

Comparative Analysis of Machine Learning and Deep Learning in Practical Scenarios

Guoxiang Ran*

Department of Electrical and Computer Engineering, Southeast University Chengxian College,
Nanjing, China

*Corresponding author: 219121215@cxxy.seu.edu.cn

Abstract. With the emergence of ChatGPT, machine learning (ML) and deep learning (DL) technology have gradually attracted people's attention, and these two technologies have already penetrated People's Daily life unconsciously. However, the application of these two technologies in various industries is still in the development stage, and choosing an inappropriate technology in different scenarios will lead to problems such as a waste of resources. The purpose of this article is to introduce the basic concepts, principles, and common algorithms of these two technologies, and by comparing the results of different technologies in the same scene, this article will compare and analyze the advantages and disadvantages of ML and DL in feature generation, model complexity, data scale, etc. Finally, the different application scenarios of the two technologies are obtained. ML is better for situations where there is less data and the problem is relatively simple, while DL performs better for tasks that deal with large and complex data and require hierarchical feature representation. This article hopes to inspire problem-solving in this subject area.

Keywords: Machine Learning; Deep Learning; Decision tree; Backpropagation.

1. Introduction

Currently, machine learning (ML) and deep learning (DL) play crucial roles in modern technology and business applications. They enable computers to learn patterns based on data and automatically adjust to make decisions or predictions. For instance, the field of medicine and healthcare has rapidly advanced and changed. The applications of data-driven, reliable, and effective ML to deep learning technologies have pushed and shaped the evolution [1]. With the development of the internet, a vast amount of data is generated, making ML and DL critical tools for processing this data. In computer vision, DL has achieved significant success in image and video processing, including AI drawing, image enhancement, and the automatic generation of short videos with dubbed subtitles [2].

ML and DL have also stimulated the flourishing development of natural language processing and speech recognition technologies. The appearance of ChatGPT, which can interact based on the context of the conversation, engaging in chat similar to a human [3], ushered in a new era, bringing tremendous convenience to people's daily lives. It is a milestone in the history of technological development. The emergence of ChatGPT immediately brought concepts like artificial intelligence, ML, and deep learning into people's daily lives. In simple terms, artificial intelligence is a broad concept, ML is one of its technical means, and DL is a specific implementation of ML.

However, current ML and DL still have shortcomings. They require large amounts of labeled data, and the performance of the final model heavily depends on the quality of the data. Insufficient or biased data can lead to a decrease in the model's capabilities. Additionally, deep learning demands significant computational power, increasing training and deployment costs. Therefore, choosing the wrong technology for the wrong use case can end up doing more with less. To solve this problem, it is important to know when to use ML and when to use DL.

This article will focus on introducing traditional ML and deep learning. Starting with their historical development, the basic logic, and representative algorithms (the decision tree algorithm in ML and the BP algorithm in DL in detail). The article will analyze the implementation principles of each algorithm, evaluate their strengths and weaknesses, and ultimately derive the relevant applicable scenarios for ML and DL.

2. ML

2.1. Overview

Probability theory, statistics, approximation theory, convex analysis, algorithm complexity theory, and many other fields are all involved in ML, which is a multidisciplinary field. Focused on the study of how computers mimic or replicate human learning behavior to pick up new abilities or information and rearrange the current knowledge structure to make it function better all the time. It is the essential component of artificial intelligence and how computers can be made intelligent.

2.2. Historical Development

There have been decades or perhaps centuries of ML. The methods and bases of widely used ML date back to the 17th century and include Bayes, Laplace's derivation of least squares, and Markov chains [4, 5].

Over the decades, researchers have published a wide variety of ML methods, which can be classified according to different aspects of emphasis. Classification based on learning strategy, learning method, learning style, etc. According to the most common classification based on learning styles, ML is mainly divided into supervised learning, unsupervised learning, and reinforcement learning. This article will emphatically introduce supervised learning, which is the largest family of ML algorithms. Figure 1 shows the classic supervised learning algorithms (DL is not included).

