Babylonian Journal of Networking
Vol.2024, **pp**. 123–135
DOI: https://doi.org/10.58496/BJN/2024/014; ISSN: 3006-5372
https://mesopotamian.press/journals/index.php/BJN

Research Article

Survey on Neural Networks in Networking: Applications and Advancements

Jenan Ayad [1,*]

[1] *Electro-mechanical Engineering dep, University of Technology, Baghdad, Iraq*

ARTICLE INFO

Article History
Received 08 May 2024
Accepted 10 Jul 2024
Published 02 Aug 2024

Keywords
Neural Networks
Networking
Machine Learning
Network Optimization
Traffic Prediction
Anomaly Detection
Network Security
Deep Learning
Network Performance
Artificial Intelligence

ABSTRACT

The integration of neural networks into networking has paved the way for substantial advancements in network performance, security, and management. This survey delves into the diverse applications of neural networks within the networking domain, highlighting their impact on traffic prediction, anomaly detection, network optimization, and security enhancements. By leveraging the inherent capabilities of neural networks to model complex and non-linear relationships, significant improvements in efficiency and reliability can be achieved. This survey provides a thorough examination of current methodologies, key advancements, and practical implementations, offering insights into the future potential of neural networks in the field of networking.

1. INTRODUCTION

The advent of neural networks has revolutionized various domains, including the field of networking. Neural networks, known for their ability to learn and model complex patterns and relationships, have found extensive applications in networking. These applications range from predicting network traffic and detecting anomalies to optimizing network performance and enhancing security measures [1][2].

In the context of networking, neural networks facilitate the efficient handling of vast amounts of data, enabling more accurate and real-time decision-making processes [3]. The ability to adapt to changing network conditions and detect subtle patterns makes neural networks invaluable for improving network reliability and performance [4]. Furthermore, advancements in deep learning and neural network architectures have opened new avenues for research and development, pushing the boundaries of what can be achieved in networking [5].

This survey aims to provide a comprehensive overview of the current state of neural network applications in networking. It explores the theoretical underpinnings, practical implementations, and recent advancements in this field. By examining various case studies and real-world applications, the survey highlights the transformative impact of neural networks on modern networking technologies

2. CLASSIFICATION, CLUSTERING, AND REGRESSION

Neural networks have become a cornerstone in the field of machine learning, offering robust solutions for a variety of tasks such as classification, clustering, and regression. These tasks are fundamental to many real-world applications, including image recognition, customer segmentation, and predictive analytics [1-4]. Classification involves assigning inputs to predefined categories, making it essential for tasks like email filtering, disease diagnosis, and object detection [3,6].

*Corresponding author. Email: jinan.a.namuq@uotechnology.edu.iq

Clustering groups similar data points together without prior labels, which is crucial for market segmentation, social network analysis, and anomaly detection[5,7]. Regression predicts continuous outcomes based on input features, playing a key role in financial forecasting, environmental modeling, and resource management [8,9] . Each of these tasks leverages the powerful capabilities of neural networks to learn and model complex patterns and relationships within data. Neural networks, with their layered architecture and non-linear processing capabilities, can capture intricate dependencies that traditional models might miss. This enables advanced analytical and predictive performance, allowing for more accurate and efficient solutions to complex problems [10]. Neural networks are highly versatile and can be tailored to specific tasks by adjusting their architecture, activation functions, and training algorithms. This flexibility has led to their widespread adoption across various industries, driving innovations in technology, healthcare, finance, and more [2].

This paper delves into the mechanisms, advantages, disadvantages, and practical applications of neural networks in classification, clustering, and regression, providing a comprehensive understanding of these essential machine learning techniques. By exploring these areas, we can appreciate the transformative impact of neural networks in the modern data-driven world. Figure 1 shows an overview of applications in machine learning and how they relate to each type.

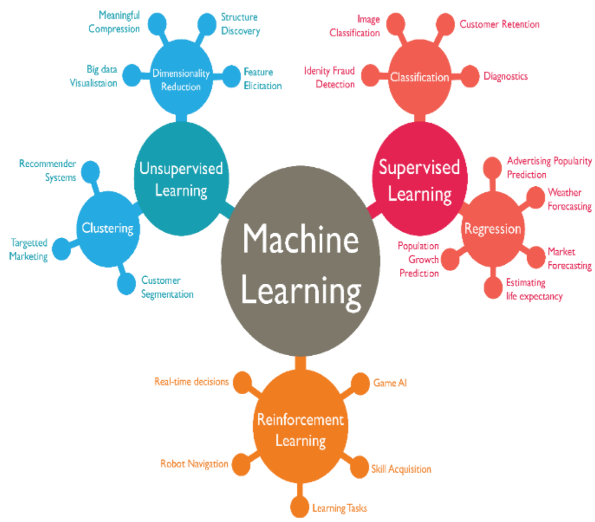


Fig. 1. overview of applications in machine learning.

A. **Classification**

Classification is a supervised learning task where the goal is to predict the categorical class labels of new instances based on past observations. Neural networks are widely used for classification tasks due to their ability to model complex relationships and patterns in data [6].

First. **How Works:** In a neural network used for classification:

a) Input Layer: The input layer receives the input features.
b) Hidden Layers: One or more hidden layers process the inputs with activation functions like ReLU (Rectified Linear Unit).
c) Output Layer: The output layer typically uses a softmax activation function to output class probabilities. Figure 2 (a and b) displays how classification works, and output.

