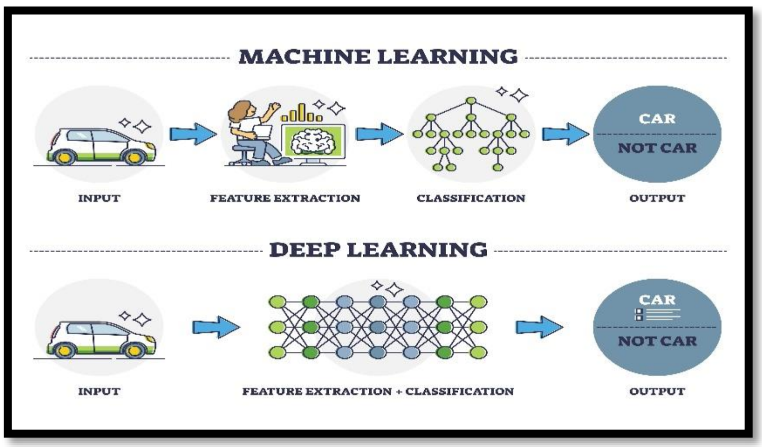
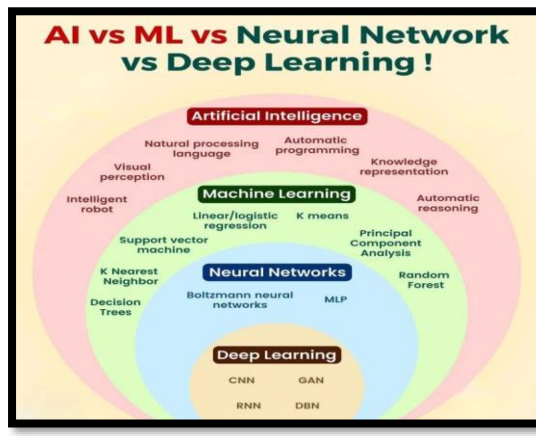
Major differences between ML and DL

Machine learning uses statistical learning algorithms to find patterns in available data and perform predictions and classifications on new data. ML also comprises both supervised and unsupervised learning.

Deep learning relies on multi-layered neural network models to perform complex tasks. There is a significant difference in the capabilities and applications of both. Understanding them is essential to knowing which to use in projects and get the best results.

Deep learning is a subset of machine learning that uses multilayered neural networks, called deep neural networks, to simulate the complex decision-making power of the human brain. Some form of deep learning powers most of the artificial intelligence (AI) applications in our lives today.
Deep learning can be used in a wide variety of applications, including: ***Image recognition***: To identify objects and features in images, such as people, animals, places, etc. Natural language processing: To help understand the meaning of text, such as in customer service chatbots and spam filters.





Three types of Deep Learning

Whether you are a beginner or a professional, these top three deep learning algorithms will help you solve complicated issues related to deep learning:

1. ***CNNs or Convolutional Neural Networks.***

CNNs are a deep learning algorithm that processes structured grid data like images. They have succeeded in image classification, object detection, and face recognition tasks.

How its work:

Convolutional Layer: This layer applies a set of filters (kernels) to the input image, where each filter slides (convolves) across the image to produce a feature map. This helps detect various features such as edges, textures, and patterns.
Pooling Layer: This layer reduces the dimensionality of the feature maps while retaining the most essential information. Common types include max pooling and average pooling.
Fully Connected Layer: After several convolutional and pooling layers, the output is flattened and fed into one or more fully connected (dense) layers, culminating in the output layer that makes the final classification or prediction.

2. ***LSTMs or Long Short Term Memory Networks.***

LSTMs are a special kind of RNN capable of learning long-term dependencies. They are designed to avoid the long-term dependency problem, making them more effective for tasks like speech recognition and time series prediction.

How it Works:

Cell State: LSTMs have a cell state that runs through the entire sequence and can carry information across many steps.
Gates: Three gates (input, forget, and output) control the flow of information:

Input Gate: Determines which information from the current input should be updated in the cell state.
Forget Gate: Decides what information should be discarded from the cell state.
Output Gate: Controls the information that should be outputted based on the cell state.

3. ***RNNs or Recurrent Neural Networks (RNNs).***

RNNs are designed to recognize patterns in data sequences, such as time series or natural language. They maintain a hidden state that captures information about previous inputs.