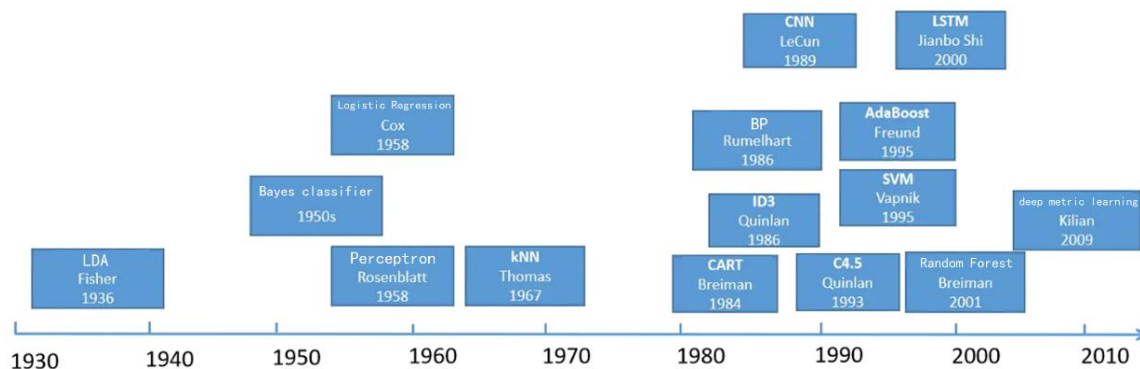


Fig. 1 The classic supervised learning algorithms (Picture credit: Original).

One of the earliest examples is Linear Discriminant Analysis (LDA), invented by Fisher and dating back to 1936 when there was no concept of ML. The perceptron model, which appeared in 1958, is a linear classifier and can be regarded as the predecessor of the artificial neural network, but it is too simple, so it has no practical value, and plays the role of ideological enlightenment, laying the ideological foundation for the following algorithms. These ML methods were dispersed and unorganized prior to 1980. They did, however, play an important part in the advancement of ML overall. Since 1980, ML has emerged as a distinct field. After this, a large number of ML algorithms were proposed and developed rapidly.

2.3. Decision Tree

The decision tree algorithm is typical in ML algorithms. Decision trees are widely utilized in many fields because they are simple to use, clear-cut, and resilient even when there are missing values. It is now commonly used in the following areas: Variable selection; Assessing the relative importance of variables; Handling of missing values; Prediction, etc. The nodes and branches are the main components of a decision tree model.

Nodes come in three varieties. First, the option that divides the records into two or more subsets that are mutually exclusive is represented by the root node. Second, internal nodes, similar to the root node, represent one of the possible choices. The last one, leaf nodes, each of them represents a final result of a combination of decisions (root node and internal nodes) [6]. The branches that connect the

nodes form paths. Each path represents a classification decision rule called the ‘if-then’ rule. For example, “if condition A and condition B, then result in C”.

A straightforward decision tree model with two continuous variables, x_1 and x_2 , that range from 0 to 1, and a single binary target variable, Y (0 or 1) is depicted in Figure 2 [6].

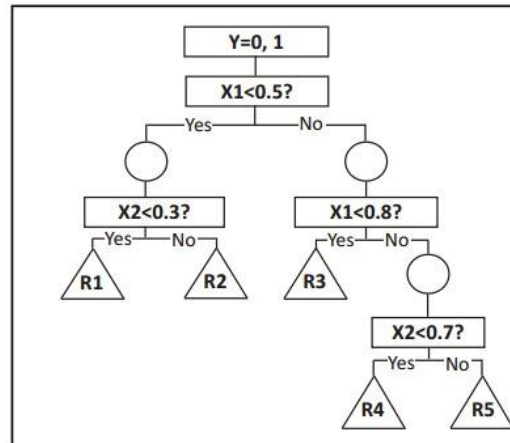


Fig. 2 Sample decision tree based on binary target variable Y [6].

In this figure, the node “ $Y=0,1$ ” is the root node, nodes like “ $X_1<0.5?$ ”, “ $X_2<0.3?$ ” are internal nodes, and R_1 means “if $Y=0,1$ and $X_1<0.5$ and $X_2<0.3$, then R_1 occurs”.

