Dean&Francis
ISSN 2959-6157

Hand-Written Digit Classification Using Hardware-Implemented Neural Networks: Design, Optimization, and Challenges

Minhao Shi

Zhejiang University-University of Illinois Urbana-Champaign Institute, Zhejiang University, Haining, China

minhaos3@illinois.edu

Abstract:
This study introduces a neural network-based system designed to expedite the classification of handwritten characters, particularly focusing on numbers. Utilizing a perceptron that aggregates weighted inputs followed by a non-linear function, the system estimates the likelihood of a character being any digit from 0 to 9. The implementation, achieved through Very-Large-Scale Integration (VLSI) and programmed using Verilog, adopts a tree structure that saves over one thousand clock cycles compared to a serial approach in the addition phase. It requires 393 clock cycles for the system to accurately recognize a single digit. This research presents a practical approach to handwritten character classification. The system has potential applications across various handwriting resources, including images, paper documents, and touchscreen inputs. Further studies could broaden its utility by extending recognition capabilities to letters and symbols, enhancing its applicability in educational and scientific domains where quick and precise character recognition is crucial. Such advancements could significantly benefit sectors requiring efficient data digitization and processing from handwritten sources.

Keywords: Handwritten number classification; neural network; perceptron; tree; very-large-scale integration.

1. Introduction

In an era increasingly dominated by digital communication and data processing, the ability to quickly and accurately convert handwritten input into digital format has become paramount. With the rise of internet-based technologies, handwritten documents, whether educational materials, personal notes, or business records, need to be digitized efficiently. This necessity extends beyond mere convenience; it underpins crucial processes in global communication, data management, and archival work. The challenge lies not only in recognizing clearly written text but also in deciphering varied handwriting styles and symbols that conventional optical character recognition systems often struggle with.

The advent of neural networks has revolutionized many aspects of text recognition, providing a foundation for systems that learn from vast amounts of data to improve their accuracy over time. Traditional methods, reliant on predefined algorithms and patterns, falter with the inherent variability of human handwriting. Neural networks, however, leverage their learning capability to adapt and interpret a wide range of handwriting styles. The application of these networks in tasks such as the transformation function on writing pads or screens, where users employ capacitive pens to input characters, illustrates the shift towards more dynamic and adaptable recognition systems. These systems not only recognize but also standardize characters, offering personalized recognition tailored to individual handwriting quirks, thereby enhancing user interaction and accessibility [1, 2].

This paper introduces a neural network-based system specifically designed for the classification of handwritten numbers and characters using a perceptron, a fundamental component of deep learning. Implemented through Very-Large-Scale Integration (VLSI) and coded in Verilog, this system marks a significant stride in the field. The perceptron module weighs input signals and processes them through a nonlinear sigmoid function, enhancing the ability to decode complex character data into standardized digital forms [3]. The system design utilizes a tree structure to optimize the classification process, drastically reducing the number of clock cycles needed for computation—from 1737 to just 393, achieving a 77.3% improvement in efficiency [4]. This optimization not only accelerates processing but also reduces the overall cost associated with handwritten character recognition, underscoring the utility of integrating computer engineering principles with neural network technology. The resulting system exemplifies the potential of multidisciplinary approaches in addressing complex technological challenges, setting a new benchmark for future research in the domain.

2. Relevant Theories

2.1 Neural Network Algorithms

Neural networks are the core of the self-learning and weighing part. It is the steppingstone for the latter classification. Creating a handwritten digit classification use anyone of the following algorithms, but there will be some pros and cons.

Linear Models: Including linear regression model, ANOVA model, applied in biology, medicine, economy, management. It is usually tested in a univariate manner, so it is not suitable for large data integration training.

Artificial neural network: Artificial neural network is a nonlinear and adaptive information processing system composed of a large number of interconnected processing units. It tries to process information by simulating the way that the brain neural network processes and memorizes information.

KNN(K-Nearest-Neighbor): If most of the K nearest neighbors of a sample in the feature space belong to a certain class, the sample also belongs to this class. It is often used to train models with absolutely vast database.