How it Works

- ***a.*** Hidden State: At each time step, the hidden state is updated based on the current input and the previous hidden state. This allows the network to maintain a memory of past inputs.
- ***b.*** Output: The hidden state generates an output at each time step. The network is trained using backpropagation through time (BPTT) to minimize prediction error.

4. ***Generative Adversarial Networks (GANs)***

GANs generate realistic data by training two neural networks in a competitive setting. They have been used to create realistic images, videos, and audio.

How it Works

- Generator Network: Creates fake data from random noise.
- Discriminator Network: Evaluates the authenticity of the data, distinguishing between real and fake data.
- Training Process: The generator and discriminator are trained simultaneously. The generator tries to fool the discriminator by producing better fake data, while the discriminator tries to get better at detecting counterfeit data. This adversarial process leads to the generator producing increasingly realistic data.

5. ***Transformer Networks***

Transformers are the backbone of many modern NLP models. They process input data using self-attention, allowing for parallelization and improved handling of long-range dependencies.

How it Works

- Self-Attention Mechanism: This mechanism computes the importance of each part of the input relative to every other part, enabling the model to weigh the significance of different words in a sentence differently.
- Positional Encoding: Adds information about the position of words in the sequence since self-attention doesn't inherently capture sequence order.
- Encoder-Decoder Architecture: Consists of an encoder that processes the input sequence and a decoder that generates the output sequence. Each consists of multiple layers of self-attention and feed-forward networks.

6. ***Autoencoders***

Autoencoders are unsupervised learning models for tasks like data compression, denoising, and feature learning. They learn to encode data into a lower-dimensional representation and then decode it back to the original data.

How it Works

- Encoder: Maps the input data to a lower-dimensional latent space representation.
- Latent Space: Represents the compressed version of the input data.
- Decoder: Reconstructs the input data from the latent representation.
- Training: The network minimizes the difference between the input and the reconstructed output.

7. ***Deep Belief Networks (DBNs)***

DBNs are generative models composed of multiple layers of stochastic, latent variables. They are used for feature extraction and dimensionality reduction.

How it Works

- ***Layer-by-Layer Training:*** DBNs are trained in a greedy, layer-by-layer fashion. Each layer is trained as a Restricted Boltzmann Machine (RBM), which learns to reconstruct its input.
- ***Fine-Tuning:*** After pretraining the layers, the entire network can be fine-tuned using backpropagation for specific tasks.

8. ***Deep Q-Networks (DQNs)***

DQNs combine deep learning with Q-learning, a reinforcement learning algorithm, to handle environments with high-dimensional state spaces. They have been successfully applied to tasks such as playing video games and controlling robots.

How it Works

- ***Q-Learning:*** Uses a Q-table to represent the value of taking an action in a given state.
- ***Deep Neural Network:*** Replaces the Q-table with a neural network that approximates the Q-values for different actions given a state.

- ***Experience Replay:*** Stores past experiences in a replay buffer and samples from it to break the correlation between consecutive experiences, improving training stability.
- ***Target Network:*** A separate network with delayed updates to stabilize training.

9. ***Variational Autoencoders (VAEs)***

VAEs are generative models that use variational inference to generate new data points similar to the training data. They are used for generative tasks and anomaly detection.

How it Works

- ***Encoder:*** Maps input data to a probability distribution in the latent space.
- ***Latent Space Sampling:*** Samples from the latent space distribution to introduce variability in the generated data.
- ***Decoder:*** Generates data from the sampled latent representation.
- ***Training:*** This method combines reconstruction loss and a regularization term to encourage the latent space to follow a standard normal distribution.

10. ***Graph Neural Networks (GNNs)***

GNNs generalize neural networks to graph-structured data. They are used for social network analysis, molecular structure analysis, and recommendation systems.

How it Works

- ***Graph Representation:*** Nodes represent entities, and edges represent relationships between entities.
- ***Message Passing:*** Nodes aggregate information from their neighbors to update their representations. This process can be repeated for several iterations.
- ***Readout Function:*** After message passing, a readout function aggregates node representations to produce a graph-level representation for tasks like classification or regression.

Reference:(https://www.simplilearn.com/tutorials/deep-learning-tutorial/deep-learning-algorithm#:~:text=Whether%20you%20are%20a%20beginner,Recurrent%20Neural%20Networks%20(RNNs).)

VIDEOs:

1. **https://www.youtube.com/watch?v=6M5VXKLf4D4**
2. **https://www.youtube.com/watch?v=d2kxUVwWWwU**