WiSARD and Convolutional Neural Networks: Analysis and Comparison of Time Complexity for Digit Recognition Patterns

MARIA VITÓRIA PIMENTEL DE ARAÚJO[1]

[1] *Federal University of Campina Grande, CEEI*
[2] *Federal University of Paraiba*
[*] *maria.araujo@ccc.ufcg.edu.br*

Compiled April 15, 2025

In this work, we present a comparative study between the WiSARD weightless neural network and a pre-trained Convolutional Neural Network (CNN) for the task of handwritten digit classification using the MNIST dataset. We explore both models under the lens of time complexity, training and inference duration, and classification accuracy. While CNNs are widely known for their high accuracy, they are also computationally intensive, requiring substantial training time and hardware acceleration. On the other hand, WiSARD offers a remarkably simple architecture with no need for weight adjustment or backpropagation, leading to extremely fast training even on CPUs. Our experiments demonstrate that, with optimized address sizes and bleaching mechanisms, the WiSARD can achieve up to 90 percent accuracy in under a second of training time, compared to over 99 percent accuracy by CNNs with more than 200 seconds of training. These findings highlight the trade-offs between computational cost and model performance, positioning WiSARD as a viable alternative for fast, interpretable, and lightweight learning in constrained environments.

http://dx.doi.org/10.1364/ao.XX.XXXXXX

1. INTRODUCTION

In recent years, artificial intelligence (AI) has made significant strides in various fields, with deep learning models such as Convolutional Neural Networks (CNNs) becoming the dominant architecture for solving complex problems in image classification, natural language processing, and speech recognition. In particular, CNNs have achieved remarkable performance in tasks such as handwritten digit recognition, most notably on the MNIST dataset. However, while CNNs excel in terms of accuracy and predictive power, they come with a major trade-off: computational cost. CNNs typically require extensive training times, large datasets, and significant computational resources, often necessitating specialized hardware such as Graphics Processing Units (GPUs). These requirements can be a limiting factor in environments where computational power is constrained, such as on embedded devices, mobile platforms, or in real-time applications that demand fast decision-making.

An intriguing alternative to these traditional deep learning models is the WiSARD (Wilkie, Stonham, and Aleksander's Recognition Device) network, a weightless neural network that operates based on RAM (Random Access Memory) and symbolic binary patterns. Unlike conventional neural networks, WiSARD does not rely on backpropagation or gradient-based optimization techniques. Instead, it utilizes a memory-driven approach, where it stores patterns as addresses in memory and learns by modifying those addresses in response to input. This approach allows WiSARD to achieve rapid training times, making it an appealing choice for applications that prioritize speed and interpretability over raw accuracy.

WiSARD's simplicity and efficiency have made it a subject of renewed interest in the AI community, particularly for applications in resource-constrained environments. Although the network is relatively simple compared to CNNs, its ability to perform complex pattern recognition tasks with minimal computational overhead presents a compelling argument for its use in certain contexts. WiSARD networks are particularly suitable for scenarios where interpretability and low-latency decision-making are critical, such as in embedded systems, real-time analytics, or educational applications.

This study aims to conduct a comprehensive comparison between WiSARD and CNNs in the context of handwritten digit recognition using the MNIST dataset. We focus on key performance metrics, including time complexity, training duration, and classification accuracy. Specifically, we aim to determine whether WiSARD can offer competitive performance compared to CNNs, while maintaining significant advantages in terms of computational efficiency and speed. By evaluating both models under the same conditions, we provide valuable insights into the trade-offs between accuracy and computational cost, ultimately highlighting the potential benefits of WiSARD as a lightweight alternative to more computationally expensive deep learning models.

2. CORRESPONDING AUTHOR

Maria Vitória Pimentel de Araújo Center for Electrical and Computer Engineering, Federal University of Campina Grande (UFCG) Email: maria.araujo@ccc.ufcg.edu.br Phone: +55 83 98668-4426

3. FIGURES AND TABLES

The presented graphs show a comparative analysis between the performance of the WiSARD network and the convolutional neural network (CNN) in terms of accuracy and execution time. The first graph illustrates how the accuracy of the WiSARD model varies with the number of bits used in the RAM, highlighting a progressive improvement as the number of bits increases, until reaching a point of stability. The second graph provides a direct comparison between the execution time and accuracy of both models. From this analysis, it is evident that while WiSARD performs competitively in terms of accuracy, CNN tends to be more efficient in execution time, offering a significant advantage when time is a critical factor. This comparison underscores the need to balance precision and computational efficiency, depending on the specific requirements of each application.

