Untangling the Concept of Artificial Intelligence, Machine Learning, and Deep Learning

Muhammad Juliandri, Goce Ristanoski, and Uwe Aickelin

1 The Fame of AI, ML and DL

The words *Artificial Intelligence (AI)*, *Machine Learning (ML)*, and *Deep Learning (DL)* have been used extensively in recent years. They suddenly become the top computer science terms that appear in other research domains such as *AI for medical diagnosis*, *ML in finance*, or *DL for linguistics*. Even the mainstream media such as news articles now promote the benefit of exploiting them to face Industry 4.0.[1] As a result, there is an increasing number of organizations that start adopting these concepts. They believe that AI, ML and DL can help them to improve business processes in their organizations.

2 Why This Chapter Is Important

Despite their growing popularity, not many people are able to clearly understand the main ideas behind AI, ML and DL. Most of them find it a challenge to tell the difference between them and use those terms interchangeably. This common misconception needs to be rectified since they are different in several aspects. In this first chapter we walk through the concept of AI, ML, and DL by showing the definition

[1] https://www.forbes.com/sites/bernardmarr/2018/09/02/what-is-industry-4-0-heres-a-super-easy-explanation-for-anyone/#448f66849788

M. Juliandri · G. Ristanoski · U. Aickelin (✉)
School of Computing and Information Systems, Faculty of Engineering and Information Technology, The University of Melbourne, Melbourne, VIC, Australia
e-mail: gri@unimelb.edu.au; uwe.aickelin@unimelb.edu.au

M. Raz et al. (eds.), *Artificial Intelligence in Medicine*,
https://doi.org/10.1007/978-981-19-1223-8_1

and methodology involved for each of them. We also discuss *Data Mining* (DM) as a separate concept as well.

3 Artificial Intelligence

3.1 Definitions and Objectives

3.1.1 Understanding AI by Examples

The use of computing machines to assist humans in their tasks has been significantly evolving over the past few decades. Activities such as calculating numbers, storing documents, or reading articles can now be performed with a computer. However, none of these tasks can be categorized as a valid intelligent system because they still need a large portion of human instruction in reaching their end goals. Their computation is simply an execution of a quantified human intention, rather than a mimic of some form of human perception.

Contrary to such tasks, chatbots, machine translation, or autonomous vehicles are more fitted to be classified as applications of Artificial Intelligence. They require more than a simple function to be implemented on a machine in order to perform intellectual tasks – they require an amount of decision making conditioned by other inputs than human instructions. We can label them as systems that hold some intelligence [1].

3.1.2 The Basic Idea of AI

Provided with the above examples, we define Artificial Intelligence as a collection of processes in a system with the capability to learn and solve a problem like a human [2]. If a human needs a thinking process to solve a certain problem and a system could simulate the process into the expected solution, that means the system is leveraging Artificial Intelligence. However, this is not the only definition of Artificial Intelligence. There are in fact many different views on the definitions since the concept of Artificial Intelligence has shifted in accordance with the desired objectives.

3.1.3 AI Definitions in Literatures

Given the basic notion of Artificial Intelligence (AI), we observe other perspectives in literature. Russell and Norvig [3] highlighted four technical terms that briefly summarized multiple definitions of AI: *Acting Humanly*, *Thinking Humanly*, *Acting Rationally* and *Thinking Rationally*. They adopted the *Acting Rationally* as the essential characteristic of AI. There are two reasons for this: first is because it is more general compared to others, as in order to achieve intelligence, the machine

does not only rely on a rational thinking but also on actions taken accordingly to any conditions; second, it is more adaptable since the improvement of AI would not only revolve around human behaviour. The idea of making a machine smarter without explicitly giving it instructions is also supported by these two reasons. Hence, we consider this as the fundamental definition of AI.

3.1.4 Acting Rationally in Humans and Machines

Despite that definition, we still have another question left that is how to implement AI so the machine can act rationally. Humans can act rationally because they learn from their environment. The way they learn is by experiencing it or it is something given as knowledge. We know fire is hot because we may have touched it, or we have been taught about this as other people had touched it before. The result of learning process is pairs of actions and consequences that can be used as responses. By comparing consequences, we can choose the rational response in a particular condition to obtain an expected result. A machine needs a similar process to act rationally. It needs to digest information in various forms, extract knowledge, perform measuring consequences, and perform responses.