3. DL

3.1. Overview

As it is mentioned in the introduction, DL is a new research direction in ML. Its appearance makes ML gets closer to its first and final goal, Artificial Intelligence (AI). DL is to discover the inherent characteristics of the sample data. The knowledge acquired throughout these learning processes is very beneficial for interpreting textual information, including words, images, and sounds. Eventually, it allows machines to learn and analyze data just the way people do, and to identify information like text, images, and sounds [7]. DL is a complicated ML algorithm that achieves far better results than previous techniques.

3.2. Historical Development

The first mathematical model of neurons was proposed in 1943 by neurophysiologist Warren McCulloch and young mathematician Walter Pitts, and it was called the MP model after them [8]. It roughly mimics how human neurons work, but requires manual weight setting, which is very inconvenient. Even so, the MP model is a trailblazer and serves as the foundation for further studies.

In 1958, a neurobiologist at Cornell named Frank Rosenblatt added the learning function based on the MP model and proposed a single-layer perceptron model, which can select weights more automatically and sensibly (see Figure 3). The model can help classify a single-valued set of inputs into one of two classes [8]. However, the Perceptron can only solve the linearly separable problem.

It was not until 1986 that Rumelhart et al. published an article in Nature, proposing a multi-layer feedforward network trained according to the error reverse propagation algorithm called Back Propagation Network (BP network). It solves some problems that can not be solved by single-layer perceptron. Subsequently, the Boltzmann machine, convolutional neural network, recurrent neural network, and other neural network structural models have been well developed in this period.

In 2006, Hinton and his team published a breakthrough article on the concept of neural networks in Science, which first proposed the concept of DL and pointed out that the training problems of deep neural networks can be solved layer by layer initialization [9]. Professor Hinton solved the problem

of gradient disappearance of the BP neural network algorithm, and the idea of DL returned to the public's vision again, because of this, 2006 is known as the first year of the development of DL.

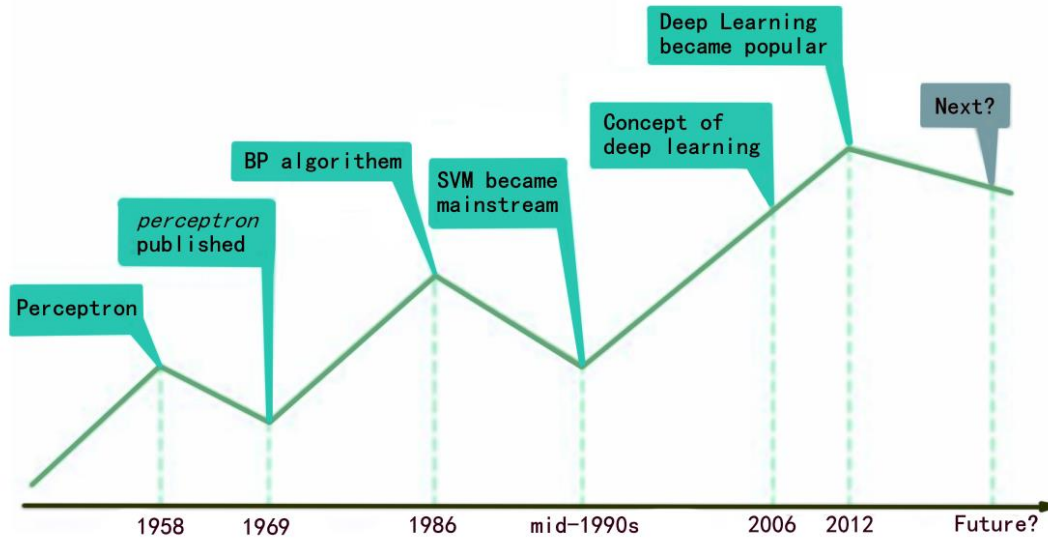


Fig. 3 History of neural network development (Picture credit: Original).

