depends heavily on the initial centroids
  - categorical data doesn't have a natural notion of distance or similarity

KMeans - Evaluation

- How to evaluate the model? We don't have any labels?

KMeans - Evaluation

- How to evaluate the model? We don't have any labels?
- Inertia (J) measures the sum of squared distances between data points (x_i) and their assigned cluster centroids (μ_k).

KMeans - Evaluation

- How to evaluate the model? We don't have any labels?
- Inertia (J) measures the sum of squared distances between data points (x_i) and their assigned cluster centroids (μ_k).
- Goal: Have low inertia!

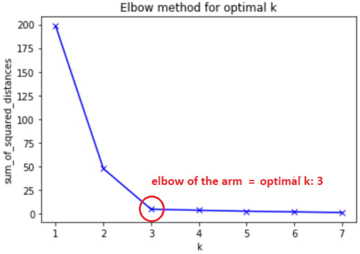


Figure 3: Elbow Method

KMeans - Evaluation

- How to evaluate the model? We don't have any labels?
- Inertia (J) measures the sum of squared distances between data points (x_i) and their assigned cluster centroids (μ_k).
- Goal: Have low inertia!

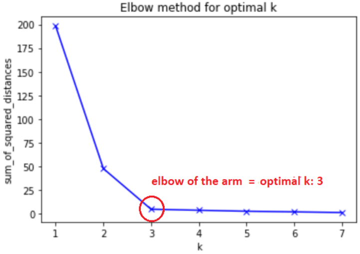


Figure 3: Elbow Method

- Let's try a notebook!

Outline

Object Detection

- Divides the image into $n \times n$ grid-cells, which detects objects whose center falls within the cell

Object Detection

- Divides the image into $n \times n$ grid-cells, which detects objects whose center falls within the cell
- For each grid cell, Yolo:

Object Detection

- Divides the image into $n \times n$ grid-cells, which detects objects whose center falls within the cell
- For each grid cell, Yolo:
  - Predicts B bounding boxes and its box confidence score
  - Each box will have its class probability
  - Output contains box coordinates, confidence scores, and class probabilities for each grid cell.

Object Detection

- Divides the image into $n \times n$ grid-cells, which detects objects whose center falls within the cell
- For each grid cell, Yolo:
  - Predicts B bounding boxes and its box confidence score
  - Each box will have its class probability
  - Output contains box coordinates, confidence scores, and class probabilities for each grid cell.
- Finally duplicate detections of the same object are suppressed using non-max suppression

Object Detection

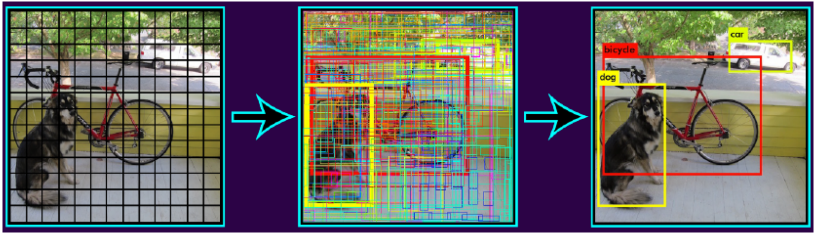


Figure 4: Yolo Object Detection
(Source)

Semantic Segmentation

- Every Pixel is associated with a class

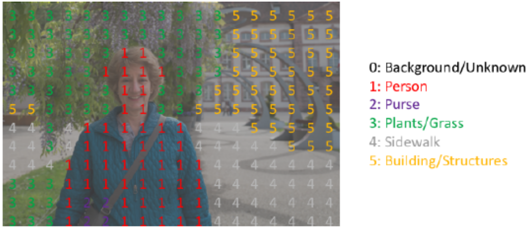


Figure 5: Semantic Segmentation
(Source)

Semantic Segmentation

- Every Pixel is associated with a class
- UNets having Encoder-decoder structure are really powerful