3.1.5 The Growing Domain Inside AI

Interestingly, to build the machine that can respond to incoming information would require knowledge from several domains. For instance, to make the machine have the capability of doing two-way communication with humans, it needs to incorporate a Natural Language Processing engine. To make the machine learn the pattern from the data, it needs Machine Learning techniques. And to make the machine distinguish objects, it needs Computer Vision proficiency. All of these have brought AI to a broader concept that covers all those domains. Thus, now we can see Natural Language Processing (NLP), Machine Learning (ML) or Computer Vision (CV) as some branches inside AI.

4 Branches of Artificial Intelligence

4.1 Natural Language Processing

4.1.1 Why Does NLP Require Human Intelligence?

Natural Language Processing (NLP) is AI's branch that handles textual and linguistic data. This type of data is quite challenging to address because it may come in an unstructured form. An example of this type of data is medical patient records – the examination performed by doctors involves the documentation process of the

patient's condition. By employing NLP this documentation can be conducted automatically by recording the conversation between the patients and doctor during examination [4]. It can save the time taken for each examination and the doctor can focus more on the patient. However, this conversation data may have different accents, incomplete sentences, or any variation of voices. Humans may easily understand this conversation and transform it into medical text, while it is a difficult task for a machine to achieve the same goal.

4.2 Computer Vision

4.2.1 Why Does CV Require Human Intelligence?

While NLP deals with textual data, Computer Vision (CV) is another branch of AI that treats digital images as their primary source of knowledge. This kind of input is also challenging to process since it does not have any recognizable features that a machine can learn from. To better illustrate this complexity, let us assume that we need to identify pneumonia from frontal chest radiograph images [4]. When a radiologist sees those images, they can directly interpret the possibility of pneumonia without much conscious cognitive effort. But for a machine, this picture is just a set of pixels represented as numbers. Each of the object's characteristics, like colours and shapes, need to be extracted from this numeric representation before the machine can detect if a patient has pneumonia or not.

4.3 Recommendation System

4.3.1 Why Does RS Require Human Intelligence?

We are often provided with some suggestion videos on streaming service platforms. It looks like they certainly know what kind of videos that are relevant to us. Different people will have different video recommendations, but more interestingly these suggestions happen without having us to search for videos manually. The reason for this is because they are adopting Recommendation System (RS) in their platforms. RS can be seen as an engine that provides a user with relevant items, products, or even services. In the context of streaming platforms, the items can be videos or songs. However, the discovery process of these relevant items is challenging as the engine needs to understand how relevant an item is for a user. In addition, this relevancy is measured on large collections of items where the manual approach will be infeasible.

5 Data Mining

5.1 Definitions and Objectives

5.1.1 Challenges in Extracting Knowledge from Data

Having defined the idea and branches of AI, we can clearly see that data plays a central role in achieving intelligence since it contains a lot of valuable knowledge. If the machine is able to obtain that knowledge, then it may be able to act like humans. However, finding the knowledge from data is a difficult task due to many factors. Occasionally the data produced may grow significantly over a period of time. This affects an exhaustive process of knowledge identification. In addition, data may also consist of many attributes that need to be considered. An advanced technique is required to perform the knowledge extraction. In such a case, we can incorporate Data Mining as a technique to handle that process.

5.1.2 Data Mining in Literature

Data Mining (DM) actually isn't a new concept, even though its popularity just accelerated recently. The term itself was coined in the 1990s by a database community [5]. However, this term is inaccurate as what we want to discover is not data, but useful information trapped in it. That is also the reason why many researchers formerly called Data Mining *Knowledge Discovery from Data* (KDD) [6]. Additionally, in KDD, the definition is presented in a wider context including data pre-processing, data transformation, data mining, and data evaluation.

5.1.3 Data Mining vs Machine Learning

Provided with the Data Mining definition, it will be also necessary to know the differences between Data Mining and Machine Learning. This is because Data Mining (DM) is commonly compared to Machine Learning (ML). Even though the clear line that separates them is now hardly to be seen, we are still able to find the distinctions. While DM explores knowledge from a pattern, ML is more concerned about the algorithm to extract that pattern [7]. As a result, DM uses a lot of ML techniques. Another significant difference between them is the objectives. In ML, the primary objectives involve making improvements to the machine based on the experience so it can predict the future well. DM on the other hand, is leveraged massively on finding patterns, looking for trends, or discovering outliers on the current data. Typical examples in medicine would be multi-objective semi-supervised clustering to identify health service patterns for injured patients [8].

5.2 Data

5.2.1 Understanding the Data Properties

Before diving into some techniques used for Data Mining, it is necessary to understand common terminology used to describe an essential part of data. Data is used as the input of a learning process in a machine and it can be divided into three different parts: features, labels, and instances.