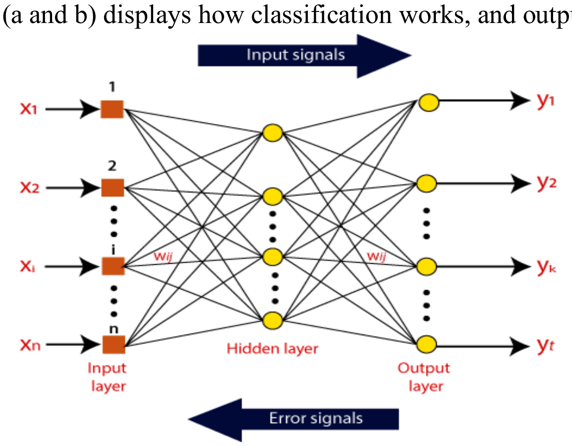


Fig. 2 (a). Structure of classification ANN.

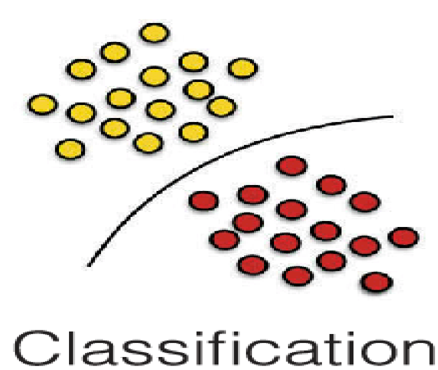


Fig. 2 (b). Output of classification.

Second. Advantages and Disadvantages

Neural networks are widely used for classification tasks due to their ability to learn complex patterns from data [9]. However, while they offer several advantages, they also come with certain disadvantages that need to be considered. Table 1 outlines the key advantages and disadvantages of using neural networks for classification.

TABLE I. ADVANTAGES AND DISADVANTAGES OF NEURAL NETWORKS FOR CLASSIFICATION

Advantages	Disadvantages
Flexibility: Can model complex, non-linear decision boundaries.	Computationally Intensive: Requires significant computational resources and time for training.
Accuracy: Often provides high accuracy in classification tasks with large datasets.	Data Requirements: Needs a large amount of labeled data for training.
Feature Learning: Automatically learns the best features for classification from raw data.	Interpretability: Often seen as a "black box" model, making it hard to interpret results.

B. Clustering

Clustering is an unsupervised learning task where the goal is to group a set of objects in such a way that objects in the same group (or cluster) are more similar to each other than to those in other groups [5]. Neural networks can be used for clustering through techniques like autoencoders and self-organizing maps (SOMs).

First. How Works:

a) Input Layer (x1,x2,…..xj): Takes the input data.
b) Encoding Layers: Compress the input data into a lower-dimensional representation.
c) Clustering Layer (c1,c2,….ck): Assigns cluster labels based on the compressed representations. Figure 3 (a and b) displays how clustering works, and output

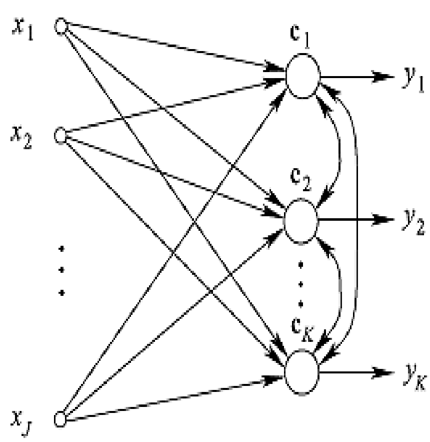


Fig. 3 (a). Structure of clustering ANN.

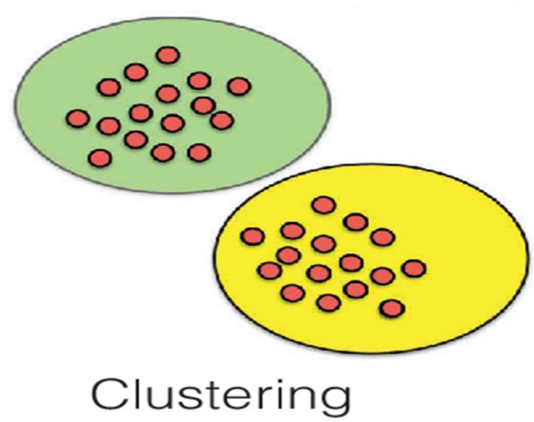


Fig. 3 (b). Output of clustering.

Second. Advantages and Disadvantages

Clustering with neural networks is a powerful technique for grouping similar data points without prior labels [3]. This approach is especially valuable for exploratory data analysis and discovering hidden patterns in data. However, like any method, it has its strengths and challenges. Table 2 outlines the main advantages and disadvantages of clustering using neural networks.

TABLE II. ADVANTAGES AND DISADVANTAGES OF CLUSTERING USING NEURAL NETWORKS.

Advantages	Disadvantages
No Labeled Data Needed: Works with unlabeled data, making it useful for exploratory data analysis.	Choosing the Number of Clusters: Often requires a predefined number of clusters.
Dimensionality Reduction: Can handle high-dimensional data efficiently.	Complexity: May involve complex architectures and tuning.
Adaptability: Can be fine-tuned to adapt to different clustering tasks.	Scalability: Computationally intensive, especially with large datasets.

C. Regression

Regression is a supervised learning task where the goal is to predict a continuous target variable. Neural networks, especially deep networks, can effectively model the relationship between input features and a continuous output [6].

First. How Works:

a) Input Layer: Receives the input features.
b) Hidden Layers: Use activation functions to transform inputs through multiple layers.
c) Output Layer: Outputs a continuous value, often using a linear activation function. Figure 4 (a and b) displays how regression works, and output.

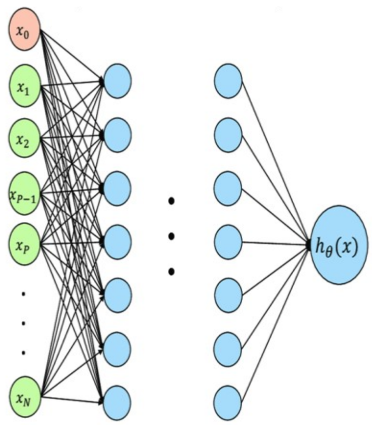


Fig. 4 (a). Structure of regression (ANN).