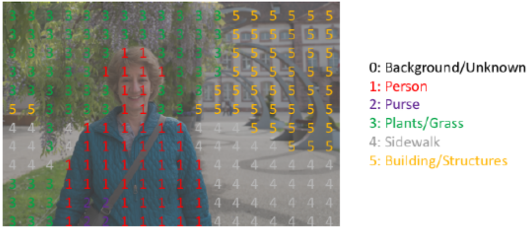


Figure 5: Semantic Segmentation
(Source)

Semantic Segmentation

- Every Pixel is associated with a class
- UNets having Encoder-decoder structure are really powerful
- Decode using transposed convolution/deconvolution

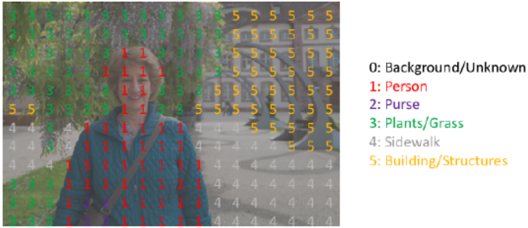
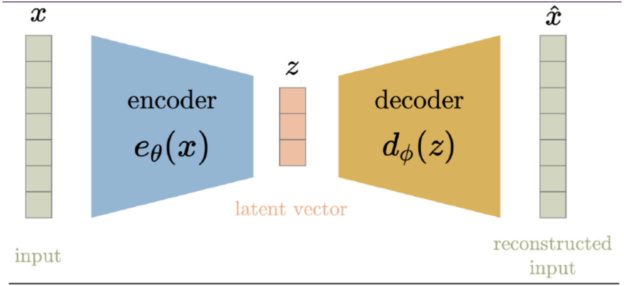


Figure 5: Semantic Segmentation
(Source)

AutoEncoders

- Encoder-Decoder structure

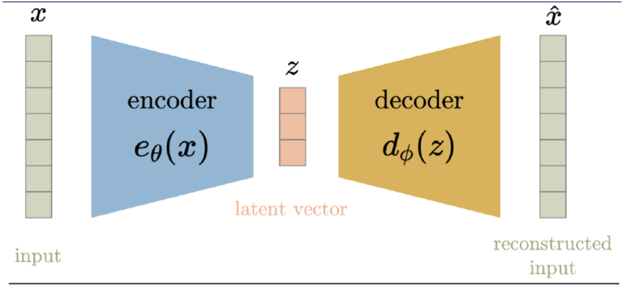


$$loss = \|x - \hat{x}\|_2 = \|x - d_\phi(z)\|_2 = \|x - d_\phi(e_\theta(x))\|_2$$

Figure 6: AutoEncoder structure
(Source)

AutoEncoders

- Encoder-Decoder structure
- Encoder helps in creating latent representations

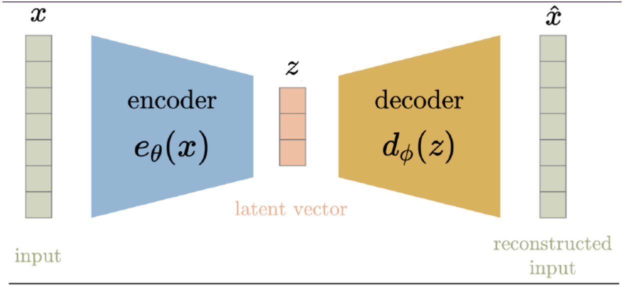


$$loss = \|x - \hat{x}\|_2 = \|x - d_\phi(z)\|_2 = \|x - d_\phi(e_\theta(x))\|_2$$

Figure 6: AutoEncoder structure
(Source)

AutoEncoders

- Encoder-Decoder structure
- Encoder helps in creating latent representations
- Decoder helps in generating outputs from the latent representation



$$loss = \|x - \hat{x}\|_2 = \|x - d_\phi(z)\|_2 = \|x - d_\phi(e_\theta(x))\|_2$$

Figure 6: AutoEncoder structure
(Source)