3.3. Backpropagation

Artificial neural networks are trained via the backpropagation algorithm in combination with optimization techniques like gradient descent. The goal of this algorithm is to adjust the weights and biases within the network to make the output closer to the expected result. It determines the gradient of the loss function concerning every network weight. To update the weights and minimize the loss function, this gradient is then fed back into the optimization process.

The BP algorithm is often used to train Multilayer Feedforward Neural Networks. This neural network consists of an input layer, a hidden layer (there can be more than one), and an output layer. Figure 4 depicts a three-layer neural network with two inputs and one output to demonstrate this procedure.

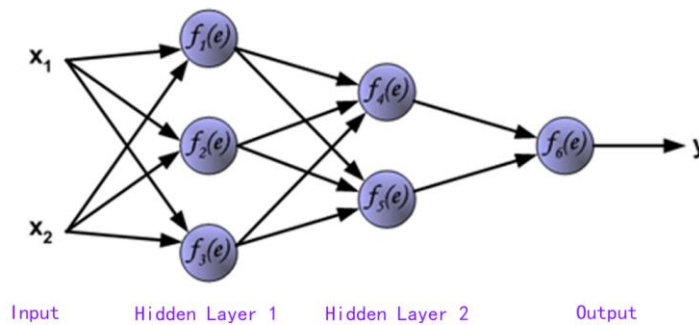


Fig. 4 Neural network illustration [10].

In general, the BP algorithm is divided into four steps:

Step 1. Forward propagation:

The input signals (x_1 and x_2) assigned to the associated target (desired output) z (not shown in the illustration) make up the training data set, which is necessary to train the neural network. The method is iterative. Using fresh information from the training data set, the weight coefficients of nodes are adjusted during each iteration. The method listed below is used to calculate modification: Forcing both input signals from the training set is the first step in every teaching phase. It may then ascertain the output signal values for every neuron in every network layer. This is the cause of the appearance of $f_1(e)$, as seen in Figure 5. F stands for the neuron activation function in this instance. The output

signal of the adder is signal e , and the nonlinear element's output signal is $y = f(e)$. A neuron's output signal is also represented as signal y .

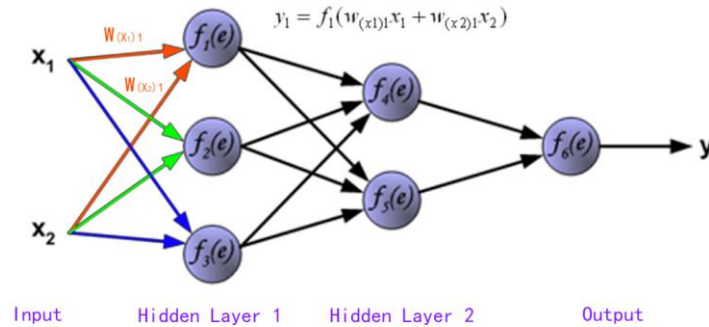


Fig. 5 Forward propagation [10].

Transmission of impulses via every layer that is veiled. Figures 6 and 7 illustrate the weighted connections between neuron m 's output and neuron n 's input in the subsequent layer, denoted by the symbols w_{mn} .

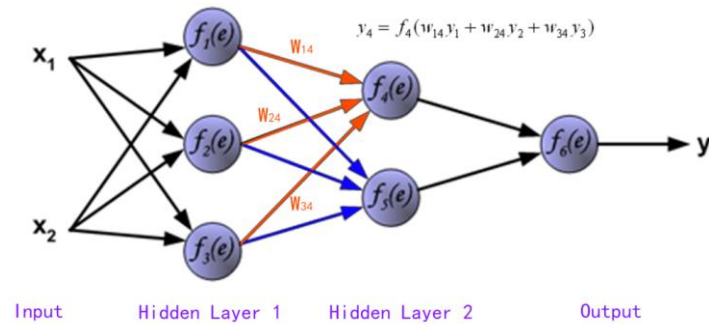


Fig. 6 Forward propagation in the next layer [10].

