

Deep Learning

Table of Contents

- [1. Artificial Intelligence](#)
 - [1.1. Machine Learning](#)
 - [1.1.1. Deep Learning](#)
- [2. Neural Network/Artificial Neural Network\(ANN\)](#)
- [3. Machine learning challenges](#)
- [4. Applications of ML](#)
- [5. Deep Neural Networks](#)
- [6. Limitations of Neural Network](#)
 - [6.1. Overfitting:](#)
 - [6.2. Underfitting:](#)
- [7. Introduction to TensorFlow](#)

1. Artificial Intelligence

What is intelligence? Intelligence is the ability to process information which inform you future decision making ability. **Artificial Intelligence** is the ability to give computers an intelligence ability. AI is any technique that enables computers mimic human behavior.

1.1. Machine Learning

What is learning? Learning is any process by which a system improves performance from experience. **Machine Learning** is a subset of **AI** which is the ability to learn without explicitly being programmed. That is processing information and informing a future decision making ability directly from data. **Data mining**: is a process that fuels AI. It involves sifting through massive dataset to uncover hidden patters, trends, and relationships. This extracted knowledge is the used to train AI models, enabling them to and make intelligence decisions. **Machine Learning** is great for:

- Problems in which existing solutions require a lot of hand-tuning or long list of rules.
- Complex problems for which there is no good solution at all using traditional approach.
- Fluctuating environment- a ML system can adapt to new data
- Getting insights about complex problems and large amount of data.

1.1.1. Deep Learning

Deep learning is a subset of machine learning that extracts patterns from data using neural networks.

1. Perceptron

The structural building block of deep learning. Every single neural network is built up on multiple perceptrons.

2. Neural Network/Artificial Neural Network(ANN)

A neural network is a machine learning program, or model, that makes decisions in a manner similar to the human brain by using processes that mimic the way biological neurons work together. Every neural network consists of layers of nodes, or artificial neurons - *an input layer, one or more hidden layers, and an output layer*. Each node connects to others, and has its own associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network. One of the best known neural network is Google's search algorithm.

ANN is a subset of machine learning and at the heart of deep learning. If deep learning is a subset of machine learning, how do they differ?

Deep learning distinguishes itself from the classical machine learning by the *type of data* that it works with and *the methods* in which it learns. Machine learning algorithms leverages *structured, labeled data* to make predictions- meaning that specific features are defined from the input data for the model and organized into tables. This doesn't necessary mean that it doesn't use unstructured data; it just means that if it does, it generally goes through some pre-processing to organize it into a structured format.

Deep learning eliminates some of the data pre-processing that is typically involved with machine learning. These algorithms can ingest and process unstructured data, like texts and images, and it automates feature extraction, removing some of the dependency on human experts. Keys that distinguish machine learning from deep learning are:

- Number of layers they use(deep learning > 3 layers)
- human intervention (Machine learning needs human intervention for labeling)

Machine learning is capable of different types of learning, based on whether or not they 're trained with human supervision.

1. *Supervised learning(Classification vs Regression)*: It utilizes labeled datasets(desired solutions) to categorize or make predictions. This requires some kind of human intervention to label input data correctly. Most important supervised learning algorithms are:
 - K-Nearest neighbors
 - Linear and logistic regression
 - Support vector machines(SVMs)
 - Decision tree and random forests
 - Neural networks
2. *Unsupervised learning*: It doesn't require labeled datasets, and instead it detects patterns in the data, clustering them by any distinguishing characteristics. Most important unsupervised learning algorithms are:
 - **Clustering**
 - K-means
 - Hierarchical Cluster Analysis(HCA)
 - Expectation Maximization

- **Visualization and Dimensionality Reduction**
 - Principal Component Analysis(PCA)
 - Kernel PCA
 - Locally-Linear Embedding(LLE)
 - **Association rule learning**
 - Apriori
 - Eclat
3. *Semi-supervised learning*: It utilizes partially labeled training datasets, usually a lot of unlabeled data and a little bit of labeled data.
 4. *Reinforcement learning*: Is a process in which a model learns to become more accurate for performing an action in an environment based on a feedback in order to maximize the reward. The learning system, called an **agent** in this context, can observe the environment, select and perform an actions, and get **rewards** in return.

Types of ML based on whether or not they can learn incrementally on the fly:

1. Online Learning
 - You train the system incrementally by feeding it data instances sequentially, either individually or small groups called *mini-batches*.
 - Each learning step is fast and cheap, so the system can learn about new data on the fly, as it arrives.
2. Batch Learning:
 - The system is incapable of learning incrementally.
 - It must be trained using all the available data, then evaluated on separate test data.
 - First the system is trained, and then it is launched into production and runs without learning anymore; it just applies what it has learned.
 - This is called offline learning, and it generally consumes a lot of time and computing resources.

Types of ML based on whether they work by simply comparing new data points to known data points, or instead detects patterns in the training data and build a predictive model, much like scientists do:

1. Instance based learning
2. Model based learning

3. Machine learning challenges

- Insufficient quantity of training data
- Non-representative training data The training data must be representative of the new cases you want to generalize
- Poor-quality data The training data must be free of errors, outliers, and noise
- Irrelevant features The training data must contain enough relevant features
- Overfitting the training data A model that works well on the training set but is not able to generalize new data. It has **high variance**
- Under-fitting the training data A model that doesn't work well on both training or new datasets. It has **high bias**

4. Applications of ML

- Recognizing patterns
 - Facial identities or facial expressions
 - Handwritten or spoken words
 - Medical images
- Generating patterns
 - Generating images or motion patterns.
- Recognizing anomalies
 - Unusual credit card transaction
 - Unusual patterns of sensor readings in a nuclear power plant
- Prediction
 - Future stock price or currency exchange rates.