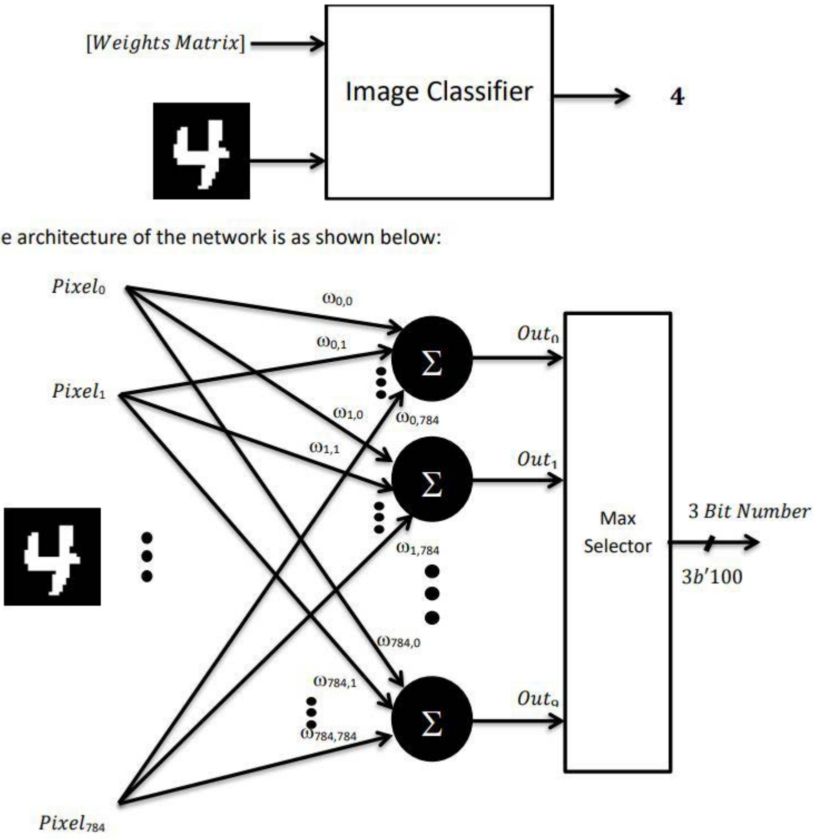

Fig. 1 Perceptron and max selector (Photo credit: Original).

This study uses perceptron, which is a popular deep learning neuron model aimed at actuating the calculation into practical hardware circuits. Perceptron is summing up the multiplication of the input with the weight and transport the sum into the non-linear sigmoid function. In the input layer, users should prepare input sets and an original weight set ranging through all real numbers. Weights values will be modified toward the one that recognize the handwriting inputs best through the process of feedback [5]. Here is the Backpropagation Algorithm (BP algorithm), it enables perceptron to gradually learn to classify the numbers more and more accurately step by step.

The formula for a single perceptron should be: out = sigmoid [wx + b].

Here ‘w’ stands for the weights, ‘b’ stands for the tendency or the bias, ‘x’ is the input. So the processing times of the out function is based on the pixel accuracy of the input picture (expansion of input set). As show in the fig. 1.

2.2 Classification Methods

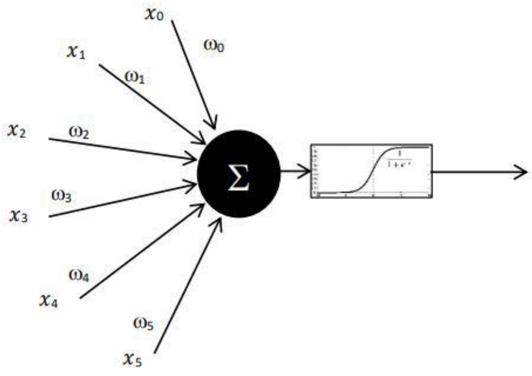

Fig. 2 Sigmoid (Photo credit: Original).

As show in the fig. 2. Since the system needs to recognize numbers from zero to ninety, ten perceptrons are needed to compute in parallel the ten possibilities that the input image is these ten numbers. In this study, the input set is usually 28x28 images, so a total of 784-pixel channels of information are fed into the classification system [6]. Each pixel is displayed on a gray scale and has 0 to 255 gray levels. The pixel will be processed by the perceptron to get ten unnormalized values. These are the values that will be fed into the sigmoid and normalized. After the standardization, ten values ranging from 0 to 1 will be put into the max selector and the channel with the greatest possibility is the number of the input picture. The final output is represented in 4 bits binary. For example, ten outputs of the sigmoid function are [Out0 Out1 Out2 Out3 Out4 Out5 Out6 Out7 Out8 Out9] = [0.1, 0.2, 0.3, 0.44, 0.88, 0.15, 0.2, 0.33, 0.6, 0.76], then the greatest number is 0.88, thus, the picture is probably "4" and the final output in binary will be 4b'0100.

