Generative AI

Data + Algorithms = AI Machines

Large size and High diversity dataset would be ideal to use for ML

Real-world Examples:

- **Apple Siri**
- **Netflix Recommendation**
- **Autonomous Vehicles** (Tesla, Waymo)
- **Chatbots**

Generative AI (Gen-AI)

Generative AI is a powerful subset of AI focused on creating new content, including:

- Text
- Images
- Audio
- Videos

Machine Learning (ML)

Machine Learning is a branch of computer science that uses data and algorithms to enable AI to imitate human learning, gradually improving accuracy through experience.

Definition by Tom Mitchell:

"A subset of AI algorithms that learn without being explicitly programmed with rules. They use data to learn and match patterns."

Types of Machine Learning:

1. **Supervised Learning**
2. **Unsupervised Learning**
3. **Reinforcement Learning**

Clustering

Clustering is an unsupervised machine learning technique designed to group unlabeled examples based on their similarity to each other. (If the examples are labelled, this kind of grouping is called classification.)

Types of Clustering

1. Centroid-based clustering:

The centroid of a cluster is the arithmetic mean of all the points in the cluster. Centroid-based clustering organizes the data into non-hierarchical clusters. Centroid-based clustering algorithms are efficient but sensitive to initial conditions and outliers. Of these, k-means is the most widely used. It requires users to define the number of centroids, k, and works well with clusters of roughly equal size.

2. Density-based clustering:

Density-based clustering connects contiguous areas of high example density into clusters. This allows for the discovery of any number of clusters of any shape. Outliers are not assigned to clusters. These algorithms have difficulty with clusters of different densities and data with high dimensions.

3. Distribution-based clustering:

This clustering approach assumes data is composed of probabilistic distributions, such as Gaussian distributions. As the distance from the distribution's centre increases, the probability that a point belongs to the distribution decreases. The bands show a decrease in probability.

4. Hierarchical clustering:

Hierarchical clustering creates a tree of clusters. Hierarchical clustering, not surprisingly, is well suited to hierarchical data, such as taxonomies. Any number of clusters can be chosen by cutting the tree at the right level.

- A regression model predicts a numeric value. For example, a weather model that predicts the amount of rain in inches or millimetres is a regression model.
- Classification models predict the likelihood that something belongs to a category. Unlike regression models, whose output is a number, classification models output a value that states whether or not something belongs to a particular category. For example, classification models are used to predict if an email is spam or if a photo contains a cat.

Classification models are divided into two groups: ***Binary classification and Multiclass classification***. Binary classification models output a value from a class that contains only two values. Multiclass classification models output a value from a class that contains more than two values.

- Once we're satisfied with the results from evaluating the model, we can use the model to make predictions, called inferences, on unlabeled examples.

Decision Forest

Decision forests are most effective when you have a tabular dataset (data you might represent in a spreadsheet, CSV file, or database table). Tabular data is one of the most common data formats, and decision forests should be your “go-to” solution for modelling it. Decision forests perform best when lots of data are available. Decision forests typically infer faster than comparable neural networks. Decision forest models are composed of decision trees.

A decision tree is a model composed of a collection of "questions" organized hierarchically in the shape of a tree. The questions are usually called a condition, a split, or a test. Each non-leaf node contains a condition, and each leaf node contains a prediction.
Inference of a decision tree model is computed by routing an example from the root (at the top) to one of the leaf nodes (at the bottom) according to the conditions. The value of the reached leaf is the decision tree's prediction. The set of visited nodes is called the inference path.

Types of conditions:
An axis-aligned condition involves only a single feature. An oblique condition involves multiple features.
For example, the following is an axis-aligned condition: num_legs ≥ 2
For example, the following is an oblique condition: num_legs ≥ num_fingers

Conditions with two possible outcomes (for example, true or false) are called binary conditions. Decision trees containing only binary conditions are called binary decision trees.

