U-1 (Deep Learning)

- Machine learning
  - subfield of AI which concerns in developing computational theories of learning & building learning machines.

- Importance of ML
  - Learns from data
  - automates tasks
  - Improve decision making
  - Personalizes experience
  - drives innovation.

Appl^n → NLP
Healthcare
Agriculture
Self driving cars
cybersecurity
Img & speech.

How Machines Learn –
- Training
- Validation
- Application

Data Collection,
Data preparation
Model Selection
Model Training
Model evaluation
Model Tuning
Deployment

- Deep Learning – Imitates way human gain certain type of knowledge. subset of AI.

Reasons for DL –
- Analyze unstructured data
- Data labelling
- Feature engineering
- efficiency
- Training

Appl^n → Aerospace & defense
Financial services
Medical research
Industrial automation
Facial recognition

Adv → no need for feature engineering
→ solves problem on end to end basis
→ gives more accuracy

Disadv → • high performance h/w
• more time to train
• difficult to assess its perform
• Hard to understand.

	ML	DL
	Field of Artifical Intelligence	Subset of Machine Learning
	structured & unstructured data	unstructed data (images, text, audio)
	smaller datasets	large datasets → good performance
	More interpretable models.	Less Interpretablee Models.
	Can run on CPU & GPUs	Can run on GPU or more powerful
	lesser accuracy	More accuracy
	Less complex models	More complex models.
	manual feature extraction	automatic feature extraction
	Fast training time	Longer training times
	Learns patterns from data for prediction or decisions	Utilizes neural network with multiple layers of learning.
Usecases	Regression, classification, clustering	computer vision, NLP
ex →	Decision trees, SVM	CNNs, RNNs, DBNs.

Supervised Learning	Unsupervised Lear
• Learns from labelled data. with input-output pairs	• Learns patterns & structures from unlabelled data
• labelled training data	• Doesn't require labelled
• Predicts ouput based on input data	• Discovers hidden patterns or structure in data.
• Receives feedback during training	• Doesn't Receives feedback.
• can test our model	• Can't test our model
• Desired output is given	• desired output is not given
• Also called classification	• Clustering
• eg → classification, regretion, object detection	• K-Mean Clustering, PCA

Hypermaters — settings you define before training a model

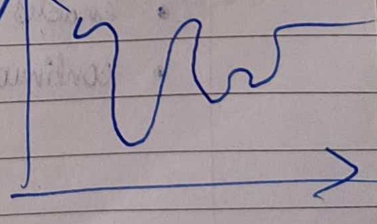
eg → Learning rate
batch size
no. of layers
no. of neurons per layer
activation function.

* Bias → phenomenon than skews results of an algo in favor or against.
  - Low Bias → fewer assumptions → form of target function → by model
  - High Bias → more assumptions → unable to capture imp features.
  - To reduce high bias →
    - Increase input feature
    - Decrease regularization time
    - Use more complicate models.
    - Increase training data
    - Increase regularization term

* Variance → changes in mode when using diff portions of training dataset.
  how much random variable differs from its expected value.
  - Model → high variance → low bias & vice versa.
  - To reduce high variance →
    - Reduce input feature
    - Don't use complex model
    - Increase training data

* Bias Variance Tradeoff -
  - Tradeoff b/w bias & variance is inversely proportional to variance & vice versa.
  - Tackle by either increasing a) complexity of model
    b) Training Dataset

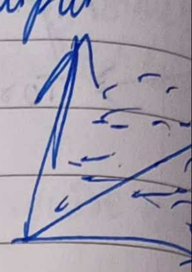
* Overfitting



- statistical model fits against its matching data.
- algo/model can't perform well on unseen data
- low bias & high variance.
- model → complex enough → match all datapoint & performs well
- Reasons → noisy data, training data is too small & large no of features.

To avoid Overfitting →
- Cross validation
- Train with more data
- Removing features
- Early stopping the training
- Regularization
- Ensembling

* Underfitting -



- Data model is unable to capture relⁿ b/w input & output
- occurs when model is too simple
- High bias low variance
- larger quantity of features
- less data → for model
- To avoid underfitting
  - ↳ growing education time of model
  - ↳ By increasing wide variety of functions.

* Learning Representation from data.

- Key Process in ML & DL
- Involves extracting features or representations from raw data.
- enables algo to understand & present process complex patterns.
- Representation learning essential for tasks like image & speech recognition
- Deep learning excels in automatically learning representations from raw data
- Captures hierarchical & abstract features the data.
- enables transfer learning, leveraging representations learned from one task to
- continuously evolving field with ongoing research in optimization & efficiency

How deep learning works in three figures -

Fig1 - Basic Architecture of a Neural Network.

1) Input layer → raw data or features
2) Hidden layers process → input through ~~output~~ interconnected neurons.
3) each neuron applies a weighted sum & activation function.
4) Output layer → predictions or classifications.

Fig2 - Convolutional Neural Network (CNN) for Image Recognition.

1) Input layer → pixel values of an image.
2) Convolutional layers apply filters to extract features.
3) Pooling layers reduce spatial dimensions, imp features.
4) Fully connected layers interpret features for final classification.
5) Utilizes local connectivity & weight sharing for efficiency.

Fig3 - Recurrent Neural Network (RNN) for Sequential Data Processing.

1) Input sequence is fed sequentially into network.
2) Recurrent connections allow information to persist over time.
3) Hidden state capture context from previous inputs.
4) Output generated at each time step or at end of sequence
5) suitable for tasks involving sequences like language modeling or time series prediction.

DOMS Page No.
Date / /

* Common Architectural Principals of Deep Networks -

- Parameters
- Layers
- Activation Function
- Loss fuctions
- optimization methods
- Hyperparameters.

* Architecture Design -

- customized for Tasks
- Depth vs Width
- Flexibility
- Efficiency
- Hardware Consideration
- Transfer Learning
- Interpretability
- Scalability
- Regularization Techniques
- Continuous Innovation.

* Popular Industry tools -

- Tensorflow -
  - open source framework by Google Brain Team.
  - used for deep learning model building
  - offers scalability & flexibility
  - ecosystem → Tensorflow lite, TFX, Tensorflow.js
  - applied in img recognition, NLP, reinforcement

- Keras -
  - High level neural networks API in Python
  - simplifies model building with minimal code
  - supports convolutional & recurrent networks.
  - Integrates with Tensorflow, CNTK, Theano.
  - Used in research, education & rapid development.

- Pytorch - • open source framework by Framework AI Research.
  - Known for dynamic computation graph.
  - Provides flexibility & GPU acclelation.
  - Integrates seamlessly with Python.
  - Used in research, academia, image classification.

- ~~Tyt~~ Caffe - • Developed by Berkeley AI Research (BAIR).
  - Primarily for CNNs & computer vision
  - Optimized for speed & efficiency.
  - Offers pre-trained models in Model Zoo.
  - Applied for in object recognition, face recognition.

- Shogun - • Open source ML library for various algorithms.
  - Supports, supervised, unsupervised, kernel
  - Flexible, scalable & interoperable.
  - Modular design for combining algorithms.
  - Used in research, education, classification, regression.