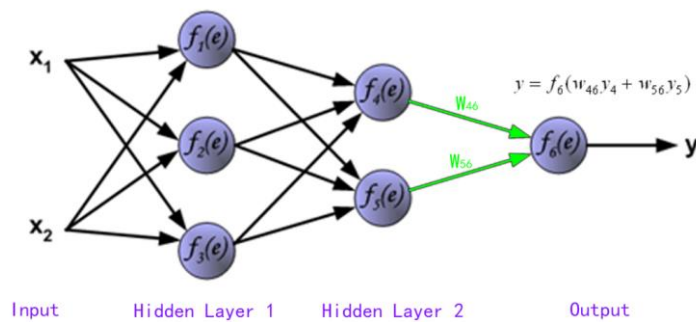


Fig. 7 Forward propagation in the last layer [10].

Step 2. Calculating error:

The output signal of the network y is compared with the goal, or desired output value z , which is present in the training data set, in the following algorithmic step, as illustrated in Figure 8. The output layer neuron's error signal, denoted by δ , is the difference.

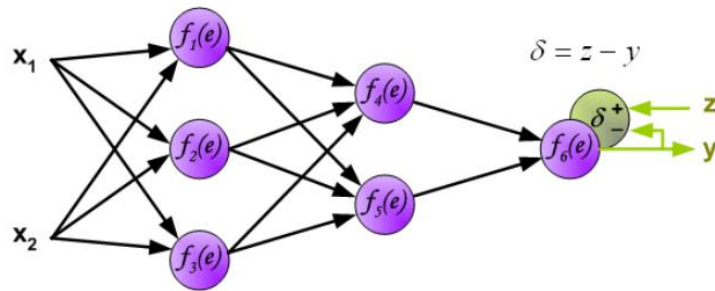


Fig. 8 Calculating error [10].

Step 3. Backpropagation:

Since the output values of internal neurons are unknown, it is not possible to directly compute error signals for these neurons. As it is shown in Figure 9. The method involves returning the error signal (δ), which is calculated in a single teaching step, to every neuron whose input signals were used to construct the neuron under discussion.

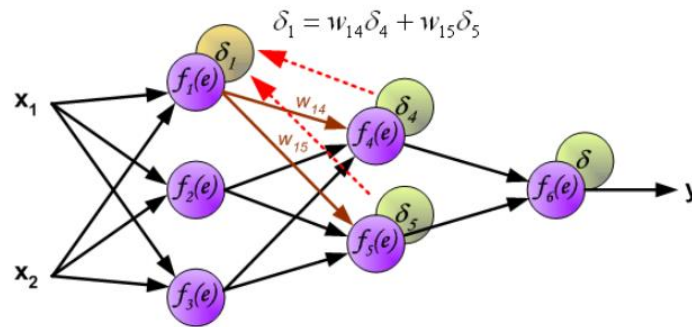


Fig. 9 Backpropagation [10].

Step 4. Updating weight:

The weight coefficients of each neuron input node may change once the error signal for each neuron is calculated. The derivative of the neuron activation function (whose weights are adjusted) is represented by the formulas below, where $df(e)/de$. As it is shown in Figure 10.

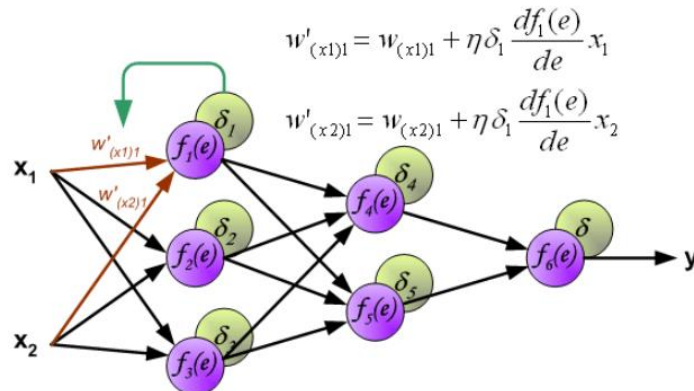


Fig. 10 Updating weight [10].

After an iteration similar to the first step, a new output value y' is generated, and the second and third steps are repeated until the network reaches satisfactory performance or the number of training times reaches the set upper limit [10].