Non-binary conditions have more than two possible outcomes. Therefore, non-binary conditions have more discriminative power than binary conditions. Decisions containing one or more non-binary conditions are called non-binary decision trees.

The most common type of condition is the threshold condition expressed as: feature ≥ threshold

Deep Learning

Deep learning teaches computers to process data in a way inspired by the human brain, utilising artificial neurons.

Uses of Deep Learning

- Computer Vision

- Natural Language Processing

- Recommendation Engines

- Speech Recognition

Key Differences:

- **Black and White Images:**
  - **Definition:** Also known as binary images, containing only black and white.
  - **Color Depth:** Each pixel is either black (0) or white (1).
  - **Usage:** Ideal for simple graphics, text, or high-contrast images.
- **Grayscale Images:**
  - **Definition:** Contains shades of grey from black to white.
  - **Colour Depth:** Typically uses 8 bits per pixel for 256 different shades.
  - **Usage:** Common in photography, medical imaging, and applications requiring detailed brightness levels.

Grayscale images are most commonly utilized.

Deep Learning Architectures:

- **ANN (Artificial Neural Network):** Used for structured data.
- **CNN (Convolutional Neural Network):** Used for images and videos.
- **RNN (Recurrent Neural Network):** Designed for sequential data.

Object Detection vs. Object Recognition:

- **Object Detection:** Identifies and localizes objects using bounding boxes.
- **Object Recognition:** Categorizes objects without precise localization.

Recommendation System

A recommendation system helps users find compelling content in a large corpus.

Embedding: A mapping from a discrete set (in this case, the set of queries or the set of items to recommend) to a vector space called the embedding space. Many recommendation systems rely on learning an appropriate embedding representation of the queries and items.

The primary components of a recommender system are candidate generation, scoring, and re-ranking.

Feed-Forward Neural Network (FNN)

A type of neural network where connections do not form cycles. Information flows in one direction: from input to output.

Structure:

1. **Input Layer:** Receives input data, with each neuron representing a feature.
2. **Hidden Layers:** Learn complex patterns from the data.
3. **Output Layer:** Provides final outputs based on the classification or regression problem.

