Article

Probabilistic Classification Method of Spiking Neural Network Based on Multi-Labeling of Neurons

Mingyu Sung [1], Jaesoo Kim [2] and Jae-Mo Kang [1,*]

[1]   Department of Artificial Intelligence, Kyungpook National University, Daegu 41566, Republic of Korea
[2]   Department of Computer Science, Kyungpook National University, Daegu 41566, Republic of Korea
*   Correspondence: jmkang@knu.ac.kr

Abstract: Recently, deep learning has exhibited outstanding performance in various fields. Even though artificial intelligence achieves excellent performance, the amount of energy required for computations has increased with its development. Hence, the need for a new energy-efficient computer architecture has emerged, which further leads us to the neuromorphic computer. Although neuromorphic computing exhibits several advantages, such as low-power parallelism, it exhibits lower accuracy than deep learning. Therefore, the major challenge is to improve the accuracy while maintaining the neuromorphic computing-specific energy efficiency. In this paper, we propose a novel method of the inference process that considers the probability that after completing the learning process, a neuron can react to multiple target labels. Our proposed method can achieve improved accuracy while maintaining the hardware-friendly, low-power-parallel processing characteristics of a neuromorphic processor. Furthermore, this method converts the spike counts occurring in the learning process into probabilities. The inference process is conducted to implement the interaction between neurons by considering all the spikes that occur. The inferring circuit is expected to show a significant reduction in hardware cost and can afford an algorithm exhibiting a competitive computing performance.

Keywords: leaky integrate fire neuron; spiking neural network; spike-timing-dependent plasticity

MSC: 68Q07

1. Introduction

Deep learning using artificial neural networks (ANNs) is showing outstanding performance in various fields, such as image recognition, classification, object recognition, speech recognition, and medical diagnosis [1,2]. Given its excellent performance in relation to the considerable development in artificial intelligence, the computational energy requirement for deep learning has increased proportionally. Therefore, research into the overhead lightening of deep learning is becoming active. However, deep learning has inherent limitations because it requires an enormous amount of computation. Moreover, its high energy consumption adversely affects the use of artificial intelligence. For instance, it is impossible to utilize deep learning operations locally on devices with minimal battery capacities. Therefore, many researchers are opting to use communication to take the results from cloud servers. This method is not only unstable in function but also causes a lot of waste in terms of energy. The small cost and small area chipsets are essential so that artificial intelligence can be operated locally.

To tackle the issue, researchers have been seeking a new computer architecture to efficiently perform the tasks. The human brain can be a good motivation for a new architecture since it offers an excellent example of a high-performance processor. The core characteristic of the human brain is event-driven architecture. Unlike general digital devices, there is no clock. Since the operation is processed only when an event occurs, the

energy loss can be extremely reduced. This outstanding energy efficiency and parallelism of the human brain show great potential as a new computer architecture. Neuromorphic processor is one of the leading candidates for solving these problems. Nowadays, this processor has been extensively studied as a low-power, high-efficiency processor. It imitates the low-power, small-volume, high-efficiency characteristics of the human brain to develop a dedicated hardware chipset for artificial intelligence different from the Von Neumann structure [3]. Moreover, a neuromorphic processor that adopts the parallelism and energy efficiency of the human brain can deal with the aforementioned disadvantages of deep learning in the Von Neumann structure. Therefore, numerous studies are actively conducted to design a neuromorphic processor that consumes extremely low power and hardware overhead by mimicking the biological brain.