Denoising AutoEncoders

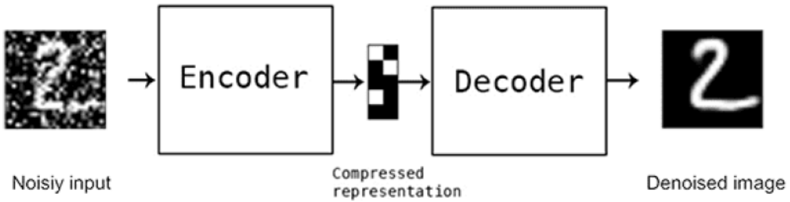
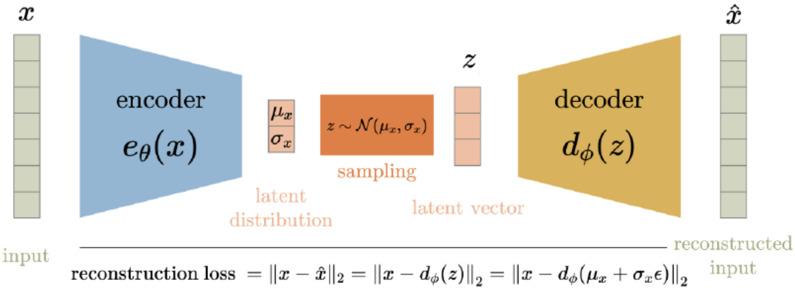


Figure 7: Denoising AutoEncoders

Variational AutoEncoders



$$\text{reconstruction loss} = \|x - \hat{x}\|_2 = \|x - d_\phi(z)\|_2 = \|x - d_\phi(\mu_x + \sigma_x \epsilon)\|_2$$

$$\mu_x, \sigma_x = e_\theta(x), \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

$$\textit{similarity loss} = \textit{KL Divergence} = D_{KL}(\mathcal{N}(\mu_x, \sigma_x) \parallel \mathcal{N}(\mathbf{0}, \mathbf{I}))$$

$$\textit{loss} = \textit{reconstruction loss} + \textit{similarity loss}$$

Figure 8: Variational AutoEncoders
(Source)

GANs - Generative Adverserial Networks

- Goal: Generate samples never seen before

GANs - Generative Adverserial Networks

- Goal: Generate samples never seen before
- There are 2 networks:

GANs - Generative Adverserial Networks

- Goal: Generate samples never seen before
- There are 2 networks:
  - Generator: Generate fake samples from noise that appear similar to real samples

GANs - Generative Adverserial Networks

- Goal: Generate samples never seen before
- There are 2 networks:
  - Generator: Generate fake samples from noise that appear similar to real samples
  - Discriminator: Tell apart real and fake samples

GANs - Generative Adverserial Networks

- Goal: Generate samples never seen before
- There are 2 networks:
  - Generator: Generate fake samples from noise that appear similar to real samples
  - Discriminator: Tell apart real and fake samples
- Generator aims to fool the discriminator
- Both learn from each other

GANs - Generative Adverserial Networks

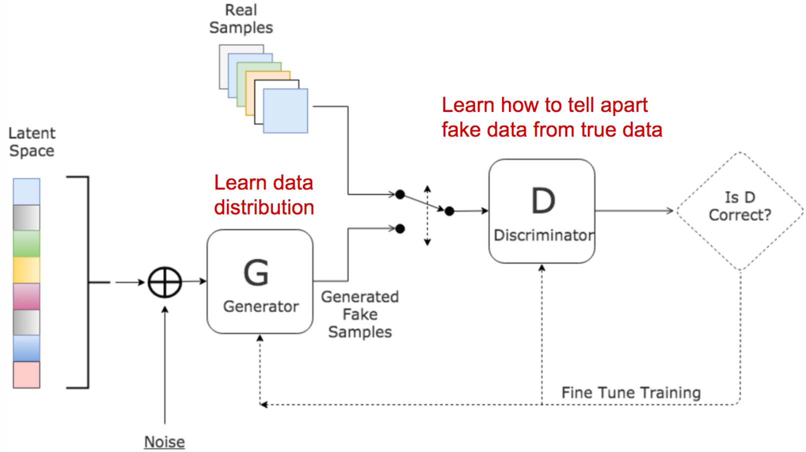


Figure 9: GANs - Architecture

GANs - Generative Adverserial Networks

Figure 10: Progress of GANs

GANs - Generative Adverserial Networks

Figure 11: Cats that don't exist

Applications of GANs - Image Coloring

Figure 12: Image colorization
(Source)