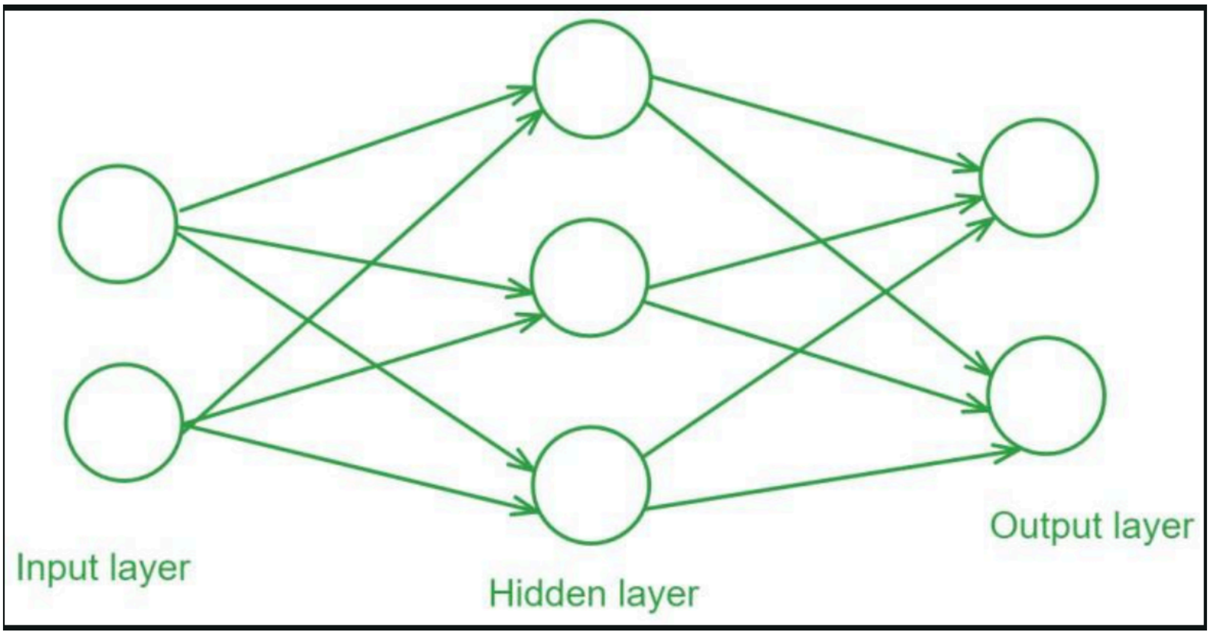


fig: Feed-Forward Neural Network

Backpropagation

An iterative algorithm that minimizes the cost function by adjusting weights and biases. During each epoch, the model learns by adapting these parameters to reduce loss.

Activation Function: Rectified Linear Unit (ReLU)

The ReLU function outputs the input directly if positive; otherwise, it outputs zero. It's widely used due to its effectiveness in training and performance.

Example Code for Neural Network:

```
model.add(Dense(16, input_dim=X_train.shape[1], activation='relu'))
# Input + Hidden Layer
model.add(Dense(16, activation="relu"))  # Second Hidden Layer
```

Adam Optimizer

The Adaptive Moment Estimation (Adam) algorithm is efficient for optimization in gradient descent, particularly with large datasets.

Epoch

An epoch refers to one complete pass through the entire training dataset. During an epoch, Every training sample in the dataset is processed by the model, and its weights and biases are updated in accordance with the computed loss or error.

Real-world Applications of CNN:

1. Object Detection
2. Optical Character Recognition (OCR)
3. Facial Recognition
4. Self-Driving Cars
5. Healthcare
6. Agriculture
7. Security

Layers of CNN:

1. **Convolution:** Extracts features from images.
2. **Max Pooling:** Reduces resolution while retaining important features.
3. **Flattening:** Converts the pooled feature map into a flat vector.
4. **Full Connection:** Links the flattened vector to the output.

Convolution and Padding

- **Convolution:** A mathematical operation that extracts features by learning image characteristics using small data squares.
- **Padding:** Adds zeros to the input matrix to maintain spatial dimensions after filtering.

Pooling Layers:

- **Max Pooling:** Selects the maximum element in the feature map region.

- **Average Pooling:** Computes the average of the elements in the feature map region.

Natural Language Processing (NLP)

NLP gives computers the ability to interpret and understand human language.

Challenges of NLP:

1. Ambiguity
2. Contextual Variability
3. Slang
4. Sarcasm

NLP Pipeline Steps:

1. Data Acquisition
2. Text Preprocessing
3. Feature Engineering
4. Modelling and Evaluation
5. Deployment

Text Processing Techniques:

Basic Text Processing:

- HTML tag removal
- Handling emojis
- Basic spell checks
- Tokenization
- Stop word removal
- Stemming/Lemmatization
- Lowercasing

Advanced Text Processing:

- Parts of Speech Tagging
- Parsing
- Coreference Resolution

Stop Words:

Stop words are common words removed from text-processing tasks due to their insignificance. Examples include: "the", "and", "is", "a", "in".

Recurrent Neural Network (RNN)