4. Difference Between ML and DL

The above two specific ML and DL algorithms are put into the same field to show the difference between the two. The data presented in Table 1 is the performance of the two algorithms in the medical domain [11].

Table 1. Heart Diseases Prediction Accuracy of Existing Techniques [11].

Scientists	Year	Technique Used	Prediction Results	Dataset Used
Ritika Chadha et. al.	2016	Implemented heart disease prediction system using ANN, Decision Tree, and Naive Bayes algorithm with only eight attributes out of fourteen	ANN has given near about 100% accuracy whereas Decision Tree has given 88.025%, and Naïve Bayes has given 85.86% prediction accuracy.	Cleveland Heart Disease Dataset from UCI ML Repository
Chaithra et. al.	2018	The author used J48 Decision Tree, Naive Bayes and Artificial Neural Network algorithms for the classification of heart disease.	ANN performed better in predicting accuracy with 97.91% and 97.20% of sensitivity whereas J48 Decision Tree and Naïve Bayes have 92.55% and 74.40% accuracy respectively.	336 records with 24 attributes were obtained from the Echocardiography heart disease database
Poornima Singh et. al.	2018	Multilayer Perceptron Neural Network (MLP) with Back Propagation model for classes predictions	Multilayer Perceptron Neural Network with Back Propagation model has given ~100% of prediction accuracy	Cleveland Heart Disease Dataset from UCI ML Repository
Archana Singh et al.	2020	KNN, Decision Tree, such as support vector machine (SVM), and Linear Regression algorithms for heart disease prediction	Prediction accuracy of KNN=87%, SVM=83%, Decision Tree=79% and Linear Regression=87%	Cleveland Heart Disease Dataset from UCI ML Repository

From the data in the table, it can be seen that DL is superior to ML in heart disease prediction.

In the past, "feature engineering" was the term used to describe the process of designing features characterizing samples by human experts when ML was applied to real-world tasks. The generalization performance is greatly influenced by the quality of the features, and creating high-quality features is difficult. By generating high-quality features using the ML algorithm itself, feature learning advances ML toward "fully automated data analysis." This may seem like ML is not as powerful as DL, but on the other hand, life is filled with a lot of data unlabeled and a variety of complex environments, in health care need to predict disease, in the financial field need to use model learning for reputation assessment and fraud detection, in marketing, users' historical data is used for analysis, classification, personalized recommendation and so on. Under the combination of different conditions and environments, it is impossible to prove the superiority and inferiority of ML and DL. Although traditional ML requires a lot of manpower for data labeling, it does not require huge data and computing power to support itself, and simple models are easy to understand and explain when data is limited. It is often less expensive to get more optimistic results than DL. For example, in the

analysis of personal Internet preferences, the data generated by individuals is limited, and the need to generate a more accurate user profile is the opportunity for traditional ML.

Although DL is a ‘subset’ of ML, there are many clear differences between these two. Figure 11 explains the main difference between the two directly.

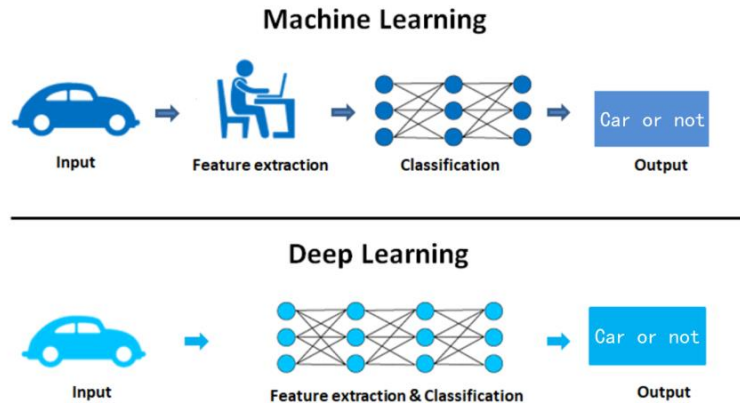


Fig. 11 The difference between ML and DL (Picture credit: Original).

Table 2 shows the differences in each aspect.

