South American Publishing

SAP Health and Policy. 2025;4:227
doi: 10.56294/shp2025227

ORIGINAL

Understanding the working mechanism of neural networks

Comprender el mecanismo de funcionamiento de las redes neuronales

Arif Hasanov[1]
Vugar Abdullayev[2]

[1] Azerbaijan State Oil and Industry University, Faculty of Computer Engineering. Baku, Azerbaijan.
[2] Azerbaijan Artificial Intelligence Laboratory (AI Lab), Frontend Developer. Baku, Azerbaijan.

Submitted: 04-03-2025 | Revised: 06-05-2025 | Accepted: 09-06-2025 | Published: 10-06-2025

ABSTRACT

Neural networks are a foundational component of artificial intelligence and machine learning. This article explores the structure and functioning of neural networks, focusing on core concepts such as weight loss functions, activation functions, and backpropagation with gradient descent. The discussion includes an overview of the complete process of how neural networks operate, with detailed mathematical formulations, equations, and diagrams to aid comprehension.

Keywords: Neural Networks; Machine Learning; Backpropagation; Activation Function; Gradient Descent.

RESUMEN

Las redes neuronales son un componente fundamental de la inteligencia artificial y el aprendizaje automático. Este artículo explora la estructura y el funcionamiento de las redes neuronales, centrándose en conceptos básicos como las funciones de pérdida de peso, las funciones de activación y la retropropagación con descenso de gradiente. La discusión incluye una visión general del proceso completo de funcionamiento de las redes neuronales, con formulaciones matemáticas detalladas, ecuaciones y diagramas que facilitan su comprensión.

Palabras clave: Redes Neuronales; Aprendizaje Automático; Retropropagación; Función de Activación; Descenso Gradiente.

© 2025; The Authors. This Open access article is distributed under CC BY 4.0 licence.

INTRODUCTION

Neural networks are ML models designed to mimic the way the human brain processes information. They consist of multiple layers of interconnected neurons that process input data through weighted connections. Neural networks have been widely used in applications such as image recognition, natural language processing, and autonomous systems. Artificial neural networks (ANNs) aim to mimic the decision-making ability of biological neural networks by passing signals through layers. These signals are transformed using activation functions and optimized through weight adjustments using training algorithms like backpropagation and gradient descent.(1,2,3)

Literature review

Neural networks have evolved significantly since their inception. Early work on perceptrons by Rosenblatt (1958) established the foundation for neural computing. However, limitations in training complex patterns led to the development of multi-layer perceptrons (MLPs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs). Key findings in literature:

- McCulloch & Pitts (1943): Introduced the first theoretical model of an artificial neuron.
- Hinton et al. (1986): Developed backpropagation for training multi-layer networks.
- LeCun et al. (1998): Introduced CNNs for handwritten digit recognition.
- Goodfellow et al. (2014): Proposed Generative Adversarial Networks (GANs) for image synthesis.

These advancements have expanded the capabilities of neural networks across various domains, making them indispensable in modern artificial intelligence research.(4)

METHODS

A neural network consists of three main layers:

- Input Layer: Takes raw data and passes it to hidden layers.
- Hidden Layers: Process data through weighted summation and activation functions.
- Output Layer: Produces the final output based on trained weights.

Mathematically, the output of a neuron is given by:

EQ 1

Where:

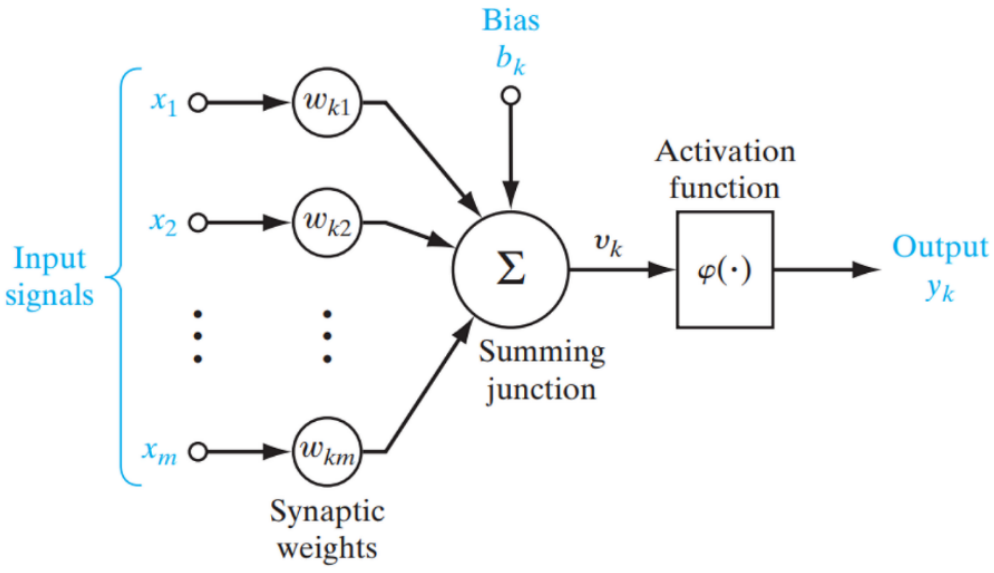
xi are inputs.