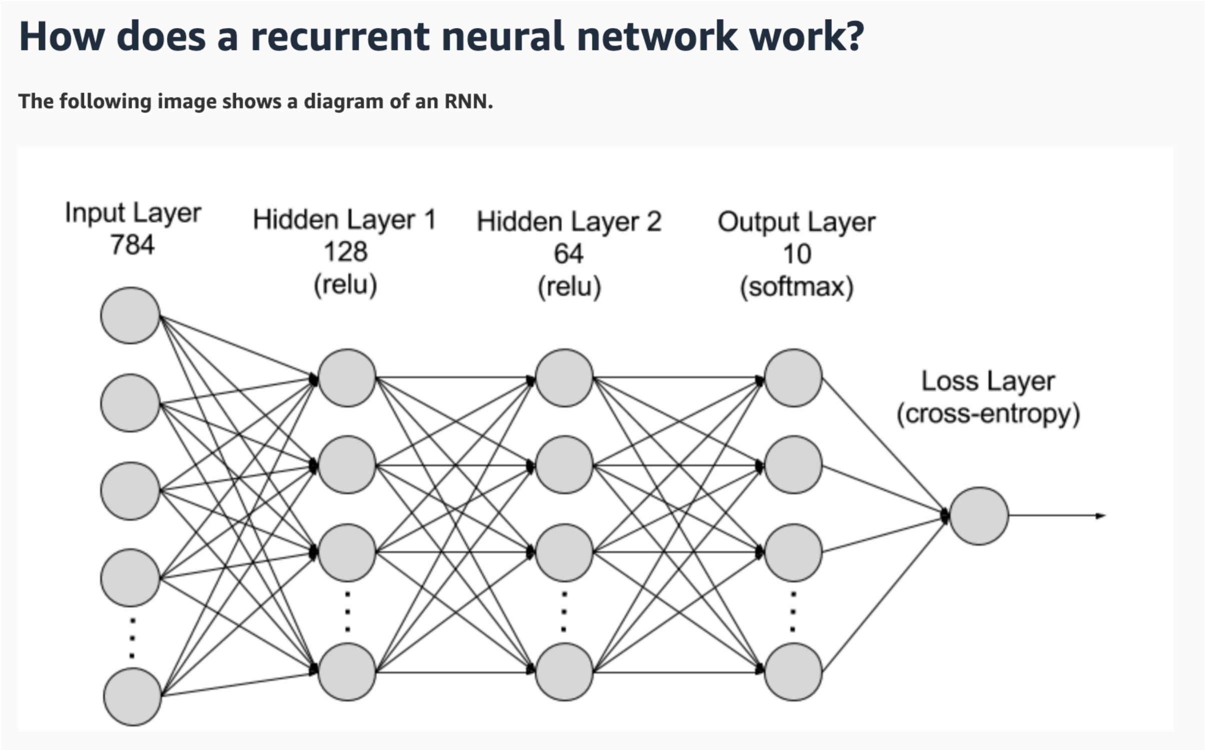
An RNN processes sequential data inputs to produce specific sequential outputs.

Long Short-Term Memory (LSTM)

An RNN variant that expands memory capacity to accommodate longer timelines, overcoming RNN limitations.

Limitations of RNN:

1. Exploding Gradient
2. Vanishing Gradient
3. Slow Training Time



How does a recurrent neural network work?

The following image shows a diagram of an RNN.

Key-Concepts behind Gen-AI

1. GANs (Generative Adversarial Networks)

2. Transformers

3. VAEs (Variational Autoencoders)

4. Diffusion Models

Image Generation

Image Generation is a process of using deep learning algorithms such as VAEs, GANs, and, more recently, Stable Diffusion to create new images that are visually similar to real-world images. Image Generation can be used for data augmentation to improve the performance of machine learning models, as well as in creating art, generating product images, and more.

Example: MidJourney, DALL-E

Video Generation

Video Generation involves deep learning methods such as GANs and Video Diffusion to generate new videos by predicting frames based on previous frames. Video Generation can often be seen in use with Speech Generation. The models used for speech generation can be powered by Transformers. Speech Generation can be used in text-to-speech conversion, virtual assistants, and voice cloning.

Example: DeepBrain and Synthesia

Encoder-Decoder Architecture

I. Encoder: The encoder is responsible for processing the input sequence (e.g., a sentence or a sequence of data) and converting it into a fixed-size representation. The key function of the encoder is to summarise the input sequence into a more abstract form. The output of the encoder is the hidden or context vector, which captures the essence of the entire input sequence.