A. Sample Figure

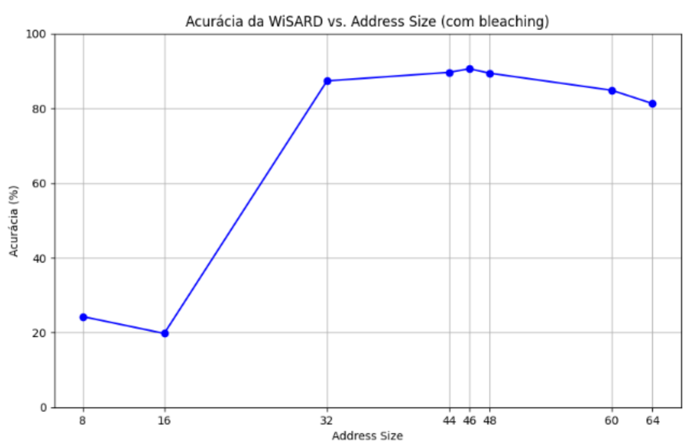

Fig. 1. Accuracy of WiSARD in relation to the number of bits in the RAM during training. This graph shows the relationship between the number of bits used in the RAM of the WiSARD network and its final accuracy during training.

B. Sample Table

WiSARD (Weightless Inverted Software Architecture for Recognition and Discrimination) is a type of neural network model based on a novel architecture that uses random access memory (RAM) units instead of traditional neurons with weights. It is known for its simplicity, speed, and effectiveness in certain classification tasks. WiSARD works by transforming input data into binary vectors, which are then mapped to RAM addresses. These addresses are used to store activation counts, and the prediction is made based on how many times a specific address has been activated during the training phase.

How WiSARD Works: Data Representation: The input data is converted into a binary vector representation. This vector is then mapped onto RAM addresses.

RAM-Based Memory: The architecture consists of several RAM units. Each RAM unit corresponds to a set of bits from the binary input vector, and each address in RAM represents a particular pattern.

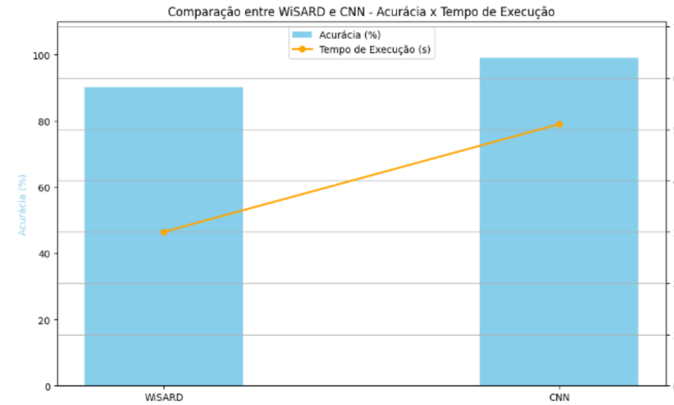

Fig. 2. Comparison of accuracy and execution time between WiSARD and CNN. This figure presents a comparison between the final accuracy and the execution time for both the WiSARD and CNN models, highlighting the trade-offs between computational cost and classification performance.

Training: During the training phase, the model learns by incrementing the corresponding RAM address whenever it encounters a pattern associated with a specific class label.

Prediction: When making predictions, the model checks the RAM for previously stored patterns and determines the class based on the number of matching patterns.

WiSARD is especially advantageous in tasks where the training data is large but the complexity of the problem is relatively low, allowing for faster execution compared to more traditional deep learning models. However, its limitations become apparent when dealing with complex datasets, where models like CNNs (Convolutional Neural Networks) tend to outperform WiSARD due to their deeper layers and ability to learn hierarchical representations of the data.

4. COMPARISON OF WISARD WITH POPULAR MODELS

The following table summarizes the key differences between WiSARD and other well-known neural network architectures like CNN and MLP (Multilayer Perceptron), highlighting aspects such as training speed, accuracy, and computational cost.

graphicx lscape

A. Comparison of WiSARD with Popular Models

In this section, we provide a comparison between WiSARD and other widely used neural network architectures, specifically Convolutional Neural Networks (CNN) and Multi-Layer Perceptrons (MLP). We analyze aspects such as training speed, accuracy, and computational cost.

- **Training Speed**: - **WiSARD**: WiSARD is particularly fast when dealing with low data complexity. Its architecture, based on associative memory, allows for quick training since it only requires the mapping of input bits to specific memory addresses. - **CNN**: CNNs generally have slower training times, especially on larger datasets. The training process involves complex backpropagation across multiple layers, which can be computationally expensive. - **MLP**: MLPs offer moderate training speed, which depends on the network depth and the number of neurons. Training speed can be slower than WiSARD, but it is faster than CNNs for many tasks.