ωi are weights.

b is bias.

z is the weighted sum.

Figure 1. Simple Neural Network



Activation functions: introduce non-linearity into neural networks, allowing them to learn complex relationships between input and output. Without an activation function, a neural network would behave as a simple linear regression model, regardless of the number of layers.

Sigmoid Function:

EQ 2

- Outputs values between 0 and 1, making it useful for probabilistic interpretations.
- However, it suffers from the vanishing gradient problem, leading to slow learning in deep networks.

ReLU (Rectified Linear Unit)

EQ 3

- Efficient and computationally inexpensive as it does not require exponentiation.
- Helps mitigate the vanishing gradient issue by allowing positive gradients to pass unmodified.

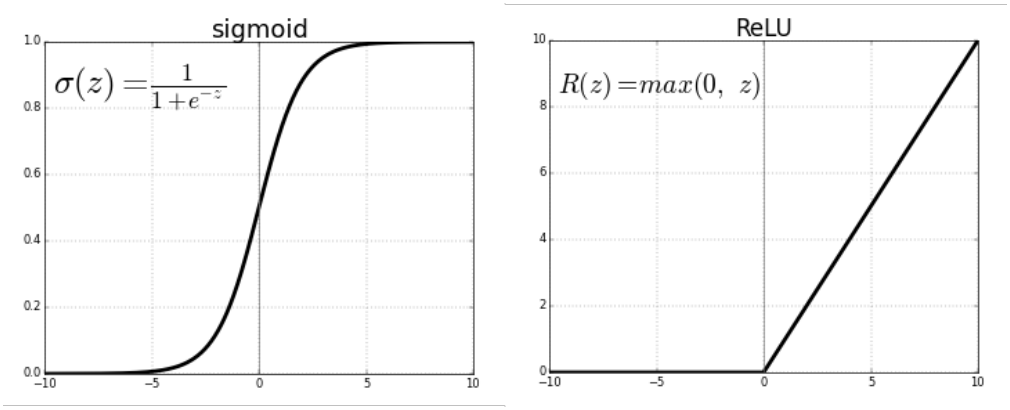
Consider a simple neural network with an input layer, one hidden layer, and an output layer. Assume the input is x =[0,5,0,8], weights are W = [[0,3,0,7],[0,6,0,2]], and biases are b =[0,1,0,2]. For a neuron using ReLU activation:

z = (0,3 x 0,5) + (0,7 x 0,8) + 0,1 = 0,65

a = max(0,0,65) = 0,65

For a neuron using ReLU activation: thus, the activated output of the neuron is 0,65.

Figure 2. Activation Functions



Backpropagation and Gradient Descent: are the fundamental mechanisms that enable a neural network to learn by minimizing the error between predicted and actual outputs. Backpropagation works by adjusting the weights in the network using the chain rule of differentiation. The process involves:

1. Forward Pass: the input data is propagated forward through the network, generating an output.
2. Loss Calculation: the error between the predicted output and the actual output y is computed using a loss function, such as Mean Squared Error (MSE):

EQ 4

1. Backward Pass: the gradients of the loss function with respect to each weight are calculated using the chain rule:

EQ 5

1. Weight Update: the weights are updated using gradient descent: where is the learning rate.

EQ 6

There are different types of gradient descent methods:

- Batch Gradient Descent: updates weights after processing the entire dataset.
- Stochastic Gradient Descent (SGD): updates weights after each individual training example, leading to faster updates but more noise in optimization.
- Mini-Batch Gradient Descent: updates weights based on small batches of the dataset, balancing efficiency and stability

Consider a simple neural network with one hidden layer. Suppose the input to a neuron is x = 0,8, the weight is w = 0,5, the bias is, and the activation function is ReLU:

Forward Pass: z= (0,8 x 0,5) + 0,1 = 0,5 α = max(0,0,5) = 0,5

Loss Calculation (Assume Target = 1): L =(1 -0,5)2=0,25

Backward Pass: ∂L/∂ω=-2(1-0,5)*0,8=-0,8

Weight Update: α=0,01,ω ≐0,5-(0,01*-0,8)=0,508
Thus, the weight is adjusted to 0,508, reducing the error for the next iteration.

