1. Introduction

Artificial Intelligence (AI) is perhaps the most talked about topic in technology today. Autonomous cars, intelligent robots, the "lights out" automated factory were the stuff of science fiction writers a decade ago, but now are within the reach of reality. But what is artificial intelligence? The term “AI” describes machines that mimic cognitive functions linked with the human mind, like solving problems and learning. How does AI work? In this learning module we attempt to answer those questions and more. We will cover an overview of AI today, spanning a brief history, fundamental concepts, algorithms, and the Bayes Theorem. We will also covers machine learning, deep learning, and typical AI applications.

2. Objectives

Upon completion of this learning module, you will be able to:
- Discuss the history of Artificial Intelligence
- Understand the fundamentals of Artificial Intelligence
- Explain the basics of machine learning
- Define Deep Learning
- Discuss how Artificial Intelligence is applied today

3. Scope

The concept of AI is not new. Artificial Intelligence is mentioned in antiquity, in the form of myths and stories, with non-human characters being gifted with consciousness or intelligence by their creators. Classical philosophers seeded the modern concept of AI, with an attempt to describe the human thinking process as mechanical manipulation of various symbols. In the 17th century, Gottfried Wilhelm Leibniz, Rene Descartes, and Thomas Hobbes explored the
possibility of rational thought being as systematic as geometry or algebra. These philosophers had started to express the important physical symbol hypothesis, which would later become the guiding light of AI research.

- **3.1 The Birth of Modern AI**

AI as we know it now was founded during a conference held at Dartmouth College, Hanover, New Hampshire in 1956. The term “artificial intelligence” was formally adopted at the conference. Marvin Minsky, the MIT cognitive scientist who attended the Dartmouth conference, was optimistic about the new field’s future. His positive outlook, however, was premature at that time, as government funding for this new field soon dried up. Research picked up again in 1994, and AI came into the public eye when IBM’s Deep Blue® became the first computer in human technological history to beat Garry Kasparov, the reigning Russian chess grandmaster, in 1997.

AI again came into prominence when Eugene Goostman, the talking computer and “chatbot,” fooled judges into thinking of it as an actual human being in 2011. The chatbot had officially passed the Turing test, a competition put together by Alan Turing, a British mathematician in 1950. The test is a way to find out whether a machine can be considered intelligent.

AI is now an essential component of the technology industry, assisting to solve multiple problems in operations research, computer science, and software engineering. The scope of this learning module is to introduce you to the basic concepts of AI and how AI works.

### 4. Basic Concepts

AI is a complex subject. We will start by introducing the fundamental concepts related to this technology.

- **4.1 Definition of Intelligence**

A system’s ability to reason and calculate, recognize analogies and relationships, along with learning from experience together with storing and subsequent retrieval of information from memory is known as intelligence. On a practical level, intelligence implies the ability to find solutions to problems, understand complex ideas, and use the natural language in a fluent manner. An intelligent entity must have the ability not only to classify and generalize situations but also to adapt to them.

- **4.2 The Elements of Intelligence**

The intelligence in AI is composed of the following elements:
Knowledge: That part of AI which deals with comprehending, designing, and implementing methods of representing information collected from the environment and stored in computers. They make sense of the information and plan future activities. They can also solve problems in areas which normally require human expertise.

Reasoning: A software system that makes decisions from available knowledge, using logical methods like presumption and training, is termed as Reasoning. Interactive and batch processing are two modes of reasoning. An Interactive system can interface with the user to ask questions and help guide the user through the reasoning process. Batch systems also take into account all the existing information simultaneously and produce the optimum answer without feedback or guidance.

Learning: The act of gaining knowledge or skill by studying, being taught, practicing, or undergoing any action.

Problem Solving: A process in which one first observes and then attempts to reach a preferred solution from a present situation by adopting a path obstructed by known or unknown hurdles.

Perception: The technique of obtaining, comprehending, selecting, and consolidating sensory information. Perception and sensing are intertwined. Sensory organs support perception. With respect to AI technology, perception refers to data acquired by sensors in any meaningful manner.

Inference: Inference provides a suitable computational framework for perception. It also includes specific models which have been proposed for how it is present in neural systems. An open problem is better comprehension of the relation between perception and action, and how inferential computations seamlessly fit into the bigger framework of sensory motor systems.

Communication: The ability to utilize, understand, communicate (both verbal and written) through language.