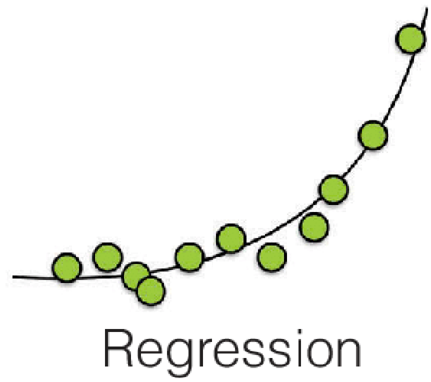


Fig. 4 (b). Output of regression.

Second. Advantages and Disadvantages

Neural networks are widely used for regression tasks due to their ability to model complex, non-linear relationships between variables. However, they also present certain challenges that need to be managed effectively [9]. Table 3 provides an overview of the key advantages and disadvantages of using neural networks for regression.

TABLE III. ADVANTAGES AND DISADVANTAGES OF NEURAL NETWORKS FOR REGRESSION

Advantages	Disadvantages
Flexibility: Can model complex, non-linear relationships.	Overfitting: Prone to overfitting, especially with small datasets.
Accuracy: Capable of achieving high accuracy with large and rich datasets.	Computational Resources: Requires significant computational resources for training.
Generalization: Good at generalizing from training data to unseen data.	Hyperparameter Tuning: Needs careful tuning of hyperparameters.

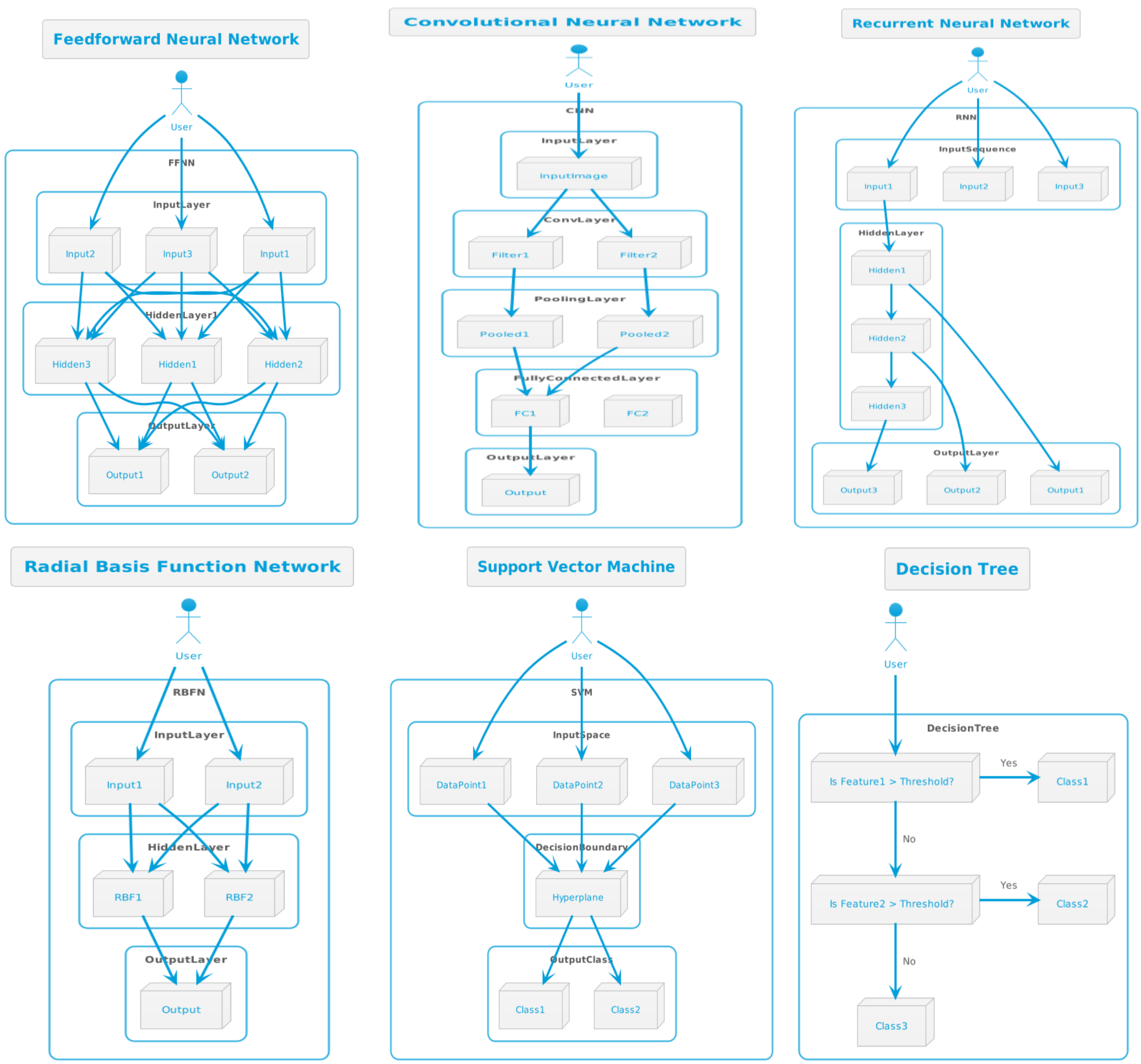
1) Most Important Algorithms Classification, Clustering, and Regression

Table 4 above provides a consolidated overview of some of the most important AI algorithms used for classification, clustering, and regression tasks. Each algorithm is categorized by its primary task and briefly described in terms of its functionality and typical applications. This summary aims to facilitate understanding of the key techniques employed in various machine learning applications.

TABLE IV. OVERVIEW OF IMPORTANT AI ALGORITHM.