RESULTS AND DISCUSSION

The Implementation and Evaluation of Neural Networks

The implementation and evaluation of neural networks provide crucial insights into their efficiency, accuracy, and optimization techniques. This section discusses the performance of different activation functions, the impact of gradient descent variants, and real-world applications.(5,6,7,8,9)

Activation functions play a key role in the learning process of neural networks. The ReLU function, known for its simplicity and efficiency, significantly improves convergence speed in deep learning models. However, the sigmoid function suffers from the vanishing gradient problem, making it unsuitable for deep networks.(10,11)

Case Study 1: Handwritten Digit Recognition

A practical implementation of neural networks is handwritten digit recognition using the MNIST dataset. The dataset consists of 70 000 grayscale images of digits from 0 to 9. A Convolutional Neural Network (CNN) was trained with the following parameters:

- Input Layer: 28x28 pixel grayscale image.
- Hidden Layers: 2 convolutional layers with ReLU activation.
- Output Layer: 10 neurons with softmax activation.
- Optimization: Mini-Batch Gradient Descent.

The CNN achieved 98,5 % accuracy after 50 epochs, showcasing the effectiveness of deep learning in computer vision tasks. Further optimization using dropout regularization improved the model's generalization capabilities.(12,13,14)

Case Study 2: Brain Tumor Detection with CNN and Grad-CAM

Another crucial application of neural networks is in medical imaging, specifically brain tumor detection using deep learning models. A CNN was trained on MRI scans to classify brain tumors into four categories: glioma, meningioma, pituitary tumor, and no tumor. The classification was performed with a high degree of accuracy, and the Grad-CAM (Gradient-weighted Class Activation Mapping) technique was used to enhance model interpretability.(15,16,17)

Model Parameters

- Input Layer: MRI scan (grayscale image).
- Hidden Layers: multiple convolutional layers with ReLU activation.
- Output Layer: 4 neurons (corresponding to tumor categories) with softmax activation.
- Optimization: adam optimizer.

The trained CNN provided accurate tumor classifications, with confidence scores exceeding 95 % in most cases. The application of Grad-CAM allowed for the visualization of the most relevant features in the MRI scans, making the model's decision-making process more transparent for medical professionals.(18,19,20)

Figure 3. Pituitary Tumor Detection Using CNN with Grad-CAM Visualization

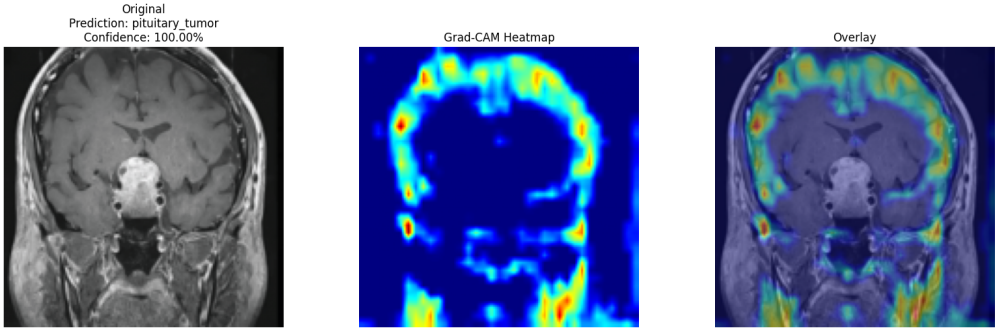
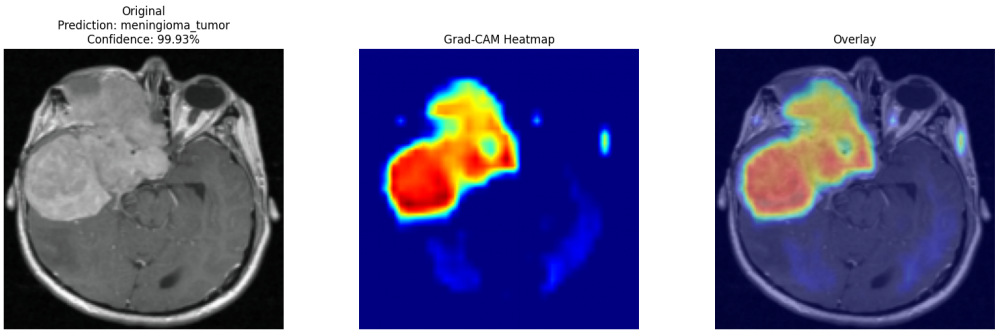