3. Circuit Design

3.1 Introduction to Verilog and Hardware Design

Verilog supports concurrent programming and it uses this ability to simulate hardware operations. It breaks code into small components that allows researchers verify their design and has a thorough process of building a circuit that make sure the final simulation can be reproduced [7]. This study basically requires multipliers, adders to finish the circuit.

3.2 Basic Design Without Optimization

The simplest way to design a perceptron is to first multiply all the single input with their corresponding weight serially, then add them up serially as well. Thus, this serial computation will have 10*784 multipliers and 10 *783 adders, which is area consuming [8]. In the time domain, one line of the picture (28 pixels) will need 2*28 clock cycles for addition and 6 clock cycles for multiplication. In total, for a 28 lines picture, it requires 62*28 clock cycles, which is 1736 clock cycles. Adding the final 1 clock cycle consumed by the max selector, the running time has reached 1737clock cycles. This is absolutely time consuming. Due to the application of serial computation, the computing system has become so bloated, complex and expensive that better way should be applied.

3.3 Optimization Techniques

Binary Addition Tree (BAT) is an accelerating way for the frequent addition operation in this classification system. Abd's research proved the effectiveness of the Binary Multiplication based on applying different structures of tree [9]. So this study expended it into the addition operation and eventually excelled the recognition. Imagine a two-layer tree, it has one father leaf and two son leaves, which adds up to be three times of efficient addition. Similarly, for a four-layer tree, it will own 15 valid addition operations. What if the tree is mirrored and grafted on one final adder along with another four-layer tree. In total this four-layer addition system can have over 27 addition times, which is what a one line add operation times for a 28 pixels line. As show in the fig. 3.

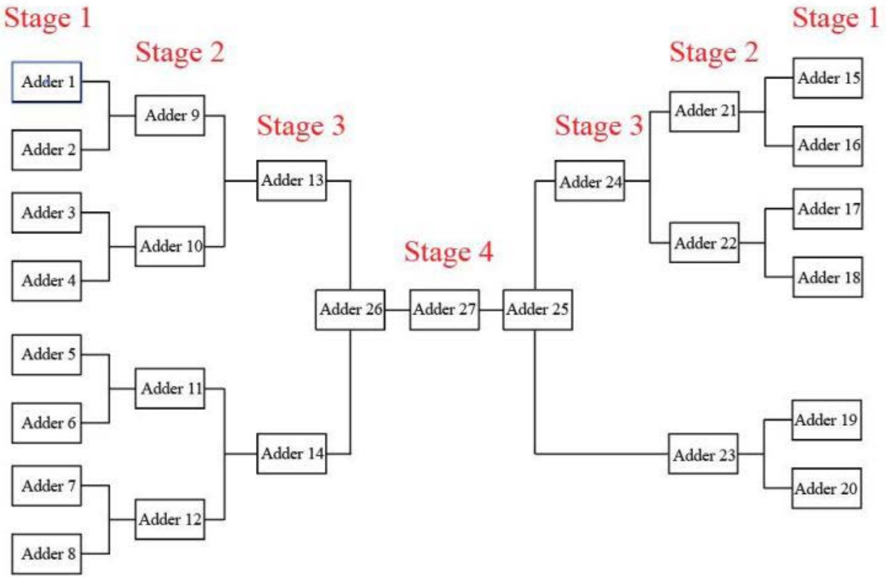

Fig. 3 BAT (Photo credit: Original).

3.4 System After Optimization

After introducing the BAT, all the addition operations are compressed by BAT, the original 27 addition operations only need to be performed five times in parallel. This greatly shortens the computing time, makes the whole system cheaper to use, and makes the system failure rate and debug difficulty greatly reduced. The computation time is thus greatly reduced to 2*4 clock cycles for addition operation, same 6 clock cycles for multiplication operations because the multiplier is still adding two 0- 255 bit numbers. For a 28*28 picture inputing 784 numbers, adding the 1 clock cycle of max selector, it only needs 393 clock cycles for a series of operations. The operation is greatly speeded up.

3.5 Results

This study shows the effectiveness of BAT optimization, which speedups the operations by 77.3%. Along with Sara's research, both show artificial intelligence neural network has advantages in the recognition and processing of handwritten images. All ten numbers from 0 to 9 can be recognized using this BAT optimized calculation flow cheaply and effectively.