- 4.3 Intelligent Agents

An agent is any entity that can view or perceive its environment through sensors, and take an action using actuators. A human agent uses sensing organs, like ears and eyes, and acts through actuators, which take the form of hands and legs. An electronic agent uses software to understand the text and generate textual responses as humans do. A robotic agent may have cameras and range finders to understand its surroundings and subsequently perform actions using actuator motors.
The goal of AI is to design an agent program that implements the agent function, meaning the mapping from percepts to actions. These programs will run on a computing device equipped with sensors and actuators, as part of its architecture.

\[ \text{Agent} = \text{Architecture} + \text{Program} \]

- **4.4 Types of Agents**

There are four kinds of agents which embody the principles underlying almost every intelligent system:

**Simple reflex agents**: These agents select actions based on the current percept, ignoring the rest of the percept history. One excellent example of this type of agent is an automated taxi. If the car in front brakes and its brake lights turn on, then the car captures the information and initiates braking in the agent program. Such a connection is called a condition-action rule. Simple reflex agents are basic, but they have limited intelligence.

![Figure 1: A simple reflex agent](image)

**Model-based agents**: The exact way to manage limited observability is for an agent to keep track of the portion of the world it cannot now see. This means the agent should uphold some internal state, which depends on the percept history, and thus replicates at least a few of the unobserved aspects of the current state. For the braking problem, the internal state is small — simply the previous frame from the camera, allowing the agent to detect when two red lights at the edge of the vehicle go on or off simultaneously. A driving task requires changing of lanes, and the agent needs to keep a record of where the other cars are if it cannot see them all at once. This knowledge about how the world works, whether implemented in simple Boolean circuits or in complete scientific theories, is called a model of the world. An agent which uses such a model is called a model-based agent.
Goal-based agents: These agents work based on a goal, and make choices based on what is the best way to reach that goal. A good example of this type of agent is a taxi idling in traffic. The taxi can turn right, left, or go straight. The decision depends on the taxi’s final destination. The agent program can associate this with the model to select actions that achieve the goal. Figure 3 shows the goal-based agent’s structure. Sometimes goal-based action selection is straightforward. One good example of this is when goal satisfaction results immediately from a single action. In some instances, it will be trickier, such as when the agent must consider long sequences of twists and turns to find a way to achieve the goal. The agent’s goal involves search and planning.

Utility-based agents: We cannot depend only on goals to generate high-quality behavior, and often we have several actions, which all satisfy our goals. Goals provide a fundamental binary difference between happy and unhappy states. A more common performance metric must permit a comparison of different world states according to how happy they would make the agent. Happiness does not sound scientific, so we use the term “utility” instead. A utility-based agent keeps
track of its environment, which involves research on reasoning, learning, and perception.

Figure 4: Utility-based agents

- 4.5 Searching

Searching is a universal method of problem-solving in AI. The route of looking for a sequence of actions that reach a goal is termed a search. In a search algorithm, a problem is taken as an input and returns an output based on sequences of action. Once a solution is found, the recommended actions are executed.

A search problem requires the following:

- A **State Space**. A set of all probable states where you can be.
- A **Start State**. This is the state where a search begins.
- A **Goal Test**. A function returns the current state, and checks whether it has reached the goal state.

The solution to a search problem is a sequence of actions, called the plan that transforms the start state to the goal state. This plan is achieved via search algorithms.

- 4.6 Types of search Algorithms

There are many powerful search algorithms in existence, but search algorithms can be broadly divided into two categories:

**Uninformed Search Algorithms**: In uninformed search algorithms, no extra information exists on the goal node. The information is solely provided in the problem definition. To reach the goal from the start state, the plans differ only by the length of the actions or order. Uninformed search is also called a Blind search.
**Informed Search Algorithms:** In an informed search, the algorithms have information on the goal state. This information helps to search effectively, and the information is obtained by the heuristic. The latter is a function that guesses how the present state is closer to the goal state. The lesser the distance, the closer the goal.

- **4.7 Bayes Rule**

The Bayes Rule (also called Bayes Theorem) is an important principle used in AI to calculate the probability an occurrence, based on prior knowledge of conditions that might be related to the occurrence. It also helps in updating the predicted probabilities of an occurrence by with new information. The Bayes Rule is given as:

\[
P(A|B) = \frac{P(B|A)P(A)}{P(B)}
\]

Here, A is the event for which we want the probability, and B is the new evidence that is related to A in some way. Estimate \(P(B|A)\) is the probability of observing new evidence. \(P(A|B)\) is called the posterior. \(P(B)\) is the total probability of finding the evidence.

There are many algorithms based on the Bayes Rule in AI. We will better understand the Bayes Rule with the help of a Naive Bayes example.