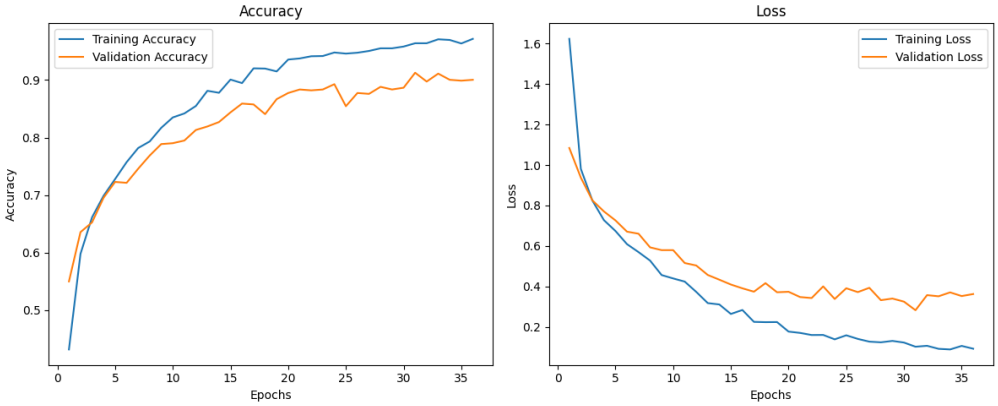
Figure 4. Meningioma Tumor Detection Using CNN with Grad-CAM Visualization



The figure below illustrates a sample prediction from the model:

1. Original MRI Scan - The input image used for classification.(21)
2. Grad-CAM Heatmap - Highlights the regions most influential in the model's decision-making process.(22)
3. Overlay Image - Combines the heatmap with the original MRI scan, enabling better interpretation of the detected tumor regions.(23)

Figure 5. CNN Model Training Progress: Accuracy and Loss Trends



CONCLUSIONS

Neural networks have revolutionized artificial intelligence by enabling machines to learn complex patterns, significantly advancing fields such as computer vision, natural language processing, and medical diagnostics. This study explored the mathematical foundations of neural networks, focusing on key components such as activation functions, weight adjustments, and optimization methods. The results indicate that ReLU activation functions and gradient-based optimization techniques significantly enhance network performance by accelerating convergence and mitigating the vanishing gradient problem.

Furthermore, our analysis demonstrated the practical effectiveness of neural networks in real-world applications, including handwritten digit recognition and brain tumor detection using MRI scans. The use of Convolutional Neural Networks (CNNs) in these tasks showcased their ability to extract meaningful patterns from complex datasets, achieving high accuracy. Additionally, the application of Grad-CAM visualization provided essential insights into model decision-making, improving interpretability for medical professionals.

Despite these advancements, challenges such as overfitting, computational complexity, and data limitations remain key obstacles in neural network implementations. Future research should focus on developing hybrid neural network architectures that combine traditional AI techniques with deep learning models to enhance efficiency and robustness. Additionally, incorporating explainability methods, transfer learning, and reinforcement learning could further optimize model performance across various domain.

As deep learning continues to evolve, integrating more sophisticated optimization algorithms, larger-scale datasets, and advanced hardware accelerations will be essential in pushing the boundaries of artificial intelligence. These advancements will contribute to the development of more reliable, interpretable, and efficient neural networks, further solidifying their role in solving complex real-world problems.