5.2.2 Explanation of Features, Labels, and Instances

Features are part of data that represent characteristics or attributes in our observation, while labels are part of the data that we aim to learn or predict. One important thing to note is a label is actually a feature, but its value can be determined by other features. Hence, the feature and label pairs are also known as explanatory and response variables. Moreover, instances can be considered as individual parts of data known as examples or samples. To illustrate how they relate to each other, assume that we are given data consisting of genders and heights to predict grades of each person (Fig. 1). In that case, the features are genders and heights; the labels are grades; and an instance is each person's data.

We can separate the data based on their value into two major types: Categorical Data or Numerical Data.

5.3 Categorical Data

5.3.1 What Is Categorical Data?

When the information is presented in a way that describes a certain type, most often in a distinct and discreet format, and an order of the value may or may not be present, then we are referring to this type of data as categorical data. The values in Categorical Data are denoted as classes. Examples of Categorical Data are blood type, age group, education level, location (by name), food group etc.

Fig. 1 Instances, features, and labels in data

5.4 Numerical Data

5.4.1 What Is Numerical Data?

In contrast to Categorical Data, Numerical Data deals with a wide range of continuous numbers. It often involves a real value number such as weights, blood pressures, and ages. The important thing with Numerical Data is we can perform any mathematical operations with it, unlike the categorical data.

5.5 Methodology

So far, we have presented an overview of Data Mining and how it employs Machine Learning techniques to learn the pattern. For that reason, in this section we shall show two broad classes of Machine Learning techniques that are extensively used for discovering patterns in Data Mining: Classification and Clustering.

5.6 Classification

5.6.1 What Is Classification?

Classification is a strategy in supervised learning to find a relation between features and labels in the data. In supervised learning, data comes with labels. The most important thing with this strategy is it is only applicable for the labels of categorical data. Once the relation is extracted, then it can be used to predict labels of the unseen data. A number of applications use classification in detecting patterns such as spam detection, weather forecasting, and medical diagnosis.

5.7 Clustering

5.7.1 What Is Clustering?

Clustering is another strategy in Data Mining to find patterns from the data. It is part of Unsupervised Learning as the data does not come with labels. We can only compare one instance to another instance. If these two instances shared similar features, then we can group them into one cluster. Otherwise, that instance forms a separate cluster with other instances matching its characteristics. Some applications of clustering are topic modelling, anomaly detection, and image segmentation.

6 Machine Learning

6.1 Definitions and Objectives

6.1.1 Machine Learning's Definition

In the previous section, we see Machine Learning from the perspective of Data Mining without explaining its actual definition. In literature, we may find many definitions of Machine Learning introduced by *scientist*s. Despite some differences in definitions, most of them agree that Machine Learning in general can be thought of as a program that can learn from past information to predict future information. We put an emphasis on words *learning* [9] and *predicting* [10] since these two also represent essential processes in Machine Learning.

6.1.2 Common Workflow in Machine Learning

However, learning and predicting are not the only processes. It may be preceded with two additional processes, *Pre-processing* and *Feature Engineering* (Fig. 2). Pre-processing is part of transforming the raw data into a cleaned version and this step is usually adopted if we have textual data. An example of this is to clean the document from all html tags leaving only the relevant contents of interest. Following the pre-processing we may also perform Feature Engineering to extract attributes from data. We need this since the data may not come with identifiable attributes, so we need to extract them before it can be processed further. For instance, in textual data, we need to convert them into a numeric representation because some techniques in machine learning only accept numbers as their input.

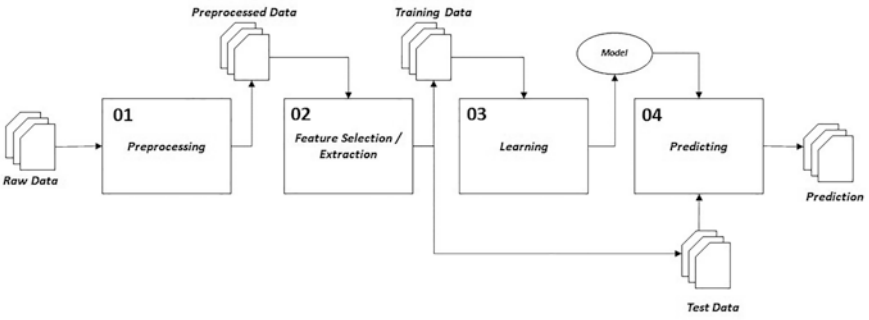

Fig. 2 Machine Learning's workflow

6.1.3 Main Factors Behind Machine Learning's Performance

Provided with the workflow, we can see that Machine Learning performance is highly dependent on two factors. The first factor is the input data. The larger the data means the machine can learn more, but it is also important to note that this would influence the running time. Therefore, we need to consider carefully if we want to use more data. The second factor is the learning technique to extract the knowledge (*model*). We should pick the appropriate learner based on the data types and goals that we want to achieve. Failing to pick the correct learner would also affect the performance. Hence, it is necessary to understand two big categories of learning: *Supervised Learning* and *Unsupervised Learning*.