**Naive Bayes**

Naive Bayes is an algorithm based on the Bayes Theorem, and is used for classification with assumptions about the independence of each input variable. The Naive Bayes model is useful for extensive data sets, and is easy to build. Let us consider Naive Bayes using a text classifier. Let us take an example of a movie review as shown in below table:

<table>
<thead>
<tr>
<th>Doc</th>
<th>Text</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I loved the movie</td>
<td>+</td>
</tr>
<tr>
<td>2</td>
<td>I disliked the movie</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>A great movie, a good movie</td>
<td>+</td>
</tr>
<tr>
<td>4</td>
<td>Poor acting</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>Great acting, a good movie</td>
<td>+</td>
</tr>
</tbody>
</table>
As we can see, there are ten unique words: I, Loved, the movie, disliked, a great, poor, acting, good. The classes for each document are given as positive or negative.

Let us convert the document into a feature vector. We will give numbering 1, 2 and 3 based on the repeatability of each word in a particular document as shown in below table:

<table>
<thead>
<tr>
<th>Doc</th>
<th>I</th>
<th>Love</th>
<th>The</th>
<th>Movie</th>
<th>Dislike</th>
<th>A</th>
<th>Great</th>
<th>Poor</th>
<th>Acting</th>
<th>Good</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

Now let us look at the probabilities per outcome, i.e. + or -. Let us do the positive first. Document with positive outcomes

<table>
<thead>
<tr>
<th>Doc</th>
<th>I</th>
<th>Love</th>
<th>The</th>
<th>Movie</th>
<th>Dislike</th>
<th>A</th>
<th>Great</th>
<th>Poor</th>
<th>Acting</th>
<th>Good</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

The document is positive in 3 out of 5 documents:

P(+) = 3/5 = 0.6

Now we have to compute the probabilities of each individual word:

Compute: P(love|+), P(I|+), P(the|+), P(movie|+), P(a|+), P(great|+), P(acting|+), P(good|+)

If n is the number of words in the (+) case: i.e 14 and nk the number of times word k occurs in these cases (+).
Let \( P(W_k|+) = \frac{(n_k+1)}{n_+|Vocabulary|} \)

\[
P(I|+) = \frac{1+1}{14+10} = 0.8333 \quad \text{P(Loved|+)} = \frac{(1+1)}{(14+10)}=0.8333
\]

\[
P(\text{the}|+) = \frac{(1+1)}{(14+10)}=0.8333 \quad \text{P(movie|+)} = \frac{(4+1)}{(14+10)}=0.2083
\]

\[
P(a|+) = \frac{2+1}{14+10} = 0.125 \quad \text{P(great|+)} = \frac{2+1}{14+10} = 0.125
\]

\[
P(\text{acting}|+) = \frac{1+1}{14+10} = 0.8333 \quad \text{P(good|+)} = \frac{2+1}{14+10} = 0.125
\]

\[
P(\text{disliked}|+) = \frac{0+1}{14+10} = 0.417 \quad \text{P(poor|+)} = \frac{0+1}{14+10} = 0.417
\]

Now let us look at the negative examples:

<table>
<thead>
<tr>
<th>Doc</th>
<th>I</th>
<th>Love</th>
<th>The</th>
<th>Movie</th>
<th>Dislike</th>
<th>A</th>
<th>Great</th>
<th>Poor</th>
<th>Acting</th>
<th>Good</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td>-</td>
</tr>
</tbody>
</table>

The document is negative in 2 out of 5 documents:

\[ P(-) = \frac{2}{5} = 0.4 \]

\[
P(I-) = \frac{1+1}{6+10} = 0.125 \quad \text{P(the|-)} = \frac{1+1}{6+10} = 0.125
\]

\[
P(\text{movie|-}) = \frac{1+1}{6+10} = 0.125 \quad \text{P(disliked|-}) = \frac{1+1}{6+10} = 0.125
\]

\[
P(\text{poor|-}) = \frac{1+1}{6+10} = 0.125 \quad \text{P(acting|-}) = \frac{1+1}{6+10} = 0.125
\]

\[
P(\text{loved|-}) = \frac{0+1}{6+10} = 0.0625 \quad \text{P(a|-}) = \frac{0+1}{6+10} = 0.0625
\]

\[
P(\text{great|-}) = \frac{0+1}{6+10} = 0.0625 \quad \text{P(good|-}) = \frac{0+1}{6+10} = 0.0625
\]
Now that we have trained our classifier. Let’s now classify a new sentence:

“I disliked the poor acting.”