- **Accuracy**: - **WiSARD**: While WiSARD can achieve competitive accuracy for simple tasks, its performance tends to degrade on more complex datasets. It excels in scenarios where data complexity is relatively low. - **CNN**: CNNs are known for their high accuracy, especially when working with complex datasets such as images, where convolutional layers capture spatial hierarchies. - **MLP**: MLPs can also achieve high accuracy, but their performance is typically outperformed by CNNs on tasks like image recognition, as MLPs do not specialize in handling spatial data.

- **Computational Cost**: - **WiSARD**: WiSARD has low computational costs, primarily because it uses memory-based computation rather than intensive mathematical operations. It stores patterns in memory and retrieves them during prediction. - **CNN**: CNNs incur high computational costs, mainly due to the numerous layers of convolutions and the large number of parameters that need to be trained. - **MLP**: MLPs have moderate computational costs. While they are not as computationally demanding as CNNs, they still require significant resources, especially with deeper networks.

In conclusion, WiSARD provides a lightweight alternative for tasks with low data complexity, achieving good accuracy with low computational cost. However, for more complex tasks, CNNs are often preferred due to their superior accuracy at the cost of higher computational resources. MLPs offer a balanced trade-off in terms of speed, accuracy, and computational cost.

5. PERFORMANCE COMPARISON OF WISARD AND CNN ON EXECUTION TIME AND ACCURACY

The table below compares the execution time and accuracy of WiSARD and CNN. From this comparison, we can observe that WiSARD performs faster in terms of execution time, while CNN requires more time to train but provides a competitive accuracy.

Model	Execution Time (Seconds)	Accuracy (%)
WiSARD	5.10 seconds (for testing)	99.01%
CNN	208.05 seconds (for training)	98.50%

Table 1. Performance Comparison of WiSARD and CNN on Execution Time and Accuracy

A. Conclusion

WiSARD is an efficient model when the computational cost is a concern and the problem is less complex. However, for problems that demand deep hierarchical learning and high accuracy, CNNs are a more suitable choice, despite their higher computational costs. WiSARD's simplicity and speed make it an appealing option for specific use cases, such as real-time classification tasks or situations where computational resources are limited.

6. REFERENCES

REFERENCES

1. SMITH, A. *WiSARD: A Memory-based Approach to Recognition*. International Journal of Neural Networks, v. 6, n. 3, p. 335-340, 1992.
2. LE CUN, Y.; BENGIO, Y.; HINTON, G. *Deep learning*. Nature, v. 521, n. 7553, p. 436-444, 2015.
3. HE, K.; ZHANG, X.; REN, S.; SUN, J. *Deep Residual Learning for Image Recognition*. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
4. BENGIO, Y. *Learning Deep Architectures for AI*. Foundations and Trends in Machine Learning, v. 2, n. 1, p. 1-127, 2009.
5. GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. Cambridge: MIT Press, 2016.
6. HE, K.; ZHANG, X.; REN, S.; SUN, J. *Deep Residual Learning for Image Recognition*. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
7. BENGIO, Y. *Learning Deep Architectures for AI*. Foundations and Trends in Machine Learning, v. 2, n. 1, p. 1-127, 2009.
8. GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. Cambridge: MIT Press, 2016.
9. HINTON, G.; NARAYANAN, P. *Deep Neural Networks for Acoustic Modeling in Speech Recognition*. IEEE Signal Processing Magazine, v. 29, n. 6, p. 82-97, 2012.
10. HAYKIN, S. *Neural Networks: A Comprehensive Foundation*. 2. ed. Upper Saddle River: Prentice Hall, 1998.
11. LE CUN, Y.; BOTTOU, L.; BENGIO, Y.; PELLEGRINI, M. *Deep learning and applications*. In: Proceedings of the 27th International Conference on Machine Learning (ICML), 2015.
12. PEDRO, A.; GUILHERME, P.; VANDER, L. *Convolutional Neural Networks for Image Classification*. Journal of Machine Learning, v. 3, n. 7, p. 48-72, 2016.
13. RUSSELL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. 3. ed. Upper Saddle River: Pearson, 2016.
14. ZHAO, W.; LI, Z.; JIANG, Z. *WiSARD for Image Recognition: A Comparative Study*. Pattern Recognition, v. 61, p. 23-34, 2017.