Applications of GANs - Image Synthesis

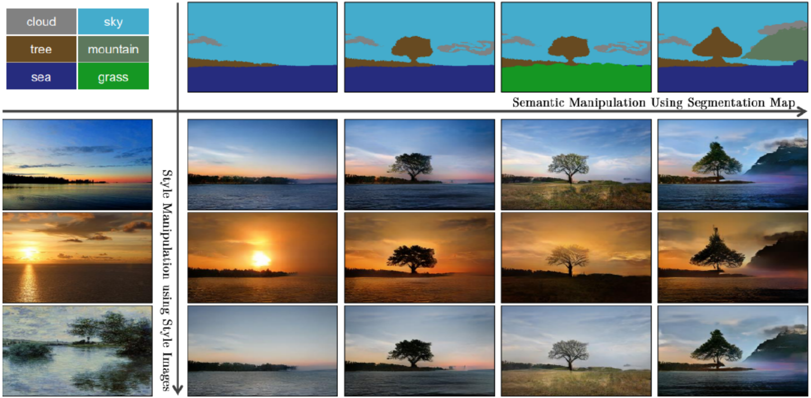


Figure 13: Image Synthesis
(Source)

Applications of GANs - Image Super Resolution

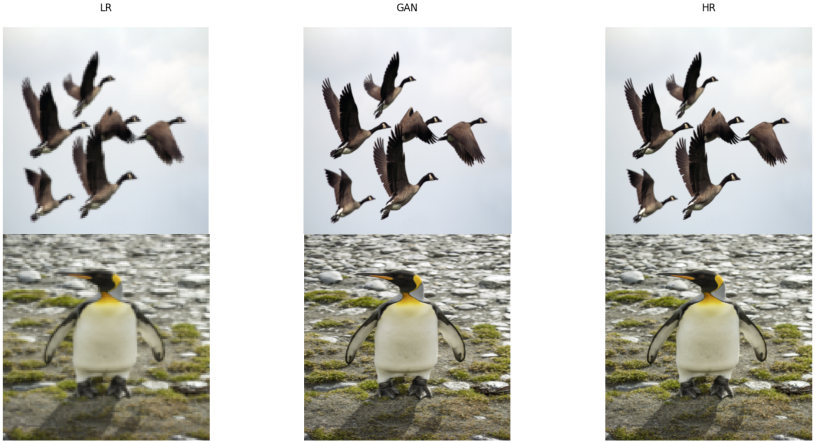


Figure 14: Image Super Resolution

Outline

1. Supervised Learning
2. Unsupervised Learning
3. Advanced Problems
4. Social Impact of ML
5. Course Takeaways

Bias in Models

- Neural Networks before training have randomly initialized weights
- It trains on a given dataset

Bias in Models

- Neural Networks before training have randomly initialized weights
- It trains on a given dataset
- Do the models harbor any discriminatory properties (racism, sexism, homophobia, or transphobia)?

Bias in Models

- Neural Networks before training have randomly initialized weights
- It trains on a given dataset
- Do the models harbor any discriminatory properties (racism, sexism, homophobia, or transphobia)?
- No! It isn't sentient. But it may have biased outputs

Example of Bias

- PULSE is a face depixelizing algorithm -

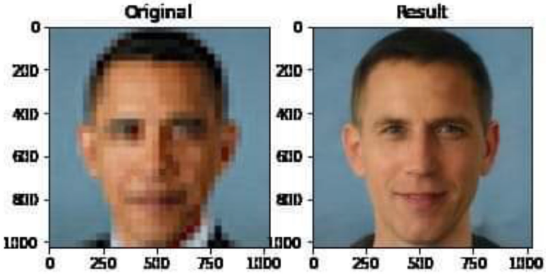


Figure 15: Bias shown in Models

- So where does this Bias come from?