Algorithm	Task	Description
Classification		
Logistic Regression	Binary Classification	Linear model based on the logistic function.
Decision Trees	Classification	Tree-like models that split data based on feature values.
Random Forest	Classification	Ensemble method using multiple decision trees for improved accuracy.
Support Vector Machines (SVM)	Classification	Finds optimal hyperplanes for linear and non-linear classification.
k-Nearest Neighbors (k-NN)	Classification	Instance-based learning using proximity to classify new examples.
Naive Bayes	Classification	Probabilistic classifier based on Bayes' theorem with strong independence assumptions.
Neural Networks	Classification	Deep learning models with multiple layers for learning complex patterns.
Clustering		
K-means	Clustering	Partitioning method that divides data into K clusters based on centroids.
Hierarchical Clustering	Clustering	Builds a tree of clusters where each node is a cluster.

DBSCAN	Clustering	Clusters data based on density, suitable for data with complex shapes.
Mean Shift	Clustering	Identifies dense areas and shifts centroids to maximize data points within clusters.
EM Algorithm	Clustering	Iterative method for fitting mixture models, particularly Gaussian Mixtures.
Regression		
Linear Regression	Regression	Predicts continuous values based on linear relationships.
Ridge Regression	Regression	Linear regression with L2 regularization to prevent overfitting.
Lasso Regression	Regression	Linear regression with L1 regularization, useful for feature selection.
Decision Trees for Regression	Regression	Tree-based method for predicting continuous values.
Random Forest for Regression	Regression	Ensemble method using multiple decision trees for regression tasks.
Gradient Boosting Machines (GBM)	Regression	Boosting technique combining weak learners to improve accuracy.
Neural Networks for Regression	Regression	Deep learning models designed for regression tasks using neural layers.



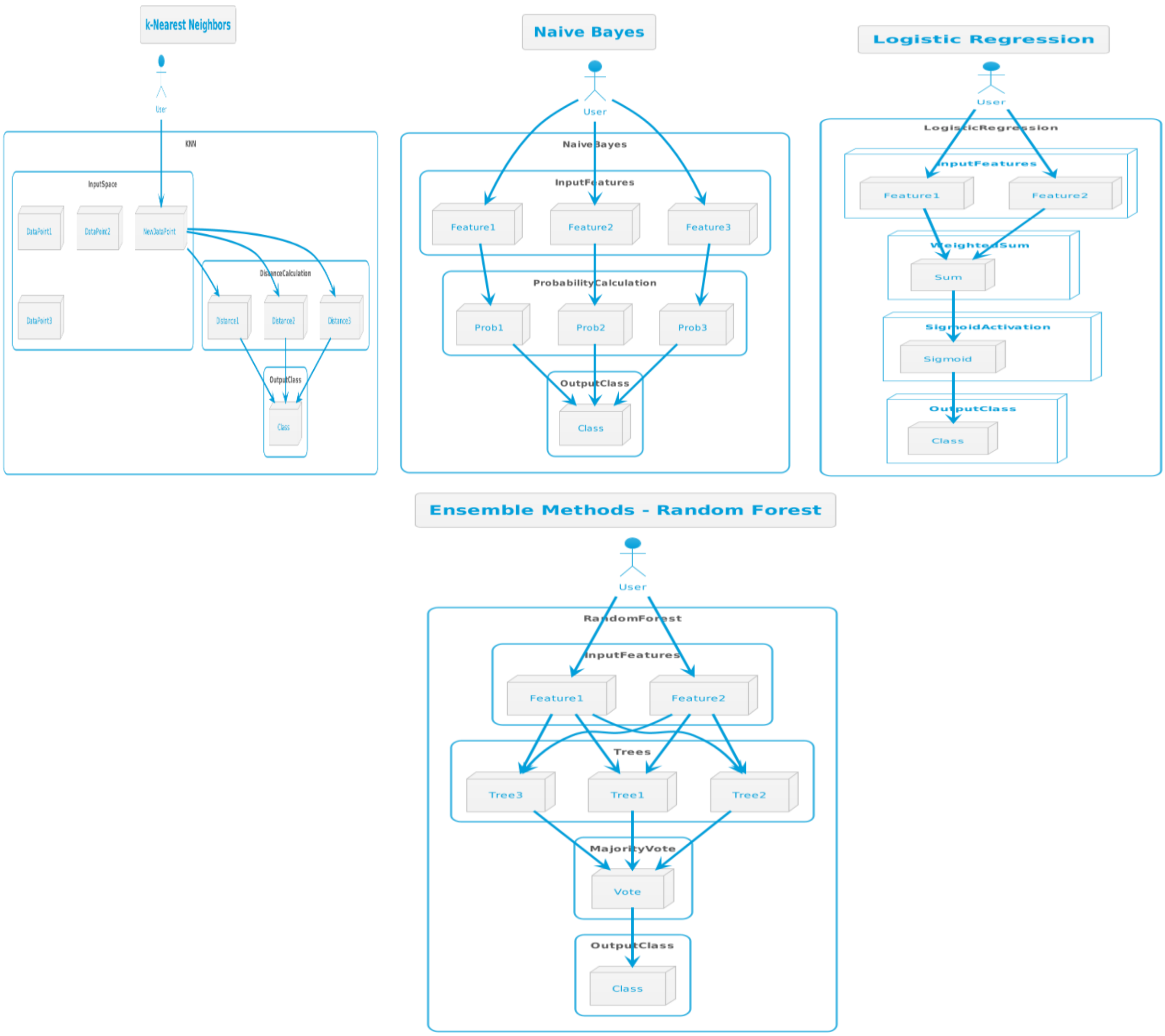


Fig. 5. Structures some important algorithms

2) **Activation Functions**

Activation functions are a crucial component of neural networks, enabling them to capture complex patterns in data by introducing non-linearity. This report explores six widely used activation functions: Sigmoid, Tanh, ReLU, Leaky ReLU, Softmax, and Swish. We will discuss their mathematical formulations, advantages, disadvantages, applications, and provide additional context to enhance understanding. In Figure 6 shows the activation functions in ANN [11-16].