II. Hidden Vector: The hidden vector (also called the context vector) is the condensed representation of the input sequence generated by the encoder. This vector holds the crucial information about the input sequence in a form that can be passed to the decoder.

III. Decoder: The decoder takes the hidden vector from the encoder and generates the output sequence step by step. The decoder typically works by predicting the next token in the sequence, and it continues doing so until it produces the entire output.

Workflow:

A. Encoder: Processes the input sequence and outputs the hidden vector.

B. Hidden Vector: Contains the summarised representation of the input.

C. Decoder: Uses the hidden vector to generate the output sequence (one token at a time or all at once, depending on the architecture).

Real-life application of Encoder-Decoder

- Google machine translation
- Speech Recognition
- Time Series Analysis

Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) are a class of machine learning models designed for generating new data that resembles a given dataset. GANs are made up of two neural networks: a discriminator and a generator. They use adversarial training to produce artificial data that is identical to actual data.

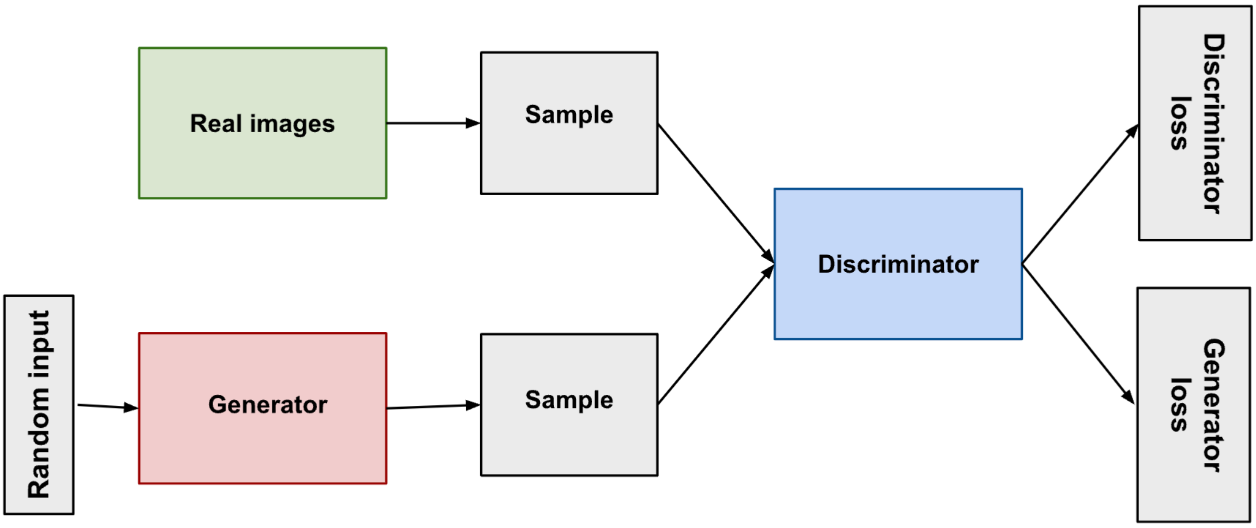


fig: GAN Structure

Key Components of GANs

I. Generator (G):

- The generator's role is to generate fake data (e.g., fake images, text, or sound) from random noise.
- It takes as input a random vector (usually sampled from a simple distribution like Gaussian noise) and tries to produce data that looks as close as possible to the real data.
- Over time, the generator learns to produce more realistic data through its interactions with the discriminator.