Let us say V is a value or class:

If $V_j = +$;

$$P(+)(I|+) P(\text{disliked}|+) P(\text{the}|+) P(\text{poor}|+) P(\text{acting}|+) = 6.03 \times 10^{-7}$$

If $V_j = -$;

$$P(-)(I|-) P(\text{disliked}|-) P(\text{the}|-) P(\text{poor}|-) P(\text{acting}|-) = 1.22 \times 10^{-5}$$

Hence, as the value of the negative class is higher, we can state that the sentence is a negative review.

- 4.8 Natural Language Processing

Natural Language Processing (NLP) denotes the AI technique of communicating where English or any other natural language is used to interact with intelligent systems. NLP is needed when you want a robot or any other intelligent system to perform as instructed. This condition is also applicable when there is a need to hear any decision from dialogue-centric clinical systems or similar systems. The NLP field involves pushing computers to execute useful tasks with human-used natural languages. The NLP system’s input and output can be:

- Speech
- Written Text

NLP has two components:

**Natural Language Understanding (NLU):** Understanding involves the following tasks:
- Mapping the input given in natural language to useful representations.
- Analysis of the language’s different aspects.

**Natural Language Generation (NLG):** This process produces meaningful phrases and sentences in natural language form from any internal representation. It comprises the following:
- **Text planning** – Includes recovering relevant content from the knowledge base.
- **Text Realization** – Mapping sentence plan to sentence structure.
- **Sentence planning** – Includes a selection of required words, the formation of meaningful phrases, and setting the sentence tone.

**4.9 Steps in NLP**

There are five steps that comprise NLP. They are:

**Lexical Analysis** – Involves the identification and analysis of the word structure. The word lexicon means a collection of words, as well as phrases in any language. By extension, lexical analysis involves dividing the full-text chunk into paragraphs and sentences. It can also be parsed into words.

**Parsing or Syntactic Analysis** – Involves analyzing words in a sentence for grammar. It may also be arranging words in such a manner which shows the relationship that exists among the words.

**Semantic Analysis** – Extracts precise meaning or the dictionary meaning from a particular text. The latter is then checked for meaningfulness. This is achieved by mapping the syntactic structures and the objects in the task domain. Any semantic analyzer disregards sentences like “hot ice-cream”.

**Discourse Integration** – Any sentence meaning depends upon the preceding sentence’s meaning. In addition, it also imports the meaning of the sentence which immediately succeeds it.

**Pragmatic Analysis** – In this step, what was said gets re-interpreted based on what is actually meant. The process involves deriving those particular language aspects that need real-world knowledge.

**5. Machine Learning**

The term machine learning refers to the automated detection of meaningful patterns within data. Machine learning is a type of AI that allows a system to learn from data rather than through explicit programming. Machine learning, however, is not a simple process. In the past couple of decades, we witnessed machine learning become a standard tool in almost any task that requires information extraction from large data sets. Machine learning-based technology surrounds us: anti-spam software learns to filter our email messages, credit card transactions are secured by software that learns how to detect frauds, and search engines learn how to bring us the best results. Digital cameras learn to identify faces, personal assistants on the phone learn how to recognize a voice, and cars also use machine learning algorithms to avoid accidents. Machine
learning is also commonly used in scientific applications such as bioinformatics, medicine, and astronomy.

Machine learning utilizes a bouquet of algorithms, which iteratively learn from available data to improve, describe the data, and consequently predict outcomes. As training data is fed into the algorithms, it is possible to produce precise models based on that specific data. The machine learning model is actually the generated output when the machine learning algorithm is trained with data. Post-training, when a model is provided input, the output will follow as a consequence. For example, a predictive model will be created from a predictive algorithm. When the predictive model is provided with data, a prediction based on that data will be sent. This data is used to train the model. Machine learning is critical for the creation of analytics models. ML is a combination of the following:

Data + model → compute prediction

Data is our observations. The model comprises our expectations based on previous knowledge, which can be derived from transfer learning, or merely can be our beliefs about the symmetries of the universe. In humans, our models include our inductive biases. The prediction is an action to be taken, a classification, or a quality score. The reason that ML has become a backbone of AI is the importance of guesses in AI.

- 5.1 When Do We Need Machine Learning?

The complexity of a given problem and the adaptive need together call for the use of programs which learn and then improve on the basis of experience. Tasks such as driving, image recognition, and speech recognition are highly complex and tough to program. Rigidity is the limiting feature of the programming tool. Once the program is written and subsequently installed, it remains unchanged regardless of the nature of the tasks changing over time, or from one specific user to another. Programs are tooled by machine learning to adapt their behavior to input data, and they consequently become adaptive to environmental changes.