5. Deep Neural Networks

Deep neural networks consist of multiple layers of interconnected nodes, each building up on the previous layer to refine and optimize the prediction or categorization. This progression of computation through the network is called **forward propagation**. The *input layer*, where the deep learning model ingests the data for processing, and the *output layer*, where the final prediction or classification is made, of a neural network are called **visible layers**. When ANN has two or more hidden layers, it's called **Deep Neural Network (DNN)**. Another process called **back propagation** uses algorithms, like gradient descent, to calculate errors in predictions and then adjusts the weights and biases of the function by moving backward through the layers in an effort to train the model. There are different types of neural networks to address specific problems or datasets. For example:

- **Convolutional Neural Networks (CNN)**: used primarily in computer vision and images classification applications, can detect features and patterns within an image, enabling tasks like object detection or recognition.

- **Recurrent Neural Networks (RNN)**: are typically used in natural language and speech recognition applications as it leverages sequential or time series data.

Deep learning applications:

- Law enforcement
- Financial services
- Customer services
- Health care.

An ANN is composed of **4** principal objects:

- *Layers*: All the learning occurs in the learning. There are **3** layers: input, hidden, output layers.
- *The input data and the corresponding target*
- *Loss function/cost function/error function*:
 - gives us a way to measure the performance of the network during the training period, to measure error of the network. So when the network gives prediction about data,

we can calculate the loss based on the prediction set the network gives against the true labels of the data.

- Quantifies the error between output of algorithm and given target value
- **Optimizer:** Updates the model in response to the output of the loss function. Improves the learning by updating the knowledge in the network.

Activation function: is a function that uses to decide whether to fire the neuron or not. It defines the output given a set of inputs. You need an activation function to allow the network to learn non-linear pattern. why **non-linear**? Because data of this world is non-linear. A common activation functions are:

- *Rectified Linear Unit(ReLU), sigmoid function, Tanh,...*

Relu: gives zero for all negative values and the value itself for positive values. **Loss function:** Used to measure the performance of the learning. Supervised learning has two types:

- **Classification:** It predicts the class of the dataset based on the independent input variable. Class is the categorical or discrete values. Like the image of an animal is a cat or a dog?
- **Regression:** It predicts the continuous output variables based on the independent variable. Like the prediction of house price based on different parameters like house age, distance from the main road, location, area, etc...

For binary classification, when specifying loss function, it is common practice to use a **binary cross entropy loss function**. In the linear regression you can use the **mean square error**. The loss function is an important metric to estimate the performance of the optimizer. The optimizer will help improve the weights of the network in order to decrease the loss. There are different optimizers available, but the most common one is **Stochastic Gradient Descent**. Conventional optimizers are:

- Momentum optimization
- Nesterov Accelerated Gradient
- AdaGrad
- Adam(Adaptive moment estimation) Optimization

6. Limitations of Neural Network

6.1. Overfitting:

A common problem with the complex neural net is the difficulties in generalizing unseen data. Overfitting is when a model offers ideal predictions when tested against training data but fails against new, unidentified(validating) data. This scenario is observable when:

- The model is highly complex or convoluted
- The model over trains on a single or specific dataset, warping its ability to analyze new data
- The training data contains inapplicable information or noise, which can taint the model's algorithm.

How to prevent overfitting?

1. Increase the volume of training data
2. Introduce data augmentation

3. Halt training when necessary- you can stop the training process before a model becomes too focused on minor details or noise the training data.

If the data are unbalanced within groups (i.e, not enough data available in some groups), the network will learn very well during training but will not have the ability to generalize such patterns to **never-seen-before** data. There is a trade-off in ML between optimization and generalization. **Optimizing** a model requires to find the best parameters that minimize the loss of the training set. **Generalization**, however, tells how the model behaves for unseen data. To prevent the model from capturing specific details or unwanted patterns of the training data, you can use different techniques. The best method is to have a balanced dataset with sufficient amount of data. The art of reducing overfitting is called **regularization**. Some of the conventional techniques:

- Network size
- Weight regularization
- Drop out

6.2. Underfitting:

An underfit model performs poorly both on training and new(validating) data. Underfitting in ML occurs when a model is too simplistic to capture or learn the underlying patterns in the training data. Other underlying reasons may include:

- Scanty or limited training data
- Inadequate model training time

e.g: you're using a weather forecasting model with only one variable, such as temperature, to predict rainfall. Devoid of crucial training factors like humidity, wind speed, or atmospheric pressure, the model will likely erroneously forecast rain due to a mere temperature decline.

How to prevent underfitting?

1. Maximize training time
2. Optimize model complexity
3. Minimize regularization

Detecting overfitting is trickier than spotting underfitting because overfitted models show impressive accuracy on their training data.

7. Introduction to TensorFlow

TensorFlow is an open source software library for **numerical computation using data flow graphs**. TensorFlow provides multiple APIs:

- Low level- gives complete programming control with high degree of flexibility.
- High level- which take care of repetitive tasks and low-level details.

Some key concepts regarding to TensorFlow: *Tensor*: Primary data structure of TensorFlow

A **Tensor** is a vector or matrix of n-dimensions that represent all types of data. All values in tensor hold identical data type with a known (or partially)known **shape**. *Feature vectors*: (in ML) will be the primary input to populate a tensor.

In TensorFlow, a tensor is a collection of feature vectors(i.e, array) of n-dimensions. A *computational graph* is a series of TensorFlow operations. The following two principles are used by TensorFlow core:

1. computational graph
2. Run the computational graph

Session: an object that encapsulates the environment in which operation objects are executed.

- are objects that place operations onto devices such as CPUs or GPUs

A place holder: is a promise to provide a value later.