6.2 Supervised Learning

6.2.1 Overview of Supervised Learning

In Supervised Learning, the model is generated by giving the labelled data to the learner. What we mean by labelled data is the data that comes in pairs, the input with its corresponding output [10]. The learning process can be seen as finding the best model that maps that input to the output. We can divide the Supervised Learning based on the types of output. We call it Classification if it handles the Categorical Data as the output. *Decision Tree*, *Support Vector Machine*, and *K-Nearest Neighbours* are examples of Classification techniques in Machine Learning. Alternatively, if the output is a continuous number, then we call it Regression. Some of the Regression techniques are *Linear Regression, Polynomial Regression,* and *Ridge Regression*.

6.3 Unsupervised Learning

6.3.1 Overview of Unsupervised Learning

While in Supervised Learning we have labelled data at our disposal, in Unsupervised Learning we may not have or need it. It is because sometimes we cannot define all the possible output or class in the data. We are still able to extract some knowledge from the data. It is performed by measuring the similarity among inputs. For instance, in the topic modelling task, we use the number of overlapping words as the similarity metrics. As a result, documents with a high number of overlapping words will form a cluster. In such a task, we call this technique Clustering since the main goal is to group the similar objects in a cluster. However, this is not the only technique covered in Unsupervised Learning. *Outlier Detection*, *Recommender System*, and *Association* are other techniques that fall within this category.

6.4 Feature Engineering

6.4.1 Why Feature Extraction?

Despite the fact that Feature Engineering is not necessarily to be performed in Machine Learning, for some cases we would not be able to continue to the next process without having it done. For example, in a topic Classification task, we have documents as the input. As a result, we need to convert this document to its numeric representation since the optimization algorithm is only able to handle numerical data. One way to meet this requirement is to extract features from documents such as the size of document, the number of special characters and the number of stop words. They can be the distinctive characters to differentiate one document to another. However, we may also use more advanced techniques of feature extraction such as *bag of words* and *term-frequency inverse document-frequency* (TF-IDF).

6.4.2 Why Feature Selection?

In addition to Feature Extraction, there is another Feature Engineering technique called Feature Selection. The aim of this process is to select the most important features in the data to obtain a better performance. It is because irrelevant features would make the solution not unique and the learning process slower. A further instance of this is having size of documents and number of characters as features. One of these features can be ignored since they are highly correlated to each other. We can obtain the size of documents based on the number of characters and vice versa. As a result, we can have a lower number of features involved in the learning process and the time to build the model can be reduced. Feature Selection can be executed with a variety of methods such as *Principal Component Analysis*, *Akaike Information Criterion (AIC)*, and *Singular Value Decomposition (SVD)*.

6.5 Learners

In this section, we explain the linear model for Regression and Classification. These two learners are quite important in Machine Learning since they form a foundation for more complex models [11].

6.6 *Linear Regression*

6.6.1 Example of Simple Linear Model

In high school, we learnt how to generate a linear equation that goes through two data points. This equation is usually expressed in the form of:

$$y = mx + c$$

where m is the slope, and c is the intercept. If we have two data points called A and B, with their coordinates: (1, 3) and (2, 5), then we can calculate the slope and intercept for a straight line that passes through A and B in following steps:

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{5-3}{2-1} = 2$$

$$c = y - mx = 5 - 2(2) = 1$$

Having the slope and intercept, we can plug them into the variables to produce the full equation.

$$y = 2x + 1$$

If a new point C comes with $x = 10$ then using that equation, we can predict the value of $y = 21$. That linear equation is known as the model of those data points. However, what if we have more than two data points and there is no straight line that runs through them? In such a case, we still need to find the closest line to be the model of that data. One approach to find that line is using Linear Regression technique.

6.6.2 The Idea Behind Linear Regression

Linear Regression finds the fittest line that represents the relationship between features x and labels y. In measuring the fittest line, Linear Regression employs a performance metric and a commonly used metric is sum of squared error. This sum of squared error is computed by comparing the actual label y and the predicted value $\hat{y}$ for each data. Hence, the smaller this value is the better the model generated.