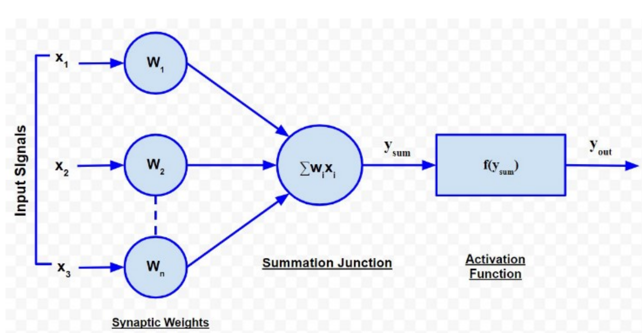


Fig. 6: Structure of ANN.

2.1. Sigmoid Activation Function [11]

The Sigmoid function, a fundamental building block in neural network understanding, has been utilized since the early days of artificial neural networks due to its probabilistic interpretation and smooth gradient. Although less common in deep networks due to the vanishing gradient problem, it remains useful in certain applications. It is by far the most commonly used activation function in neural networks. The need for sigmoid function stems from the fact that many learning algorithms require the activation function to be differentiable and hence continuous. sigmoid function as shown in Figure 7. A binary sigmoid function is of the form:

$$y_{\{out\}} = f(x) = \frac{\{1\}}{\{1 + e^{\{-kx\}}\}} \quad (1)$$

$Where\ k =$ steepness or slope parameter

$By\ varying\ the\ value\ of\ k, sigmoid\ function\ withdifferent$

$slopes\ can\ be\ obtained. It\ has\ a\ range\ of\ (0,1).\ The\ slope\ of\ origin\ is\ \frac{k}{4}$

$As\ the\ value\ of\ k\ becomes\ very\ large, the\ sigmoid\ function\ becomes\ a\ threshold\ function.$

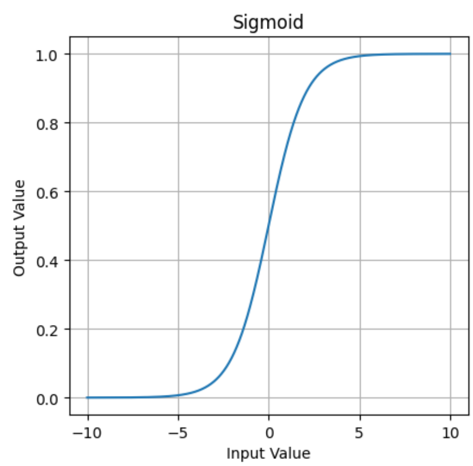


Fig. 7. sigmoid function.

2.1.1. Advantages and Disadvantages of Sigmoid Function

The Sigmoid activation function is widely used in neural networks, particularly for binary classification tasks. Table 5 summarising its key advantages and disadvantages.

TABLE V. ADVANTAGES AND DISAVDANTAGES OF SF.

Advantages	Disadvantages
Smooth Gradient: The function is smooth, providing a gradient that helps in gradient-based optimization.	Vanishing Gradient Problem: Gradients become very small for large absolute values of x, slowing down learning.
Output Range: Outputs are bounded between 0 and 1, making it suitable for binary classification problems.	Slow Convergence: Training can be slow due to the vanishing gradient problem.
Clear Predictions: Outputs can be interpreted as probabilities.	

2.1.2. Applications of Sigmoid Function

a. Commonly used in binary classification tasks.
b. Often used in the output layer of neural networks for probability predictions.

2.2. Hyperbolic Tangent (Tanh) Activation Function [12]

The Tanh function, introduced to address the limitations of the Sigmoid function, is popular in certain contexts where zero-centered outputs are beneficial, despite newer activation functions. Tanh function as shown in Figure 8. The Tanh function is defined as:

$$\mathrm{Tanh}(x) = \frac{e^{x} - e^{-x}}{e^{x} - e^{-x}} \quad (2)$$

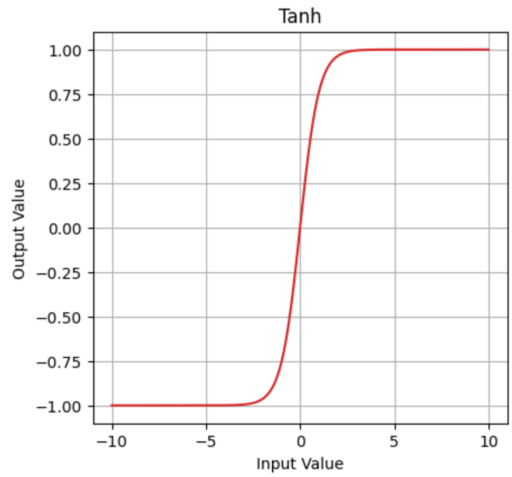


Fig. 8. Tanh function.

2.2.1. Advantages and Disadvantages of Tanh Function

The tanh activation function offers benefits such as a centered output range and zero-centered outputs, aiding in faster convergence and optimization. However, it shares the disadvantage of the sigmoid function, namely the vanishing gradient problem, particularly for large input values. As shown in Table 6.

TABLE VI. ADVANTAGES AND DISADVANTAGES OF TF.

Advantages	Disadvantages
Output Range: Outputs range between -1 and 1,	Vanishing Gradient Problem: Tanh suffers from vanishing gradients for large
centering the data which helps in faster	values of x, similar to the Sigmoid function.
Clear Predictions: Outputs can be interpreted as probabilities.	
convergence.	
Zero-centered: Makes optimization easier and	
more efficient.	

2.2.2. Applications of Tanh Function

a. Often used in hidden layers of neural networks.
b. Suitable for regression and classification problems where the data needs to be centered.