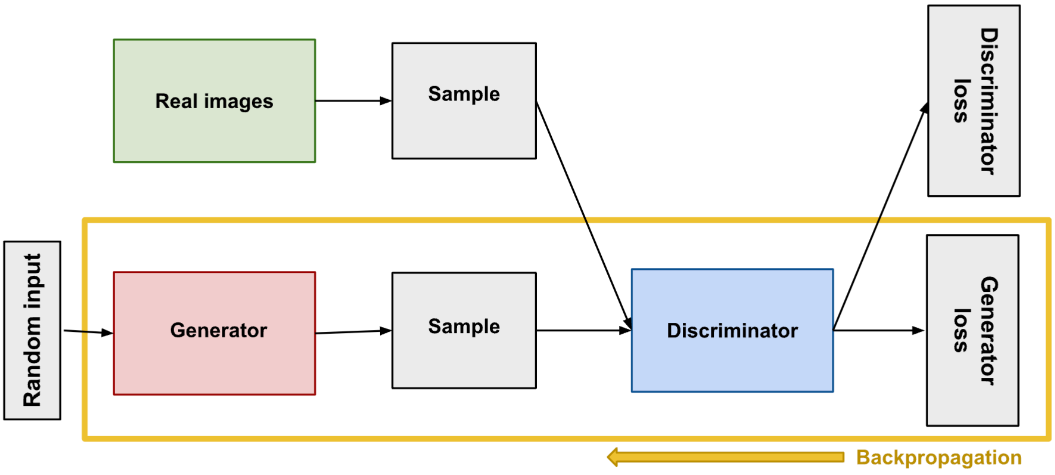


fig: Backpropagation in Generator Training

II. Discriminator (D):

- The discriminator's job is to distinguish between real data (from the actual dataset) and fake data generated by the generator.
- It's a binary classifier that outputs a probability indicating whether the input is real or fake.
- The goal of the discriminator is to get better at telling apart real data from the fake data generated by the generator.

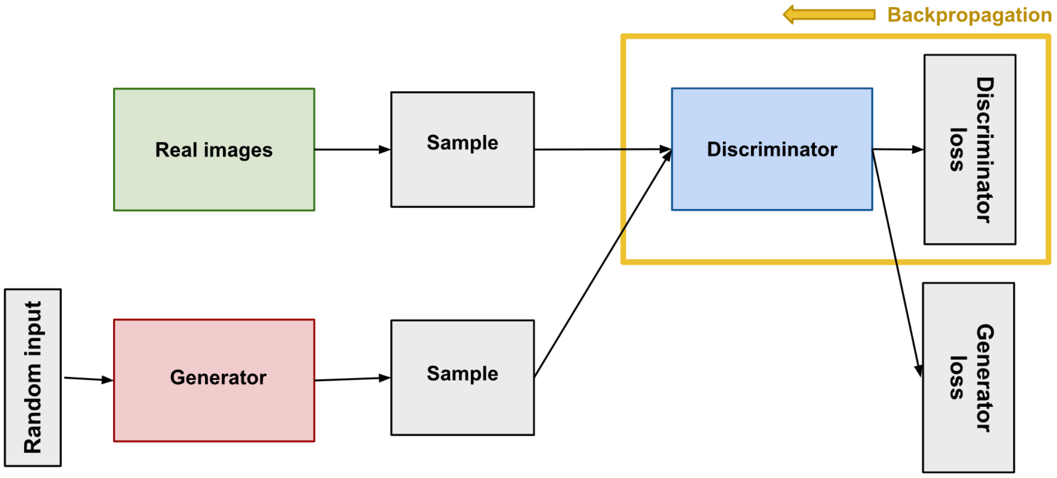


fig: Backpropagation in Discriminator Training

During generator training, gradients propagate through the discriminator network to the generator network (although the discriminator does not update its weights during generator training). So, the weights in the discriminator network influence the updates to the generator network.

A typical GAN alternates between training the discriminator and training the generator.

While a GAN can use the same loss for both generator and discriminator training (or the same loss differing only in sign), it's not required. It's more common to use different losses for the discriminator and the generator.

Small Language Models (SLMs) focus on specialised tasks with less data. Retrieval-augmented generation (RAG) enhances these models by pulling in external information for more accurate results. AI Agents use generative AI to autonomously perform tasks such as writing or research. Together, they represent cutting-edge advancements in automation and creativity.