This paper focuses on spiking neural networks (SNNs), which imitate the interaction and learning mechanisms of the human brain's neuronal spikes. While SNNs offer high parallelism and low power consumption, their accuracy is often lower than other types of neural networks. In response, this paper proposes algorithms that increase SNN accuracy with less hardware overhead. Additionally, the paper presents a method for designing SNN algorithms as hardware circuits [4,5]. To improve accuracy, the paper introduces a normalizing method used in deep learning that has been proven effective but which is typically expensive to implement a circuit. To reduce cost, the paper proposes an approximate skipping method that can be implemented as a hardware circuit. This method performs complex operations while achieving high efficiency, reducing power consumption and area while increasing accuracy. In this study, we have focused on improving the accuracy of neuromorphic systems while maintaining energy efficiency. To demonstrate the practical significance of our work, we conducted a comprehensive literature review and analyzed the importance of accuracy and energy efficiency in various domains, including power systems, renewable energy, and biomedical engineering. Energy efficiency is crucial in the field of power systems for reducing greenhouse gas emissions and meeting energy demand. Yang et al. [6] reported that the use of intelligent algorithms and machine learning techniques can improve the energy efficiency of power systems. Our proposed method can contribute to this effort by reducing the energy consumption of neuromorphic systems, which can be used in various applications, such as fault detection and classification. Furthermore, recent advances in neuromorphic computing have highlighted the need for developing brain-inspired systems that can go beyond the traditional Von Neumann architecture. One critical component of these systems is the ability to emulate biological synapses. In this regard, John et al. [7] demonstrated the development of artificial synapses based on ionic–electronic hybrid oxide-based transistors on both rigid and flexible substrates. The flexible transistors reported in the study achieved a high field-effect mobility of approximately $9\,\mathrm{cm^2\,V^{-1}\,s^{-1}}$, while maintaining good mechanical performance. The study also successfully established comprehensive learning abilities/synaptic rules, such as paired-pulse facilitation, excitatory and inhibitory postsynaptic currents, spike-time-dependent plasticity, consolidation, superlinear amplification, and dynamic logic. These functionalities enable concurrent processing and memory with spatiotemporal correlation. These results represent a significant breakthrough in developing fully solution-processable approaches to fabricate artificial synapses for next-generation transparent neural circuits. By incorporating such novel synapses in our proposed neuromorphic system, we can achieve high accuracy while maintaining energy efficiency, making it ideal for various real-world applications, such as those discussed in previous works [6–8]. In summary, our proposed method has practical significance for real-world applications in various domains, such as power systems, renewable energy, and biomedical engineering. By improving the accuracy of neuromorphic systems while maintaining energy efficiency, our method can contribute to the development of more efficient and reliable systems in these domains.

Section 2 of this paper explains the background theory of neuromorphic computing. Section 3 proposes a method that applies probabilistic classification to the inference process.

We explained the efficient design by implementing an appropriate formula for the hardware circuit. Section 4 demonstrates the experimental results and the analysis. We compared the accuracy, the area of the circuit, energy, delay, energy per accuracy, and improved effectiveness.

2. Related Work and Motivation

The SNN architecture used in this article was proposed by Diehl et al. [9]. Figure 1 shows the overall network architecture. The network was designed to employ the unsupervised spike-timing-dependent plasticity (STDP) learning rule [10] to train the MNIST dataset and classify handwritten digits. The SNN model employs a multi-layer architecture that includes input, excitatory, and inhibitory layers. The input layer comprises 784 neurons, and each neuron corresponds to one pixel of an image. Note that an MNIST image consists of $28 \times 28 = 784$ pixels.
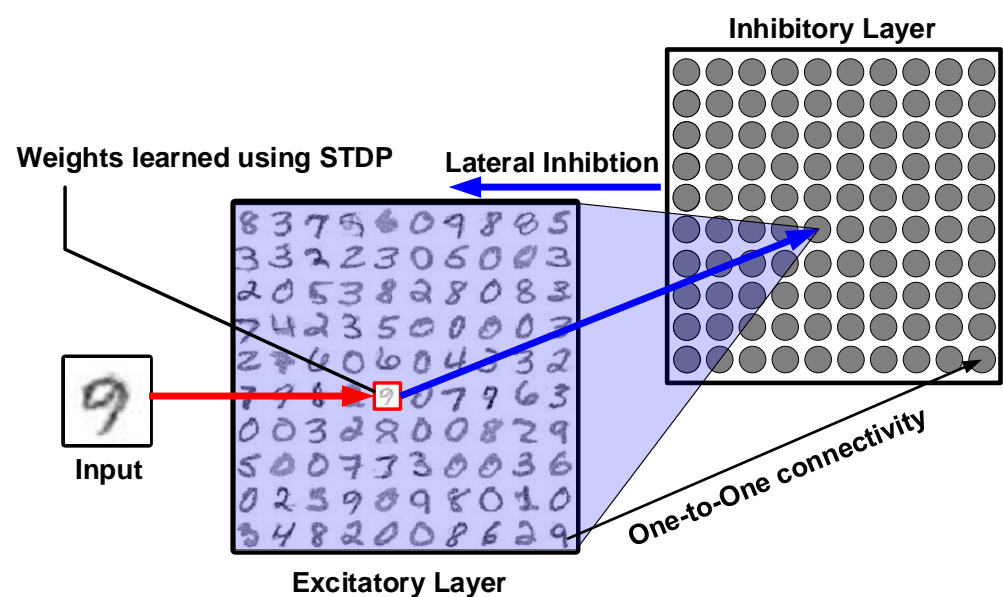

Figure 1. Spiking neural network training phases.