4.3. Rectified Linear Unit (ReLU) Activation Function [13]

ReLU, introduced in 2011 by Glorot, Bordes, and Bengio, is a simple and effective method for reshaping Boltzmann machines. Its widespread adoption in deep learning has led to the development of variants like Parametric ReLU (PReLU) and Exponential Linear Unit (ELU) for improved performance. ReLU function as shown in Figure 9. The ReLU function is defined as:

$$ReLU(x) = \max(0, x) \tag{3}$$

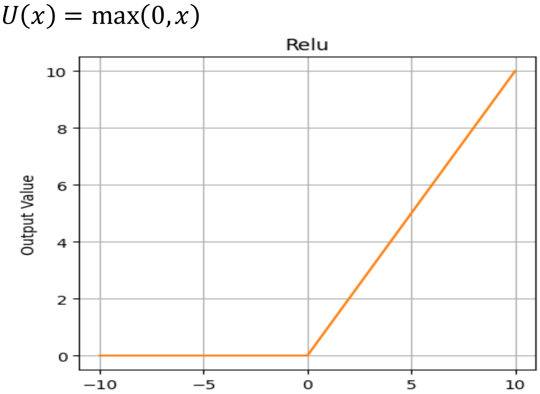


Fig. 9. ReLU function.

4.3.1. Advantages and Disadvantages of ReLU Function

The Rectified Linear Unit (ReLU) activation function is popular in neural networks for its simplicity and effectiveness in addressing the vanishing gradient issue commonly found in deep networks using sigmoid or tanh activations. However, it's important to watch out for the "Dying ReLU" problem where neurons can stop learning if their outputs consistently remain at 0 due to negative inputs. As shown in Table 7.

TABLE VII. ADVANTAGES AND DISADVANTAGES OF RELU.

Advantages	Disadvantages
Computational Efficiency: Simple mathematical formulation makes it computationally efficient.	Dying ReLU Problem: Neurons may become inactive, leading to 0 outputs if inputs fall into the negative range, which can halt learning.
Mitigates Vanishing Gradient: Helps avoid the vanishing gradient problem, facilitating faster learning.	

4.3.2. Applications of ReLU Function

a. Widely used in hidden layers of deep learning models
b. Suitable for various tasks like image recognition and natural language processing.

4.4. Leaky ReLU Activation Function [14]

Leaky ReLU addresses dying ReLU problem by providing small slope for negative inputs, with recent advances like Randomized Leaky ReLU (RReLU) improving performance and flexibility. Leaky ReLU function as shown in Figure 10. The Leaky ReLU function is defined as:

$$Leaky\ ReLU(x) = \begin{cases} x & if \quad x \geq 0 \\ \alpha x & if \quad x < 0 \end{cases} \tag{4}$$

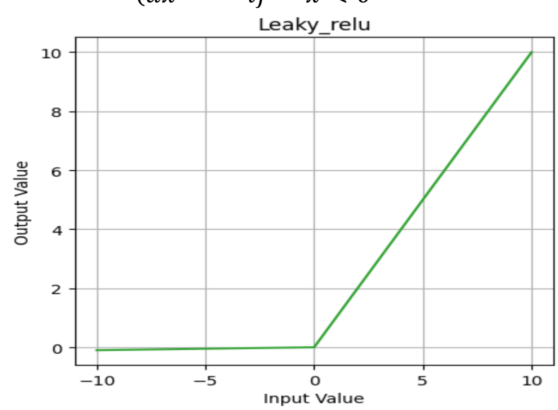


Fig. 10. Leaky ReLU function.

4.4.1. Advantages and Disadvantages of Leaky ReLU Function

The Leaky ReLU activation function addresses the issue of "dying ReLU" by allowing a small, non-zero gradient for negative inputs, which can help maintain the flow of gradients during training. However, this slight modification also introduces slightly more computational complexity compared to the standard ReLU activation function. As shown in Table 8.

TABLE VIII. ADVANTAGES AND DISADVANTAGES OF LEAKY RELU.

Advantages	Disadvantages
Prevents Dying ReLU: Allows a small gradient when x is negative, preventing neurons from becoming inactive.	Less Efficient: Slightly more complex computation compared to ReLU.

4.4.2. Applications of Leaky ReLU Function

a. Useful in deep networks where the dying ReLU problem is observed.
b. Suitable for tasks requiring robust learning.

4.5. Softmax Activation Function [15]

Softmax, a widely used feature in neural networks for classification tasks, has seen recent advancements in optimization techniques, improving its efficiency in large-scale applications, despite its basic function remaining unchanged. Softmax function as shown in Figure 12.

The Softmax function is defined as:

$$softmax(xi) = \frac{e^{\{x_i\}}}{\sum_{\{j\}} e^{x_j}} \quad (5)$$

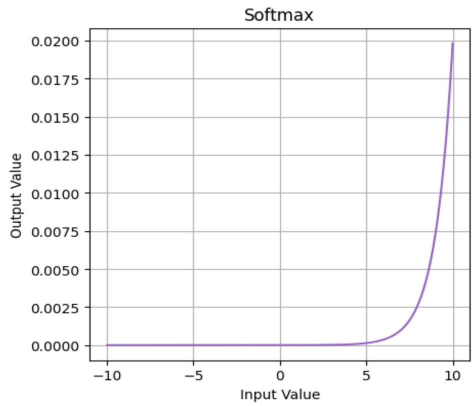


Fig. 11. Softmax function.

4.5.1. Advantages and Disadvantages of Softmax Function

The Softmax activation function is widely used in neural networks, particularly for multi-class classification tasks where it transforms the raw outputs of a network into probabilities. While it provides clear advantages in generating probability distributions over multiple classes, it also comes with computational considerations, especially as the number of classes increases. As shown in Table 9.

TABLE IX. ADVANTAGES AND DISADVANTAGES OF SOFTMAX.