- 5.2 Types of Learning

The types of learning can be classified in the following ways:

**Supervised learning:** This normally starts with an established data set and a certain comprehension of how that data gets classified. The intention of supervised learning is to find patterns in the data which can be applied to the analytics process. The meaning of this data can be defined by its labeled features. For example, there may be millions of animal images and explanations
of what each and every animal is, to help create a machine learning application which differentiates one particular animal from another. Preprocessed examples are used to train these algorithms. The performance of such algorithms is consequently evaluated with test data. Utilization of unforeseen data for the test set may help you to evaluate the model’s accuracy in predicting results outcomes. Supervised training models enjoy broad applicability to a number of business-related complications, including fraud detection, risk analysis, speech recognition, and recommendation solutions.

**Reinforcement learning:** This is a model of behavioral learning. Feedback is received by the algorithm after the data is analyzed so that the user is guided to the best outcome. This learning differs from other types of supervised learning as a sample data set is not used to train the system. The system learns by trial and error. A sequence of multiple successful decisions may result in that process being “reinforced” as it best solves the present problem.

**Unsupervised learning:** This type of learning is ideal when the problem needs a gargantuan amount of unlabeled data. For example, social media applications like Twitter, Snapchat, and Instagram are saddled with considerable quantities of unlabeled data. Comprehending the meaning buttressing this data requires algorithms which can start to understand meaning based on the ability to classify the data based on the patterns or clusters found. It means that an iterative process of data analysis is done by supervised learning without human intervention. Data is segmented by unsupervised learning algorithms into groups of clusters, examples, or features. These unlabeled data create parameter values and data classifications. The process adds labels to data so that the latter becomes supervised. It is thus clear that unsupervised learning can influence the outcome if data is present in huge amounts. In such a case, the developer is blind to the context of analyzed data, and labeling thus becomes impossible. It follows that unsupervised learning could be used as the first step prior to passing the data to any supervised learning process.

---

### 5.3 Machine Learning Algorithms

The selection of the best algorithm is part science and part art. Two data scientists tasked with solving the same business challenge may choose different algorithms to solve the same problem. However, understanding different classes of machine learning algorithms helps data scientists identify the best types of algorithms. This section gives you a brief overview of the main types of machine learning algorithms.

**Bayesian:** Bayesian algorithms allow data scientists to encode prior beliefs about what models should look like, independent of what the data states. These algorithms are especially useful when you do not have massive amounts of data to confidently train a model. A Bayesian algorithm would make sense, for
example, if you have prior knowledge of some part of the model and can, therefore, code that directly. A good example is that of a medical imaging diagnosis system that looks for lung disorders. If a published journal study estimates the probability of different lung disorders based on lifestyle, those probabilities can be encoded into the model.

**Clustering:** Clustering is just a straightforward method to understand objects with related parameters that are collected together (in a cluster). The objects in a cluster are related to each other and compared to objects in another cluster. In clustering, data is unlabeled and therefore it is a type of unsupervised learning.

**Decision Tree:** A decision tree algorithm uses a branching structure to explain the results of a decision. Decision trees are used to map the possible outcomes of a decision. Each node of a decision tree represents a possible outcome. Percentages are assigned to nodes based on the likelihood that the outcome that may occur.

**Dimensionality reduction:** Dimensionality reduction helps systems remove data that are not useful for analysis. This algorithms group is used to remove redundant data, non-useful data, and outliers. Dimensionality reduction is helpful when analyzing data collected from sensors, as well as other Internet of Things (IoT) use-cases. There could be huge numbers of data points in IoT systems simply informing you of a sensor being turned on. Storage and analysis of that “on” data may not be helpful, and moreover will occupy valuable storage space. Removal of this redundant data will help to improve the machine learning system’s performance. Dimensionality reduction will furthermore assist analysts in visualizing the data.

**Instance-based:** These algorithms are computed when it comes to categorizing new data points depending on their similarities to the training data. Instance-based algorithms are occasionally termed as lazy learners due to the lack of a training phase. The instance-based algorithms, instead, simply match the new data with the training data.

**Rule-based machine learning:** These algorithms use relational rules when it comes to describing data. A rule-centered system could be contrasted with machine learning systems which generate a model that may be typically applied to all incoming data.