For presenting the digits from the MNIST dataset to the network for training, the input pixels are converted to a Poisson-distributed spike train with a rate proportional to the pixel intensity. The excitatory and inhibitory layers include the same number of neurons. A one-to-one neural connection is established between the excitatory and inhibitory layers [11]. Alternately, a neural connection is ensured between each neuron in the inhibitory layer and all neurons in the excitatory layer except for those connected through the excitatory layer. This network connectivity offers a winner-take-all (WTA) mechanism by implementing lateral inhibitions that compete in the excitatory layer [9,12]. Figure 2 shows that the network includes three phases: the training phase, neuron labeling phase, and inference phase. (1) Training phase: Neurons insert spike trains as input and learn data using the STDP learning rule described previously. (2) Neuron labeling phase: This phase assigns the label to each neuron based on the highest response for label classes. The spike counts generated at this phase are called neuron labeling spike counts. (3) Inference phase: This phase measures the classification accuracy of the network. The sum of the classes assigned in the neuron labeling phase of firing neurons becomes the final inference. This is the only step where labels are used. This hardware-friendly, energy-efficient, parallel processing SNN architecture can be implemented as a neuromorphic processor; however, it possesses the disadvantage of low accuracy [9]. The lack of accuracy in these neuromorphic circuits is challenging. Therefore, in this paper, we propose

a method to achieve enhanced accuracy while maintaining the aforementioned advantages of the neuromorphic process.
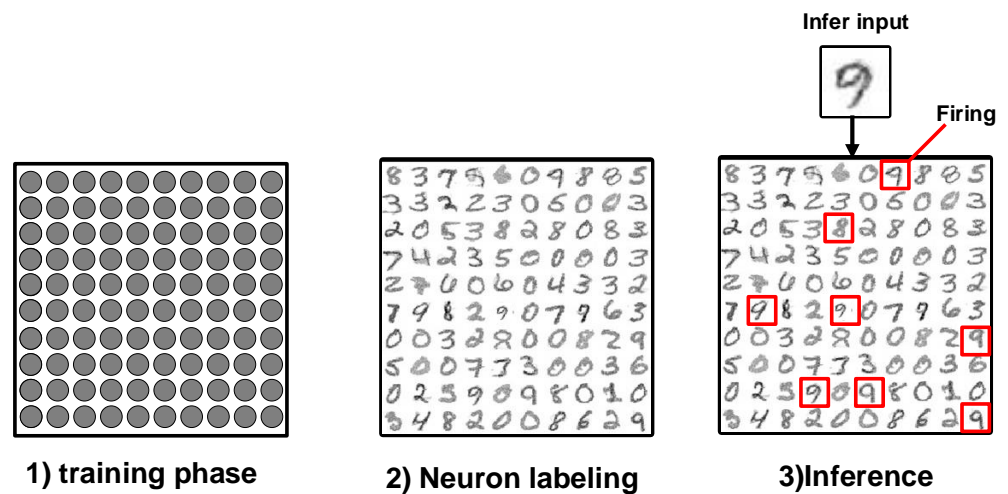

Figure 2. Spiking neural network's training phases.

The baseline SNN in Diehl et al. [9] includes 400 excitatory and inhibitory neurons, and the network is subsequently trained and tested with 60,000 training datasets and 10,000 test datasets, respectively. Figure 3 presents the label for which neurons responded most actively at the neuron labeling phase after training. Therefore, it shows the label to which each neuron exhibits the most active responsiveness. In this study, ten iterations of the training process were performed, and an average number of incorrectly classified cases were obtained. Error patterns of the classification of one specific digit as another were examined. For instance, images of digit 4 are classified as digit 9, and vice versa. This pattern is referred to as swapped classification.
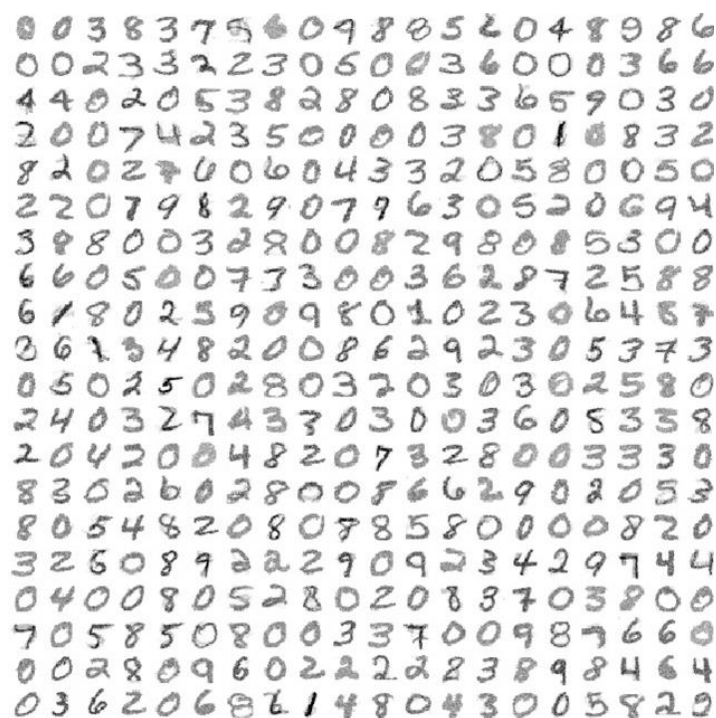