Advantages	Disadvantages
Probability Distribution: Converts outputs into probabilities, making it suitable for multi-class classification.	Computational Complexity: More computationally expensive compared to other activation functions, especially with a large number of classes.

4.5.2. Applications of Softmax Function

Commonly used in the output layer for multi-class classification problems.

4.6. Swish Activation Function [16]

Swish, a deep learning algorithm proposed by Google researchers in 2017, has demonstrated promising results in various benchmarks and is being explored for its potential to outperform traditional activation functions in diverse applications. Swish function as shown in Figure 12.

The Swish function is defined as:

$$Swish(x) = x \cdot \sigma(x) \quad (6)$$

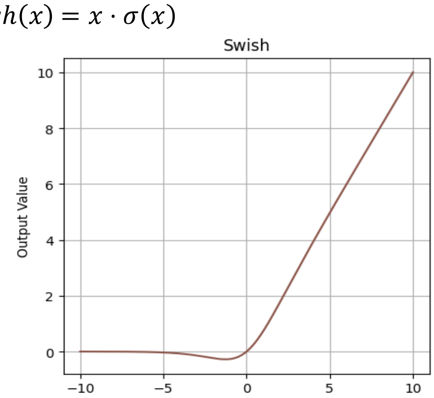


Fig. 12. Swish function.

4.6.1. Advantages and Disadvantages of Swish Function

The Swish activation function has gained attention for its performance benefits over ReLU, particularly in scenarios where smoothness and non-monotonic properties are advantageous. However, it comes with a trade-off in computational complexity compared to ReLU. This table highlights key advantages and disadvantages to consider when evaluating the Swish activation function for neural network architectures. As shown in Table 10.

TABLE X. ADVANTAGES AND DISADVANTAGES OF SWISH.

Advantages	Disadvantages
Smooth and Non-monotonic	Computational Complexity (slightly more than ReLU)
No Vanishing Gradient	

4.6.2. Applications of Swish Function

- **a.** Used in various deep learning models for improved performance.
- **b.** Suitable for tasks requiring advanced pattern recognition.

5. COMPARATIVE PERFORMANCE OF ACTIVATION FUNCTIONS IN NEYRAL NETWORKS

Activation functions are pivotal in determining the efficiency and effectiveness of neural networks. This table compares various activation functions—Sigmoid, Tanh, ReLU, Leaky ReLU, Softmax, and Swish—across different neural network architectures and tasks. The comparison includes key metrics such as training time, convergence speed, accuracy, and their ability to mitigate common issues like the vanishing gradient problem. Table 11 provides a comparative overview of how different activation functions perform across various neural network architectures and tasks. It highlights their strengths and weaknesses, aiding in the selection of appropriate activation functions based on specific application requirements and network design considerations.

TABLE XI. COMPARATIVE PERFORMANCE OF ACTIVATION FUNCTIONS.

Activation Function	Feedforward Networks	Convolutional Neural Networks	Recurrent Neural Networks	Comparative Studies
Sigmoid	Slower convergence, less used due to vanishing gradient	Limited to output layer in binary classification	Used in LSTM gates; slow due to vanishing gradient	Not preferred for deep networks
Tanh	Slower than ReLU; used in hidden layers	Limited to certain RNN architectures; controls information flow	Preferred in LSTM and GRU; handles gradient better	Outperformed by ReLU in many tasks
ReLU	Fast convergence; widely used in hidden layers	Standard choice for CNNs; prevents vanishing gradient	Suitable but can suffer from dying ReLU problem	Dominates in modern architectures
Leaky ReLU	Faster than ReLU in certain cases; prevents dying ReLU	Effective in deep CNNs; maintains non-zero gradient	Used to mitigate exploding gradient in RNNs	Competitive alternative to ReLU
Softmax	Used in output layer for multi-class classification	Standard for multi-class CNN outputs	Not commonly used in RNNs	Essential for classification tasks
Swish	Competitive with ReLU in feedforward networks	Performs well in image recognition tasks	Requires further validation in RNNs	Promising in recent benchmarks

6. REAL-WORLD APPLICATIONS OF ACTIVATION FUNCTIONS [17-18]

In real-world applications, Sigmoid finds use in industries like finance for fraud detection systems. Here, the ability to assign a probability score to transactions helps distinguish between fraudulent and non-fraudulent activities, aiding in automated decision-making processes. Additionally, Sigmoid is applied in natural language processing tasks such as sentiment analysis, where determining the sentiment (positive or negative) of textual data benefits from probabilistic outputs for classification. Real-world applications of Tanh extend to fields like speech recognition, where recurrent neural networks (RNNs) utilize its properties to capture and analyze sequential dependencies in audio signals. This is critical in accurately transcribing spoken language into text, where understanding context over time is essential. In financial analysis, Tanh activation is used to model and predict trends in time series data, leveraging its ability to handle complex, non-linear relationships in financial markets [17-18]

Real-world applications of ReLU span across industries where image processing and pattern recognition are pivotal. In computer vision applications, ReLU enables CNNs to effectively learn hierarchical features from images, contributing to advancements in automated driving systems and medical imaging diagnostics. Moreover, in natural language processing (NLP), ReLU is used in models for text classification and sentiment analysis, where rapid processing of textual data and feature extraction are critical for accurate predictions. Real-world applications of Leaky ReLU include robotics and game development, where continuous learning and adaptation to diverse inputs are essential. In robotics, Leaky ReLU facilitates the robust processing of sensory data and decision-making in autonomous systems. Similarly, in game AI, Leaky ReLU aids in dynamic decision-making processes by maintaining responsiveness to varying game states and player interactions.