**Linear regression:** These algorithms are typically used for statistical analysis. They are key algorithms for machine learning use. Regression algorithms help the analysts to forge model relationships between the data points. Regression algorithms have the advantage of being able to quantify the strength of correlation that exists between variables within a data set. Other than that, regression analysis could be useful to predict future data values based on their
historical numbers. It is vital to remember that regression analysis makes the assumption of correlation relating to causation. Inaccurate predictions can be obtained if the context around data is not understood.

- **5.4 The Machine Learning Cycle**

The selection of the correct machine learning algorithm comprises only one of the many steps. The machine learning cycle steps are:

**Data identification:** Identifying relevant data sources is the cycle’s first step. In addition, with the development of the machine learning algorithm, one should consider the expansion of the target data for system improvement.

**Data preparation:** The data must be clean and secured. It must be governed. There will be an application failure if inaccurate data is used to prepare a machine learning application.

**Selection of machine learning algorithm:** There can be several machine learning algorithms applicable to both business and data challenges.

**Train:** The algorithm must be trained to create the model. The training process could be supervised or unsupervised depending on the algorithm and data type. Reinforcement learning is a major factor.

**Evaluate:** The models must be evaluated to locate the proven best performing algorithm.

**Deploy:** Models created by machine learning algorithms can be deployed to both on-premises and cloud applications.

**Predict:** Post-deployment, predictions can be made based on new and incoming data.

**Assess prediction:** The validity of predictions gained from the analysis is then fed back into the machine learning cycle to help improve accuracy.

---

6. **Artificial Neural Networks**

An artificial neural network is an algorithm group employed for machine learning. A neural network is an interlinked assembly of simple units or nodes and other processing elements. The functionality of those elements loosely resembles the animal neuron. The network’s processing ability is stored in the weights or interunit connection strengths. These are obtained by the adaptation process from training pattern sets.
A typical human brain has approximately 100 billion neurons or nerve cells. These neurons communicate through “spikes” or momentary electrical signals in the membrane or cell wall voltage. Synapses, positioned on dendrites or cell branches, mediate interneuron connections. A single neuron is connected to thousands of other neurons, and thus constantly receives thousands of signals, all of them eventually reaching the body of the cell. Here, the signals are summed together or integrated in a specific manner. If the resulting signals exceed a certain threshold, then the concerned neuron will generate a responsive voltage impulse or ‘fire.” This is subsequently transmitted via axon or branching fiber to other neurons.

To determine whether a particular impulse should be generated (or not), a few incoming signals generate a kind of inhibitory effect which prevents firing, while others promote impulse generation and excitation. Each neuron’s unique processing ability is then believed to reside in the excitatory or inhibitory type the, as well as the strength of each unique neuron’s synaptic connections with the other neurons.

- **6.1 Layer of Neurons**

A neuron single-layer network $S$ is shown in figure 5. It is to be noted that each $R$ input is linked to each neuron and the weight matrix consists of $S$ rows.

*Figure 5: Layer of Neurons*
The layer is inclusive of the weight matrix, output vector $a$, summers, transfer function boxes, and the bias vector $b$.
Each input vector $p$ element is linked to each neuron via the weight matrix $W$.
Each neuron has a bias $b_i$, a summer, a transfer function $f$, and an output $a_i$.
Taken together, the outputs form the output vector $a$.
The weight matrix $W$ allows input vector elements to enter this network:

$$
W = \begin{bmatrix}
  w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\
  w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\
  \vdots & \vdots & \ddots & \vdots \\
  w_{S,1} & w_{S,2} & \cdots & w_{S,R}
\end{bmatrix}
$$

The row indices associated with matrix $W$ elements indicate the destination neuron linked with that weight. The column indices show the input source for that weight. It means the indices in $W_{3,2}$ proclaim that this weight is representative of the connection to the third neuron from the second source.

### 6.2 Problem Statement

Let us assume a dealer has a warehouse which stores different kinds of fruits. The fruit may get mixed together when they arrive at the warehouse. To separate the fruits, the dealer may require a sorting machine. The machine has a conveyer belt on which the items are loaded and these items pass through the conveyer belt through a set of sensors which measure three properties: texture, shape, and weight. The shape sensor may output 1 if the fruit is round, and if elliptical then -1. The texture sensor may output 1 if the surface is smooth, otherwise -1. The weight sensor may output 1 if item is one pound, and -1 if it is less.