Data Augmentation

Data augmentation is a process of generating new training data by applying various image transformations such as flipping, cropping, rotating, and colour jittering. The goal is to ***increase the diversity of training data*** and avoid overfitting, which can lead to better performance of machine learning models. Deep learning models rely on large volumes of diverse data to develop accurate predictions in various contexts. Data augmentation supplements the creation of data variations that can help a model improve the accuracy of its predictions.

Transformers

Transformers are a type of neural network architecture that transforms or changes an input sequence into an output sequence. They do this by learning context and tracking relationships between sequence components.

Transformer models fundamentally changed NLP technologies by enabling models to handle such long-range dependencies in text.

It uses a ***self-attention mechanism*** to capture relationships between different words (or tokens) in a sequence, regardless of their distance from each other, allowing the model to weigh the importance of each word when generating or processing text. Instead of processing data in order, the mechanism enables the model to look at different parts of the sequence all at once and determine which parts are most important.

Benefits of Transformers:

- Enable large-scale models
- Enable faster customization
- Facilitate multi-modal AI systems
- AI research and industry innovation

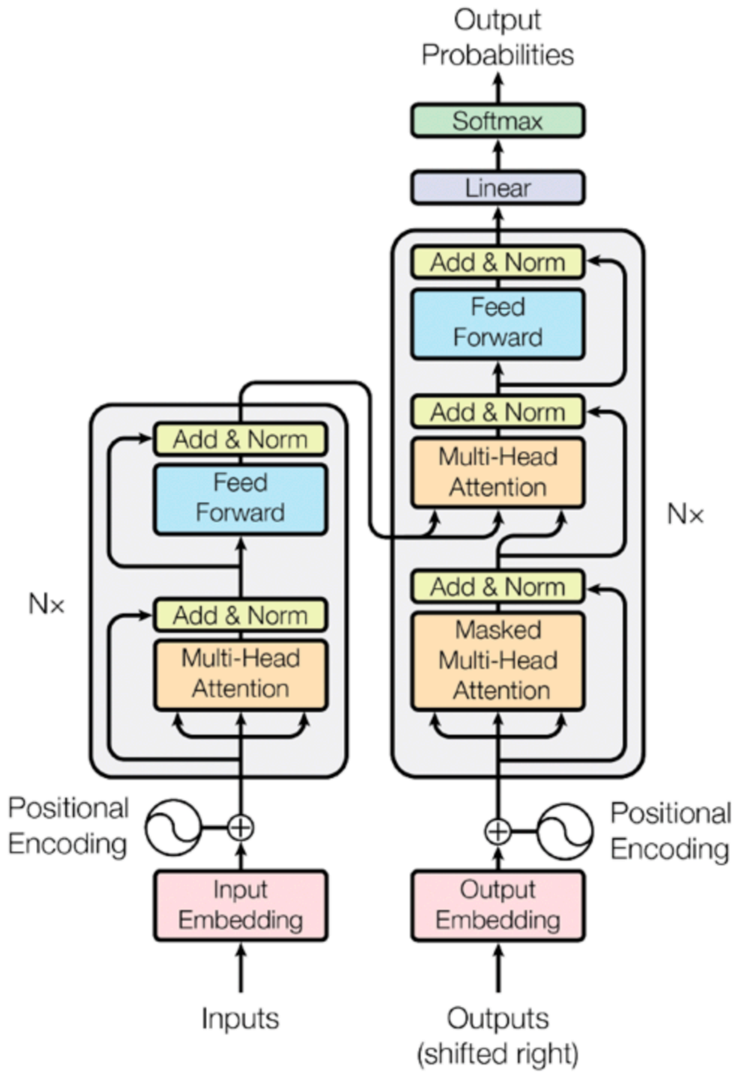


fig: Attention Layer in Transformer