4. Challenges

Implementing the Binary Addition Tree (BAT) within a neural network-based system for handwritten digit classification presents several significant challenges, primarily due to the complexities of hardware programming and optimization. One of the primary difficulties encountered in this research is integrating the BAT design with Verilog, a hardware description language that requires meticulous precision in simulation and coding. The adaptation of BAT, particularly when the additions are not a power of two, necessitates frequent adjustments and fine-tuning to ensure accurate operation and efficient computation. Furthermore, the optimization of the tree structure within the perceptron-based system involves handling a large amount of data input, which can be both time-consuming and resource-intensive. This process demands an extensive verification phase to confirm that all components are functioning correctly in parallel, without data loss or errors, which are common in complex digital circuit designs. The challenge is exacerbated by the need to maintain system integrity and reliability, especially when scaling up the architecture to handle more extensive sets of handwritten data [10]. Moreover, the integration of a non-linear sigmoid function in a hardware-based neural network requires the effective handling of floating-point arithmetic, which is inherently challenging in FPGA and VLSI environments. These environments typically favor fixed-point calculations, adding another layer of complexity to the system design and necessitating additional conversion mechanisms or approximations, which must be precisely managed to avoid degradation of system performance and accuracy. These challenges underline the need for a robust design strategy and highlight the importance of ongoing research to refine these technologies, ensuring they meet the rigorous demands of real-world applications. The iterative process of testing, optimization, and adjustment is crucial to overcoming these hurdles and achieving a reliable and efficient classification system.

5. Conclusion

This study has successfully demonstrated a significant advancement in the field of handwritten digit classification through the implementation of a neural network-based system using a tree structure for optimized processing. By integrating perceptron modules within a Very-Large-Scale Integration framework coded in Verilog, the research achieved a drastic reduction in computation time, lowering it from 1737 clock cycles to just 393. This represents a 77.3% improvement in efficiency, marking a substantial leap forward in the digital processing of handwritten characters. The application of Binary Addition Tree methodology proved instrumental in this enhancement, showcasing the power of combining traditional data structures with modern neural network architectures for high-speed, cost-effective solutions in digital text recognition. Looking to the future, several avenues appear promising for extending the capabilities and applications of this technology. The next phase of research could focus on refining the input data preprocessing to enhance the system's accuracy and efficiency further. By implementing normalization techniques and noise reduction algorithms, the system could be better tailored to handle variations in handwriting and environmental conditions that affect input quality. Additionally, exploring pipelined designs may offer avenues for even faster processing times and more efficient resource utilization. Such advancements could broaden the scope of applications, extending beyond digit recognition to encompass full alphanumeric and symbolic character sets, which would be invaluable in more complex text recognition scenarios across various digital platforms. These future directions not only promise to refine the existing framework but also open up new possibilities for the integration of this technology into broader, real-world applications, reinforcing the interdisciplinary potential of this research.

References

[1] Abdullah M, Agal A, Alharthi M, Alrashidi M 2018 Retracted: Arabic handwriting recognition using neural network classifier Journal of Fundamental and Applied Sciences 10(4S) 265-270.

ISSN 2959-6157

[2] Abe S 2010 Support Vector Machines for Pattern Classification Berlin, Germany: Springer Science & Business Media.

[3] Zhang Y, Zhao H, Zhu X, Zhao Z, Zuo J 2019 Strain Measurement Quantization Technology based on DAS System 2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC) 214-218.

[4] He K, Zhang X, Ren S, Sun J 2016 Deep residual learning for image recognition Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 770-778.

[5] Aqab S, Tariq MU 2020 Handwriting recognition using artificial intelligence neural network and image processing International Journal of Advanced Computer Science and Applications 11(7).

[6] Zhang Yuan, Wang Shuwang, Ma Yongbing 2021 Comparison of the application of machine and deep learning algorithms in handwritten digit recognition Informatization Research 5 60-64.

[7] Caplar R, Kulisic P 1973 Proc. Int. Conf. on Nuclear Physics (Munich) vol 1 (Amsterdam: North-Holland/American Elsevier) 517.

[8] Al-Nounou AAK, Ar-Knaleel U, Obeidat F, Al-Knaleel M 2022 FPGA implementation of fast binary multiplication based on customized basic cells Journal of Universal Computer Science 28(10) 1030-1057.

[9] Wang R, Zhu J, Wang S, Wang T, Huang J, Zhu X 2024 Multi-modal emotion recognition using tensor decomposition fusion and self-supervised multi-tasking International Journal of Multimedia Information Retrieval 13(4) 39.

[10] Giardino D, Matta M, Silvestri F, Spanò S, Trobiani V 2019 FPGA implementation of hand-written number recognition based on CNN International Journal on Advanced Science, Engineering and Information Technology 9(1) 167-171.