6.6.3 Evaluating Fittest Line Using Sum of Squared Error

In Fig. 3, we present two different linear models for three data points: A, B and C. The first model is shown by the dashed line that runs through two data points (B and C) and the second model is depicted by the solid line which did not sweep any

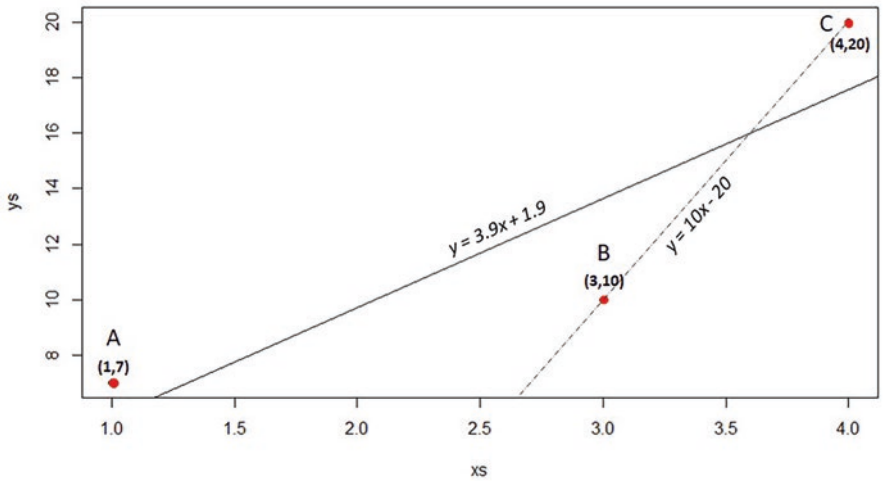

Fig. 3 Two different model for same data

points. At a glance, the dashed line looks like a fitter model for the data. However, the sum of squared error generated by the dashed line is much larger compared to the solid line. The sum of squared error for the first model is

$$\Sigma\left(\hat{y}-y\right)^2=\left(-10-7\right)^2+\left(10-10\right)^2+\left(20-20\right)^2=289$$

while the error for the second equation is

$$\Sigma\left(\hat{y}-y\right)^2=\left(5.8-7\right)^2+\left(13.6-10\right)^2+\left(17.5-20\right)^2=20.65$$

Hence, we prefer to use $y = 3.9x + 1.9$ as the model to precisely describe the data. However, in a real case, the linear models for three data points above can be infinite in number. Therefore, we need a shortcut to find the most suitable line and it can be done by minimising the sum of squared error. This shortcut is known as *the method of least square*.

6.6.4 Linear Regression for More Complex Model

The Linear Regression model is not only able to handle a single variable problem like above, but far beyond that as well. They can also address multidimensional data. In fact, they can also be used to generate more complex models like polynomial functions to fit a non-linear data.

6.7 *Logistic Regression*

6.7.1 Issues with Linear Regression

Often, we are faced with a problem that has only two different possible outcomes. For instance, to model if a person had a farsightedness issue based on his age, we can define this as a linear regression model. But it would not fit well at least for two reasons. First, Linear Regression may produce output outside of the expected values. Note that it can even produce a negative value as the outcome. Second, the larger the value of a feature may greatly influence the error for the model. In fact, for a binary classification problem, this cannot become the issue otherwise the generated model would be susceptible to outliers.

6.7.2 The Idea Behind Logistic Regression

Instead of using a Linear Regression as the learner, we need to use binary classifiers for such a case and Logistic Regression is one of them. Logistic Regression uses a logistic function to fit the non-linear model. It has the characteristic to map all real numbers into numbers between 0 and 1. This is the main factor why this function is suitable for Classification. In terms of appearance, the logistic function has an s-shaped line (Fig. 4). The formula of this logistic function is as follow:

$$P\left(y = 1 | X = x\right) = \frac{e^{mx+c}}{1 + e^{mx+c}}$$

To make it as a linear model, it is usually transformed into the logit (*inverse of logistic function*) where it adopts the concept of probability and odds. Probability can be thought of as how likely an event occurred, denoted as p, while the odds can be defined as the ratio between the probability of the event occurring and the probability of it not occuring:

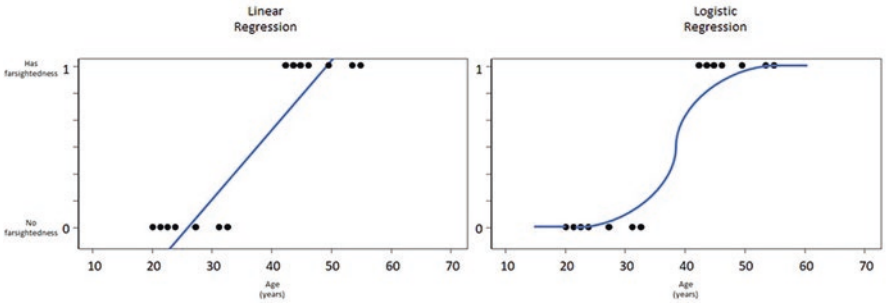

Fig. 4 Linear Regression and Logistic Regression Function

$$odds = \frac{p}{1-p}$$

Having the odds defined, the previous equation then can be converted into a new expression:

$$\log\left(odds\right) = mx + c$$

6.7.3 Evaluating Fittest Model Using MLE

To generate the most fitted model for the data, unlike Linear Regression which uses a *least square method*, Logistic Regression employs a *Maximum Likelihood Estimator* (MLE). The idea behind MLE is to find parameters that will increase the likelihood of the outcome most.

6.7.4 Logistic Regression for More Complex Model

It should be noted that all the equations shown above are only applicable for a single feature model. If we have more than one feature, then it should be modified accordingly. For example, if there are two features, then $mx + c$ should be replaced by $m_1x_1 + m_2x_2 + c$ since different features will have their own slope (*parameter*).

7 Deep Learning

7.1 Definitions and Objectives

7.1.1 The State-of-the-Art Classifiers in Machine Learning

Deep Neural Networks or Deep Learning has become one of the popular methods in Machine Learning to address difficult tasks ranging from Computer Vision to Pattern Recognition [12]. In Computer Vision domain, Deep Learning can assist systems to recognize images of handwritten digits and classify them to correct labels automatically [13]. This capability was not limited to digits, but also to other objects in images such as planes, cars, or animals [14]. Having the knowledge that makes Deep Learning a good classifier, many works propose this method as their core algorithm to perform more complicated tasks in related fields such as robots [15] and self-driving cars [16].

7.1.2 Understanding the Idea Core Behind Deep Learning

Provided with many powerful applications above, we have important questions to answer. *What is Deep Learning and why it is so powerful?* Before we jump into Deep Learning, we need to know what Perceptron is. Perceptron is a binary classifier that uses the concept of neuron. A neuron in our brain is a unit that processes some signal and transmits it to other neurons. This process will influence how strong the output signal will be. For instance, humans can hear sounds and see images because of this processed signal. Perceptron adopts this concept by introducing a single neuron to process input and produce an output in Classification tasks. This is also the main reason why Perceptron is also called an Artificial Neural Network.

In a Perceptron, the process of transforming some information into another information can be seen as a certain mathematical operation. This operation is called an Activation Function (Fig. 5). Recall that in Logistic Regression such mathematical operation is a logistic function. In other words, Perceptron is similar to Logistic Regression if we use Logistic Function as the Activation Function. The Perceptron model however, is not quite robust, primarily if the input is not linearly separable. Put it simply, there is no straight line or plane that can divide the data into two different classes. In addition, Perceptron is only applicable for binary classification. Those are the main factors behind the invention of Multilayer Perceptron.

7.1.3 What Is Multilayer Perceptron?

In Multilayer Perceptron, we may have more than a single hidden layer and for each hidden layer we also can have more than a single neuron (Fig. 6). For this reason, Multilayer Perceptron is also known as Deep Neural Network (Deep Learning). By adopting more neurons and layers, it can solve the non-linearly separable issues and build more complex functions. It can be thought of as a composite function in mathematical theorem, where we apply another function to the result of the previous function. As a result, this would transform our data into a higher dimensional representation. Imagine this as trying to separate data points into two classes, either by using a straight line or curved line. Certainly, the accuracy would be better if we use