Bias in Models

- Answer 1 - Underrepresented classes in a dataset
- Example: Less number of positive cases in the dataset

Bias in Models

- Answer 1 - Underrepresented classes in a dataset
- Example: Less number of positive cases in the dataset
- Bias not inherent in data
- CelebA dataset has images of "traditionally attractive", predomintally white and cis people with heavy makup, which are potentially photoshopped.
- In the real world, this is not the case

Real-world biases leak into Machine Learning

- Bias comes from Biased Data, not the model having any bigotry

Real-world biases leak into Machine Learning

- Bias comes from Biased Data, not the model having any bigotry
- Examples:

Real-world biases leak into Machine Learning

- Bias comes from Biased Data, not the model having any bigotry
- Examples:
  - AI in Law Enforcement: Biases and Racism in Law Enforcement can leak into model predictions.

Real-world biases leak into Machine Learning

- Bias comes from Biased Data, not the model having any bigotry
- Examples:
  - AI in Law Enforcement: Biases and Racism in Law Enforcement can leak into model predictions.
  - Dataset Generation might often be biased due to engineers not realizing the importance of diversity

Real-world biases leak into Machine Learning

- Bias comes from Biased Data, not the model having any bigotry
- Examples:
  - AI in Law Enforcement: Biases and Racism in Law Enforcement can leak into model predictions.
  - Dataset Generation might often be biased due to engineers not realizing the importance of diversity
  - When Google Translate converts news articles written in Spanish into English, phrases referring to women often become ‘he said’ or ‘he wrote’.

Real-world biases leak into Machine Learning

- Bias comes from Biased Data, not the model having any bigotry
- Examples:
  - AI in Law Enforcement: Biases and Racism in Law Enforcement can leak into model predictions.
  - Dataset Generation might often be biased due to engineers not realizing the importance of diversity
  - When Google Translate converts news articles written in Spanish into English, phrases referring to women often become ‘he said’ or ‘he wrote’.
  - Chat bot trained on data from tweets ”Tay” learns to be racist and sexist as a result of the sheer number bigoted twitter users

Real-world biases leak into Machine Learning

- Bias comes from Biased Data, not the model having any bigotry
- Examples:
  - AI in Law Enforcement: Biases and Racism in Law Enforcement can leak into model predictions.
  - Dataset Generation might often be biased due to engineers not realizing the importance of diversity
  - When Google Translate converts news articles written in Spanish into English, phrases referring to women often become ‘he said’ or ‘he wrote’.
  - Chat bot trained on data from tweets ”Tay” learns to be racist and sexist as a result of the sheer number bigoted twitter users
- What can we do?

Safety of AI

- The same model has drastically different performance for different hyperparameters.

Safety of AI

- The same model has drastically different performance for different hyperparameters.
- Should we let a medical robot with CNN-based vision system perform surgery autonomously?

Safety of AI

- The same model has drastically different performance for different hyperparameters.
- Should we let a medical robot with CNN-based vision system perform surgery autonomously?
- If a self-driving car crashes and hurts people, who should be responsible for it?

Outline

Course Takeaway

- ML is the combination of math and computer science.

Course Takeaway

- ML is the combination of math and computer science.
- We've only shown you a subsection
  - Supervised Learning: Linear/Logistic Regression and Neural Networks

Course Takeaway

- ML is the combination of math and computer science.
- We've only shown you a subsection
  - Supervised Learning: Linear/Logistic Regression and Neural Networks
- Deep learning has wide applications, but we are also responsible for its consequences. —The greater the power, the greater the responsibility!