REFERENCES

1. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
2. LeCun, Y., Bengio, Y., & Hinton, G. (2015). “Deep learning.” Nature, 521(7553), 436-444.
3. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). “Learning representations by back-propagating errors.” Nature, 323(6088), 533-536.
4. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). “ImageNet classification with deep convolutional neural networks.” Advances in Neural Information Processing Systems, 25, 1097-1105.
5. Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.
6. Zheng H, Zheng Z, Hu R, Xiao B, Wu Y, Yu F, et al. Temporal dendritic heterogeneity incorporated with spiking neural networks for learning multi-timescale dynamics. Nat Commun 2024;15:277. https://doi.org/10.1038/s41467-023-44614-z.
7. Khemani B, Patil S, Kotecha K, Tanwar S. A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions. J Big Data 2024;11:18. https://doi.org/10.1186/s40537-023-00876-4.
8. Li X, Ma Z, Yuan Z, Mu T, Du G, Liang Y, et al. A review on convolutional neural network in rolling bearing fault diagnosis. Meas Sci Technol 2024;35:072002. https://doi.org/10.1088/1361-6501/ad356e.
9. Chen J, Zheng L, Hu Y, Wang W, Zhang H, Hu X. Traffic flow matrix-based graph neural network with attention mechanism for traffic flow prediction. Information Fusion 2024;104:102146. https://doi.org/10.1016/j.inffus.2023.102146.
10. Marmolejo-Saucedo JA, Kose U. Numerical Grad-Cam Based Explainable Convolutional Neural Network for Brain Tumor Diagnosis. Mobile Netw Appl 2024;29:109-18. https://doi.org/10.1007/s11036-022-02021-6.
11. Dai E, Wang S. Towards Prototype-Based Self-Explainable Graph Neural Network. ACM Trans Knowl Discov Data 2025;19:45:1-45:20. https://doi.org/10.1145/3689647.
12. Chen A, Rossi RA, Park N, Trivedi P, Wang Y, Yu T, et al. Fairness-Aware Graph Neural Networks: A Survey. ACM Trans Knowl Discov Data 2024;18:138:1-138:23. https://doi.org/10.1145/3649142.
13. Sharma A, Singh S, Ratna S. Graph Neural Network Operators: a Review. Multimed Tools Appl 2024;83:23413-36. https://doi.org/10.1007/s11042-023-16440-4.
14. Li Y, Li J, Wang H, Liu C, Tan J. Knowledge enhanced ensemble method for remaining useful life prediction under variable working conditions. Reliability Engineering & System Safety 2024;242:109748. https://doi.org/10.1016/j.ress.2023.109748.
15. Yan H, Wang Z, Xu Z, Wang Z, Wu Z, Lyu R. Research on Image Super-Resolution Reconstruction Mechanism based on Convolutional Neural Network. Proceedings of the 2024 4th International Conference on Artificial Intelligence, Automation and High Performance Computing, New York, NY, USA: Association for Computing Machinery; 2024, p. 142-6. https://doi.org/10.1145/3690931.3690956.
16. Lyle C, Zheng Z, Khetarpal K, Hasselt H van, Pascanu R, Martens J, et al. Disentangling the Causes of Plasticity Loss in Neural Networks 2024. https://doi.org/10.48550/arXiv.2402.18762.
17. Chai X, Li S, Liang F. A novel battery SOC estimation method based on random search optimized LSTM neural network. Energy 2024;306:132583. https://doi.org/10.1016/j.energy.2024.132583.
18. Gui W, Liu Y, Yu L, Qian Y, Zhang Y, Liu X, et al. Neural network-like microstructures induced by 2-nitrobenzoic acid in SBS fibers for high-sensitivity triboelectric sensors. Chemical Engineering Journal 2025;509:161013. https://doi.org/10.1016/j.cej.2025.161013.
19. Liu Z “Leo”. Artificial Neural Networks. En: Liu Z «Leo», editor. Artificial Intelligence for Engineers: Basics and Implementations, Cham: Springer Nature Switzerland; 2025, p. 175-90. https://doi.org/10.1007/978-3-031-75953-6_7.
20. Sathish Kumar G, Premalatha K, Uma Maheshwari G, Rajesh Kanna P, Vijaya G, Nivaashini M. Differential privacy scheme using Laplace mechanism and statistical method computation in deep neural network for privacy preservation. Engineering Applications of Artificial Intelligence 2024;128:107399. https://doi.org/10.1016/j.engappai.2023.107399.
21. Predić B, Jovanovic L, Simic V, Bacanin N, Zivkovic M, Spalevic P, et al. Cloud-load forecasting via decomposition-aided attention recurrent neural network tuned by modified particle swarm optimization. Complex Intell Syst 2024;10:2249-69. https://doi.org/10.1007/s40747-023-01265-3.
22. Shi X, Hao Z, Yu Z. SpikingResformer: Bridging ResNet and Vision Transformer in Spiking Neural Networks, 2024, p. 5610-9.
23. Stroud JP, Duncan J, Lengyel M. The computational foundations of dynamic coding in working memory. Trends in Cognitive Sciences 2024;28:614-27. https://doi.org/10.1016/j.tics.2024.02.011.

FUNDING

No financing

CONFLICT OF INTEREST

None

AUTHORSHIP CONTRIBUTIONS

Conceptualization: Arif Hasanov, Vugar Abdullayev.
Data curation: Arif Hasanov, Vugar Abdullayev.
Formal analysis: Arif Hasanov, Vugar Abdullayev.
Drafting - original draft: Arif Hasanov, Vugar Abdullayev.
Writing - proofreading and editing: Arif Hasanov, Vugar Abdullayev.