Real-world applications of Softmax span domains such as image and speech recognition, where distinguishing between multiple classes is essential. In computer vision, Softmax enables CNNs to identify and categorize objects within images based on learned features, supporting applications in autonomous driving and surveillance systems. Similarly, in NLP, Softmax is used in language models for predicting the next word in a sequence or classifying text into multiple categories based on semantic content. Real-world applications of Swish are emerging in fields like automated driving and financial forecasting, where handling complex, non-linear relationships in data is critical. In automated driving systems, Swish enhances the processing of real-time sensor data and decision-making capabilities, contributing to safer and more efficient autonomous vehicles. Additionally, in financial markets, Swish is applied to predict stock prices and market trends, where accurately capturing subtle patterns in financial data is crucial for informed decision-making.

Each activation function—Sigmoid, Tanh, ReLU, Leaky ReLU, Softmax, and Swish—offers unique advantages and applications across different neural network architectures and real-world scenarios. Understanding the real-world applications helps in selecting the most suitable activation function based on specific task requirements, data characteristics, and computational considerations. By leveraging the strengths of each activation function, researchers and practitioners can optimize neural network performance and advance applications in diverse fields ranging from healthcare and finance to robotics and autonomous systems.

7. CONCLUSION

The exploration of neural networks within the field of networking has revealed significant potential for enhancing various aspects of network performance, security, and management. By leveraging the sophisticated capabilities of neural networks to model and interpret complex data patterns, substantial advancements have been made in tasks such as traffic prediction, anomaly detection, and network optimization. This survey has provided a comprehensive overview of the theoretical foundations, key methodologies, and practical implementations of neural networks in networking. The transformative impact of neural networks is evident in their ability to improve the efficiency and reliability of network operations. Through real-time decision-making and adaptability to changing network conditions, neural networks contribute to more robust and resilient network infrastructures. Furthermore, advancements in deep learning and neural network architectures continue to push the boundaries of innovation, opening new avenues for research and practical applications in the networking domain. As the field of networking continues to evolve, the integration of neural networks will undoubtedly play a pivotal role in addressing emerging challenges and driving technological progress. By fostering a deeper understanding of the interplay between neural networks and networking, this survey aims to inspire further research and development, ultimately contributing to the advancement of modern networking technologies.

Conflicts Of Interest

No competing relationships or interests that could be perceived as influencing the research are reported in the paper.

Funding

The lack of a funding acknowledgment in the paper indicates that no financial support was provided by any institution or sponsor.

Acknowledgment

The author acknowledges the institution for the intellectual resources and academic guidance that significantly enriched this research.

References

[1] X. Wang and X. Wang, "A deep stochastic weight assignment network and its application to chess playing," Journal of Parallel and Distributed Computing, vol. 117, pp. 205-211, 2020.

[2] H. Chen, M. Yao, and Q. Gu, "Pothole detection using location-aware convolutional neural networks," International Journal of Machine Learning and Cybernetics, 2020. [Online]. Available: https://doi.org/10.1007/s13042-020-01112-2

[3] J. Li, W. W. Y. Ng, X. Tian, S. Kwong, and H. Wang, "Weighted multi-deep ranking supervised hashing for efficient image retrieval," International Journal of Machine Learning and Cybernetics, 2020.

[4] J. Zhang, S. Ding, and W. Jia, "An adversarial non-volume preserving flow model with Boltzmann priors," International Journal of Machine Learning and Cybernetics, 2020.

[5] W. Zhongmin, Z. Xiaoxiao, W. Wenlang, and L. Chen, "Emotion recognition using multimodal deep learning in multiple psychophysiological signals and video," International Journal of Machine Learning and Cybernetics, 2020.

[6] O. Deniz, N. Vallez, J. Salido, and G. Bueno, "Robustness to Adversarial Examples can be Improved with Overfitting," International Journal of Machine Learning and Cybernetics, 2020.

[7] Z. Cai and N. Vasconcelos, "Cascade R-CNN: High quality object detection and instance segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019.

[8] A. Bulat and G. Tzimiropoulos, "Human pose estimation via convolutional part heatmap regression," in Computer Vision – ECCV 2016, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Cham: Springer, 2016, vol. 9906.

[9] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," arXiv preprint, arXiv:1610.02357, 2017.

[10] O. Chapelle, "Support vector machines for image classification," IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998.

[11] K. Chellapilla, S. Puri, and P. Simard, "High performance convolutional neural networks for document processing," in Tenth International Workshop on Frontiers in Handwriting Recognition, 2006.

[12] W. Chen, J. T. Wilson, S. Tyree, et al., "Compressing neural networks with the hashing trick," in 32nd International Conference on Machine Learning, ICML 2015. [Online]. Available: https://arxiv.org/abs/1504.04788

[13] M. Chevalier, N. Thome, M. Cord, et al., "LR-CNN for fine-grained classification with varying resolution," in 2015 IEEE International Conference on Image Processing (ICIP). [Online]. Available: https://doi.org/10.1109/ICIP.2015.7351572

[14] D. C. Cireşan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "High-performance neural networks for visual object classification," arXiv preprint, arXiv:1102.0183, 2011.

[15] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in Advances in Neural Information Processing Systems, pp. 2843-2851, 2012.

[16] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Mitosis detection in breast cancer histology images with deep neural networks," in Proceedings of Medical Image Computing and Computer-Assisted Intervention, MICCAI 2013, pp. 411-418. [Online]. Available: https://doi.org/10.1007/978-3-319-10404-3_51

[17] Z. Ali Abbood, D. Çağdaş Atilla, and Ç. Aydin, "Intrusion Detection System Through Deep Learning in Routing MANET Networks," Intelligent Automation & Soft Computing, vol. 37, no. 1, pp. 269-281, 2023. [Online]. Available: https://doi.org/10.32604/iasc.2023.035276

[18] Z. A. Abbood, D. Ç. ATİLLA, and Ç. AYDIN, "Enhancement of the Performance of MANET Using Machine Learning Approach Based on SDNs," Optik, vol. 272, p. 170268, Feb. 2023. [Online]. Available: https://doi.org/10.1016/j.ijleo.2022.170268