These sensor data are sent to the neural network and the neural network decides which types of fruit are placed on the conveyer belt. Let us assume the fruits are apples and oranges placed on the conveyer belt.
When the fruit goes through the sensor, the value can be represented by a three-dimensional vector which represents shape, texture, and weight.

\[ \mathbf{P} = \begin{bmatrix} \text{shape} \\ \text{texture} \\ \text{weight} \end{bmatrix} \]

Therefore, an orange can be represented as

\[ \mathbf{P}_1 = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} \]

And apple can be represented as

\[ \mathbf{P}_2 = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} \]

It means a one three-dimensional input vector will be received by the neural network for every fruit found on the conveyer. A decision must be made to recognize whether the object is an apple \( \mathbf{P}_2 \) or an orange \( \mathbf{P}_1 \).
Deep learning (DL) is simply one of many available frameworks for deciphering machine learning problems. Deep models are termed “deep” as there are multiple computation layers. In the past, the important part of ML application consisted of putting in manually engineered methods of transforming a given data into a form suitable to shallow models. The key advantage of deep learning is the replacement of not only the shallow models at traditional learning pipelines, but also labor-intensive feature engineering. Deep learning also replaces a major chunk of domain-specific pre-processing. Deep learning has eliminated the boundaries that previously demarcated computer vision, medical informatics, natural language processing, and speech recognition. DL offers a unified toolset for managing diverse problems. The modules which are found in many deep networks are:

**Fully Connected Layer**: An inputs list can be transformed into an outputs list through a fully connected network. This transformation is termed fully connected as the input value influences the output value. Such layers will invariably have multiple parameters— even those applicable for relatively small inputs.

![Fully connected Layer](image)

**Convolutional Neural Network Layer**: A convolutional neural network (CNN) uses a variation of multilayer perceptrons. A perceptron is an algorithm for supervised learning of binary classifiers of possible outcomes. A CNN contains one or more than one convolutional layers. These layers can be either completely interlocked or pooled. Convolution is the integral quantifying how much two functions overlap as one passes over the other. Convolution is a way of mixing two functions by multiplying them. Before passing the result to the next layer, the convolutional layer uses a convolutional operation on the input.
Due to this convolutional operation, the network can be much deeper but with fewer parameters. Due to this ability, convolutional neural networks have very effective results in image and video recognition, as well as natural language processing systems.

**Fig 8: Convolutional Layer**

Figure 8 is an image of the first Convolutional layer, showing the volume of neurons within the layer and is denoted by the input volume shown in red (for example, a 32 x 32 x3, CIFAR-10 image). Each neuron inside this layer is connected not only to the input volume’s local region in a spatial manner, but also to full depth (all color channels).

**Recurrent Neural Network (RNN) Layers:** These layers are primitives permitting neural networks to learn from inputs sequences. The Recurrent Neural Network layer assumes that an input evolves from one sequence step to a succeeding sequence step following a certain defined update rule. This rule can be learned from data and presents a prediction of the next stage. It will be in the sequence copying all the previous states. An RNN can be regarded as multiple copies of the same network, each transferring a message to its successor.

**Fig 9: RNN Layers**
Such a chain-like nature reveals that the RNN is closely related to lists and sequences. They can be regarded as the neural network’s natural architecture. Indeed, in the last few years, we have witnessed incredible success in applying RNNs to a number of problems, including speech recognition, image captioning, translation, and language modeling.

A big limitation of Convolutional Networks (the popular term for Vanilla Neural Networks) is that the API is extremely constrained. A fixed-sized vector as input (for example, an image) is accepted and the fixed-sized vector is produced as output (for example, the probabilities of dissimilar classes). These models also perform this mapping by using a fixed number of computational steps (for example, the number of layers present in the model).

The principal reason for recurrent nets to be exciting is that they permit operations over sequences of vectors: there will be sequences in the input, the output, or in most cases both.

**Generative Adversarial Networks:** These are a new kind of deep network using two dueling neural networks. Generative Adversarial Networks (or GANs) consist of the generator and its adversary dueling each other. The generator attempts to extract samples from any distribution (e.g., generate realistic looking bird images) and the discriminator consequently works on the differentiating samples extracted from the true data samples. This can be a real image of a specific bird or one that is generator made. GAN’s “adversarial” training seems to generate higher fidelity image samples. These images are much better than those generated by other methods.

![GAN Diagram](image)

**Fig10: GAN**

GANs have proven able to generate extremely realistic images. They will in all probability power the upcoming of computer graphics tools generation. Samples of this system have now approached close to photorealism, even though many wrinkles have to be ironed out.
Deep Learning Frameworks: Software packages are being implemented by researchers to enable the construction of neural network (deep learning) architectures for decades. These systems were previously mostly special purpose, and only used within an academic group. This state has changed radically over the last few years. Google made the DistBelief system to construct and deploy many simple deep learning architectures. Along with Distbelief many simple packages such as Keras, Caffe, Torch, Theano and others are widely adopted by many industries. Google Tensorflow, based on Theano, in particular uses the principle of tensors for powering deep learning systems. Google Tensorflow provides flexibility for building sophisticated models that is not available in Caffe or Distbelief.