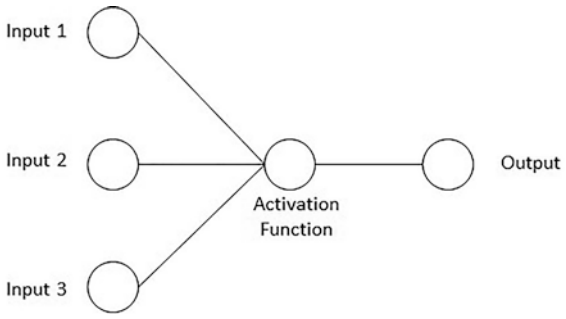

Fig. 5 Perceptron model

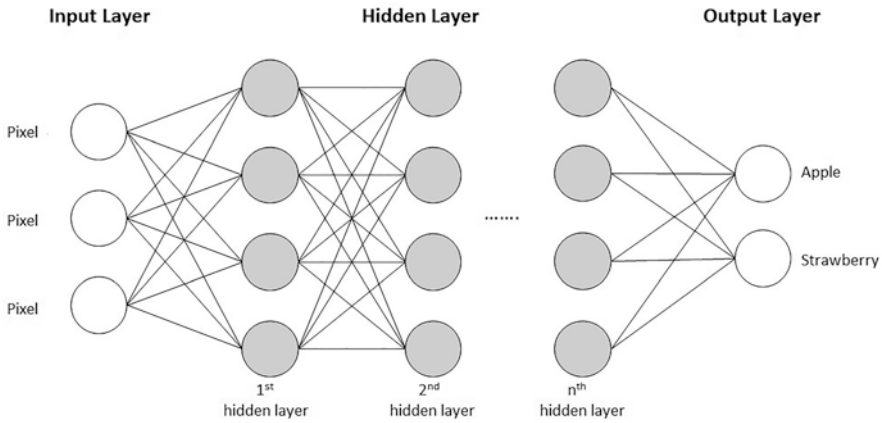

Fig. 6 Deep learning model

the curved line instead of straight line for a complex problem. This answers our question why Deep Learning is powerful to approximate all Classification tasks.

7.2 Variants

There are several variants of Deep Learning that are very popular nowadays: *Feedforward Neural Network*, *Convolutional Neural Network*, and *Recurrent Neural Network*. The variants differ in the way they process the input.

7.3 Feedforward

Feedforward Neural Network is the most common neural network used in many applications. It is a standard Multilayer Perceptron with a fully connected neuron (Fig. 7). Feedforward Neural Networks learn by randomizing the initial value of parameters. After that, it will pass this value as the input for the neuron in the next layer. The way it passes the value to the next layer is the main reason why it is called Feedforward. If the prediction is different with the actual output, then the parameter value may need to be reduced or increased. Otherwise, no penalty will be given to the current value of parameters.

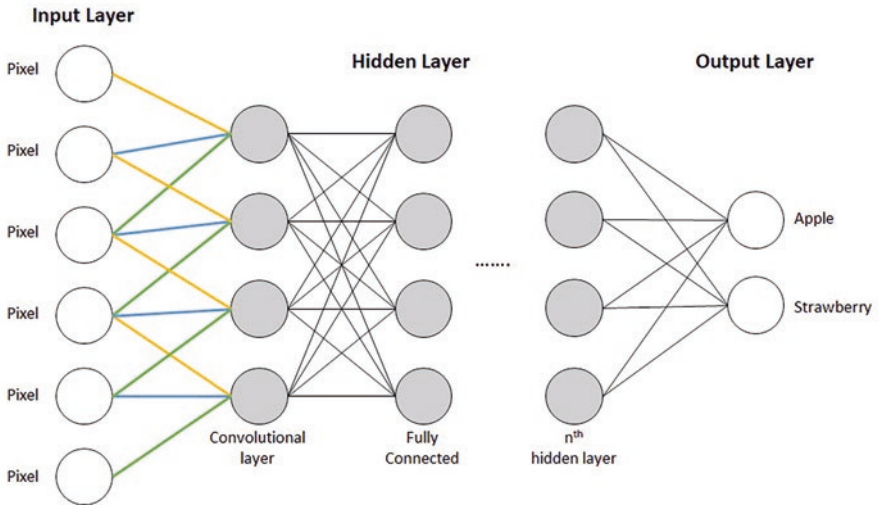

Fig. 7 Example of convolutional layer

7.4 Convolutional

In contrast to fully connected neural networks, Convolutional Neural Networks have some layer in their architecture, where its neuron is computed only by a specific number of neurons in the previous layers (Fig. 7). This can be realized by using a sliding window and filter. The filter aims to extract some pattern in the images. This approach is believed to generate a more accurate prediction primarily in the visual recognition domain. It is because an object is constructed by multiple smaller objects which is only influenced by the pixels around the neighbourhoods. A further instance of this is in handwritten digits classification. Using a filter size of 3×3 means that it only considers the pixels in the region of 3×3 and as a result the filter will detect edges for each number as a pattern. For example, digit '4' can be seen as a construction of 3 pattern: left vertical edge, middle horizontal edge and right vertical edge (Fig. 8).