Table 2. The difference between ML and DL (Table credit: Original).

Aspect	ML	DL
Definition	ML is a branch of artificial intelligence (AI) that strives to enable computer systems to learn and improve performance automatically, without having to be explicitly programmed, through the use of data and statistical techniques.	DL is a special form of ML that is based on artificial neural networks (ANNs), which often contain multiple layers (deep layers) to learn more complex features and representations.
Feature generating	Traditional ML algorithms often rely on hand-designed features, which require the knowledge of domain experts.	DL algorithms can automatically learn hierarchical feature representations without the need to manually extract features, which makes them more adaptable to large-scale and high-dimensional data.
Model complexity	Traditional ML algorithms, SVMs, decision trees, etc., often have relatively simple model structures.	DL models, especially deep neural networks, have complex structures, made up of many layers of neurons.
Data scale	ML algorithms often perform well with manually selected features and relatively small amounts of data.	DL typically requires large amounts of labeled data to train complex neural networks and performs better on large data sets.

5. Conclusion

With the continuous research of ML and DL technologies and the increasing application of these two technologies in practical scenarios, the selection of appropriate technologies has become a matter of concern. To clarify the choice of technology for different scenarios, this article starts from the history of technology development to the introduction of specific algorithms (the decision tree algorithm in ML and the BP algorithm in DL in detail), and then further compares the specific algorithms, and finally draws a conclusion: ML is better for situations where there is fewer data and

the problem is relatively simple, while DL performs better for tasks that deal with large and complex data and require hierarchical feature representation. In practical applications, the appropriate methods are usually selected or combined according to the specific problems to give full play to their respective advantages. Thus, the importance of the right choice is far greater than the technology itself.

Both ML and DL are still in the development stage, there are still many key scientific problems to be solved, and the development of technology will never be possible without the efforts of all researchers. This article hopes to help the public better understand ML and DL by introducing and comparing these two technologies, thus providing more insights for them.

References

- [1] Jia L, Yaochu J. A comprehensive survey of robust DL in computer vision. *Journal of Automation and Intelligence*, 2023, 2(4): 175–195.
- [2] Li W, Xinya C, Chung W, Lingna X, Rustam, Yan L. Chatgpt's capabilities in providing feedback on undergraduate students' argumentation: A case study. *Thinking Skills and Creativity*, 2024, 51:101440.
- [3] Chiranjib C, Manojit B, Soumen P, Sangsoo L. From ML to DL: Advances of the recent data-driven paradigm shift in medicine and healthcare. *Current Research in Biotechnology*, 2024, 7: 100164.
- [4] Clement F, Wojciech P. Non-stationary data segmentation with hidden evidential semi-markov chains. *International Journal of Approximate Reasoning*, 2023, 162:109025.
- [5] Neal B G, Reaha G, Jose M A, Sergey K. Extended least squares (els) and generalized least squares (gls) for clutter suppression in hyperspectral images: A theoretical description. *Chemometrics and Intelligent Laboratory Systems*, 2024, 244: 105032.
- [6] Song YY, Lu Y. Decision tree methods: applications for classification and prediction. *Shanghai Arch Psychiatry*. 2015, 27(2): 130-5.
- [7] Xianchang C. Research on DL algorithm and application based on convolutional neural network. Zhejiang: Zhejiang Technology and Business University, 2013.
- [8] Shi D, Ping W, Khushnood A. A survey on DL and its applications, *Computer Science Review*, 2021, 40: 1574-0137.
- [9] Hinton G E, Salakhutdinov R R. Reducing the Dimensionality of Data with Neural Networks. *Science*, 2006, 313, 504-507.
- [10] Mariusz Bi. Przemysław Włodarczyk, mgr inż. Adam Gołda (2005) Katedra Elektroniki AGH, Last modified: 06.09.2004
- [11] Kalta S, Bhatt R. A Comparison Analysis of Heart Disease Dataset Using Decision Tree and Back-Propagation Network. 2021 Sixth International Conference on Image Information Processing (ICIIP), Shimla, India, 2021, 556-561.