8. Hardware Considerations

In the early stages, computing processors were designed with only one core, which meant that they were able to perform only one operation at a time. The increased demand for computing capabilities to execute complex operations has evolved the processor technology to modern multiple 64-bit cores, capable of floating-point calculations with very high accuracy.

- 8.1 Parallelism speeds training

Most of the hardware at present uses parallelism to speed training. But we must determine which type of devices help in speeding training. The single instruction, multiple data computational model maps efficiently to vector processors, FPGAs, accelerators, and to custom chips. It has been seen for a dataset that the training performance is limited by memory performance and cache memory, and not on floating-point capability.

A traditional belief is that CPUs can deliver a higher training performance than GPUs, but accelerators reach high floating-point performance when they have concurrent threads in larger numbers for execution. In the instance of training with bigger datasets, only a small fraction of accelerator parallelism is utilized. In situations like these, better performance can be had on any many-core processor with a fast cache and a stacked memory subsystem, such as the Intel Xeon Phi processor. It is thus important to consider the amount of data to be available for training purposes at the time of hardware selection.

The Graphical Processing Unit (GPU) was historically a part of the traditional computer designed for gaming and graphic design but GPUs came to the rescue to solve the problem. They are capable of doing tons of simple calculations at a fast pace in parallel. GPUs, though expensive, are good at running neural networks, as they have thousands of cores that can do calculations simultaneously.
8.2 Calculation of Gradients

The calculation of gradients for orders of magnitude requires bandwidth and memory capacity to enjoy quicker time-to-model runtimes. A majority of highly effective optimization algorithms need a function to be evaluated, so as to calculate the neural network parameters’ gradient with respect to the objective function.

Gradient use offers an algorithmic speedup which can achieve considerably – even in terms of orders of magnitude – quicker time-to-model, and better solutions compared to gradient-free methods. The challenge arises when the gradient size becomes huge extremely fast as the ANN model’s parameter number rises. It means the memory capacity, along with bandwidth limitations (in addition to cache and the potential performance of the atomic instruction), dominate runtime of the gradient calculation.

It is vital to be aware that the hardware’s instruction memory capacity is adequately large to hold all machine instructions required to perform gradient calculation. It is to be noted that the gradient calculation code for even the most modest neural network models could be gargantuan.

9. Applications

Here are a few examples of how AI is applied today.

**Anomaly detection:** Anomaly detection refers to discovering patterns in information that don’t fit in with expected conduct. These non-adjusting examples are frequently alluded to as anomalies, outliers, discordant observations, exceptions, aberrations, surprises, peculiarities or contaminants in different application domains. Of these, anomalies and outliers are two terms utilized most regularly as a part of anomaly detection. Anomaly detection sees use in a wide assortment of applications. Fraud detection for credit cards, intrusion detection for cyber-security, fault detection in safety-critical systems, and surveillance for military activities.

**Augmented Reality:** AI is the heart of Augmented Reality (AR) platforms, empowering AR to interrelate with the physical environment in a multidimensional way. Eye tracking, object recognition and tracking, voice command recognition, and gestural input combine to let you operate 2D and 3D objects in virtual space with your hands, eyes, and words.

AI enables capabilities like real-world object tagging, enabling an AR system to predict the appropriate interface for a person in a given environment. Through these and other possibilities, AI enhances AR to create a multidimensional and responsive virtual experience that can bring people new levels of insight and creativity. The result is mixed reality in a single physical environment, seeing a sofa in your living room before you click. AI can illustrate and enable AR
platforms to act on real-time, real worlds environments. The result is more relevant and delivers more immersive experiences.

**Resource Scheduling:** The most common use of real-time AI is for dynamic scheduling of work orders through a sequence of work centers without human involvement. The prioritization of the task order is accomplished using a set of rules that take into account factors related to project completion and delivery.

**Digital Voice Assistant/Chatbot:** Chatbots in business support customer relationships. A digital assistant is used in many places such as marketing, e-commerce, customer support service, and IT automation. They can give answers to various questions, automatically take orders at a restaurant, and handle payments.

*Trademark. element14 is a trademark of AVNET. Other logos, product and/or company names may be trademarks of their respective owners.*

[Take the Quiz]