7.5 Recurrent

Occasional sequence of the data contains some knowledge that can be extracted to predict the output more precisely. A further instance of this is in the language generation model where we are provided with a sequence of words as the input and need to predict the next word. In such a case, the order of words in the input will influence the probability of the next word. If we know the input is "Recurrent Neural" then we can easily predict that the subsequent word is "Networks".

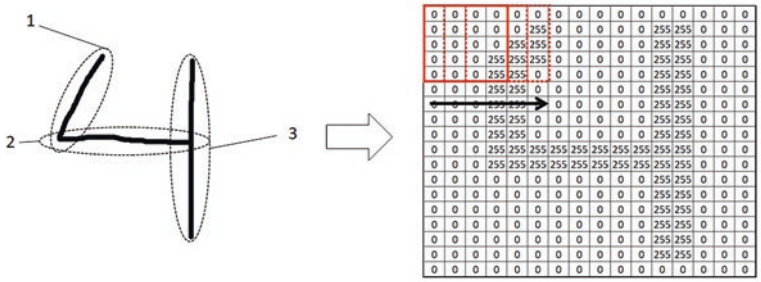

Fig. 8 Illustration of handwritten digit construction

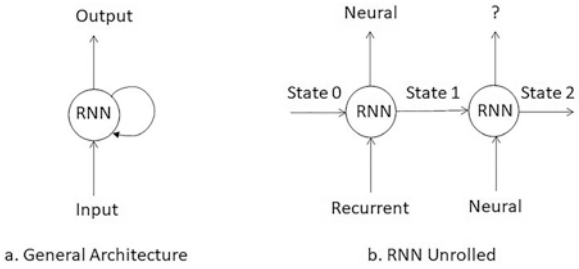

Fig. 9 Recurrent neural network architecture

Recurrent Neural Networks exploit this idea in its architecture. Instead of processing the input in parallel, it operates sequentially. The reason for this is because they need some information from the previous input to be processed along with the next input. This information is called a state vector with purpose to maintain the context. In terms of architecture, Recurrent Neural Network is similar with Feedforward Neural Network, but the neuron will have another edge that points to its own (Fig. 9).

References

1. Nilsson NJ. Principles of artificial intelligence. Illustrated, reprint. The University of Virginia: Morgan Kaufmann Publishers; 1986
2. Akerkar R (2019) Artificial intelligence for business, 1st edn. Springer
3. Russell S, Norvig P (2010) Artificial intelligence: a modern approach, 3rd edn. Prentice Hall, Upper Saddle River
4. Erik JT (2019) High-performance medicine: the convergence of human and artificial intelligence. Nat Med 25:44–56
5. History of Data Mining [Internet]. [cited 2020 Aug 11] Available from https://www.kdnuggets.com/2016/06/rayli-history-data-mining.html
6. Han J, Pei J, Kamber M (2011) Data mining: concepts and techniques, 3rd edn. Elsevier
7. Tan PN, Steinbach M, Kartpane A, Kumar V (2019) Introduction to data mining, 2nd edn. Pearson, New York

8. Khorshid HA, Aickelin U, Haffari G, Hassani-Mahmooei B (2001) Multi-objective semi-supervised clustering to identify health service patterns for injured patients. Health Inf Sci Syst 7(1):18
9. Mohri M, Rostamizadeh A, Talwalkar A (2018) Foundation of machine learning, 2nd edn. MIT Press
10. Murphy KP (2012) Machine learning: a probabilistic perspective. MIT Press, London
11. Bishop CM (2006) Pattern recognition and machine learning. Springer, Cambridge
12. Deng L, Yu D (2014) Deep learning: methods and applications. In: Foundations and trends in signal processing, pp 197–387. https://doi.org/10.1561/2000000039
13. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. In: Proceedings of the IEEE, pp 309–318. Retrieved from http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf
14. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436. Retrieved from https://link-gale-com.ezp.lib.unimelb.edu.au/apps/doc/A415563174/
15. Lee J, Ryoo MS (2017) Learning robot activities from first-person human videos using convolutional future regression. In: The IEEE conference on computer vision and pattern recognition (CVPR) workshops, pp 1–2. Retrieved from http://openaccess.thecvf.com/content_cvpr_2017_workshops/w5/html/Lee_Learning_Robot_Activities_CVPR_2017_paper.html
16. Kim J, Canny J (2017) Interpretable learning for self-driving cars by visualizing causal attention. In: The IEEE international conference on computer vision (ICCV), pp 2942–2950. Retrieved from http://openaccess.thecvf.com/content_iccv_2017/html/Kim_Interpretable_Learning_for_ICCV_2017_paper.html