Figure 3. Label, for which each neuron reacts most actively.

Figure 4 demonstrates some of the most frequently swapped classification error cases. From Figure 4, we can observe that 1790 swapped classification errors are registered out of 10,000 tests, and almost half (854 errors) of them are concentrated on 4 out in 45 cases. The swapped classification errors between digits 4 and 9 occurred most frequently. Specifically, 305 (3.05%) swapped classification errors are detected between 4 and 9 among 10,000 tests. From Figure 4, 204 images of digit 4 are misclassified as 9, and 101 images of digit 9 are incorrectly classified as 4. Significant swapped classification errors do not occur except for cases of $4 \Leftrightarrow 9$, $7 \Leftrightarrow 9$, $3 \Leftrightarrow 5$, and $3 \Leftrightarrow 8$. In these error patterns, each neuron responding to test response spikes tends to exhibit a relatively high spike count of a specific label and another label, which considerably influences the error. In short, neurons that respond to these ambiguous error patterns exhibit a high rate of response to various labels in the neuron labeling phase. Therefore, neurons that are ambiguously trained intervene in ambiguity patterns with high probability. If these neurons cannot decide between the ambiguous spike counts, they should put the decision on hold until the next unambiguous trained neurons can deal with the decision. However, the decision-holding mechanism does not function in the current method because the similar high active count labels fail to contribute to the decision. Due to this problem, it becomes difficult to ignore the ambiguity in neurons in the neuron labeling phase. To tackle this issue, we proposed here a method estimating the probabilities of all labels for the trained neurons.
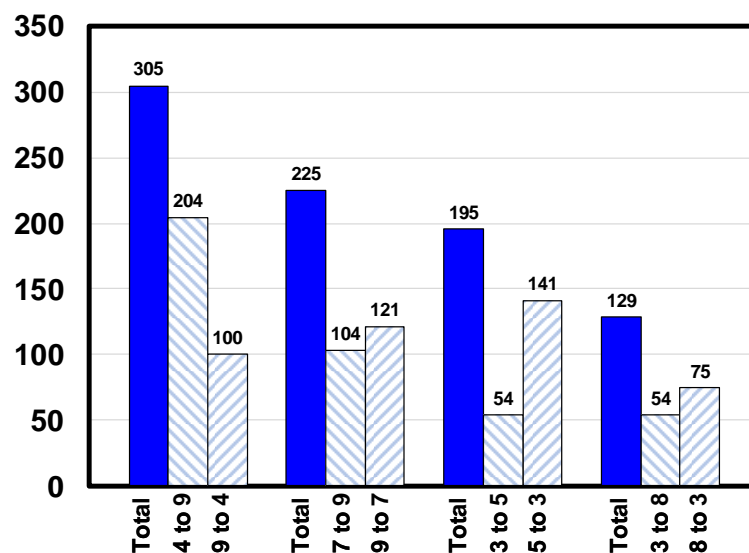

Figure 4. Top four swapped misclassification cases.

3. Proposed Approach

A neuron can respond to all ambiguous patterns. If only the largest spike count label is assigned, the training of that label with a similar high level of counts is discarded without contributing to the results. Therefore, in this section, we propose a probabilistic classification method that allows all spikes to contribute to learning. In addition, we propose a circuit design that implements the proposed method in a hardware-friendly manner using an approximation technique. Our method exhibits excellent performance across various indicators such as energy efficiency, area, delay and accuracy.

3.1. Probabilistic Classification

We present a novel probabilistic classification method that takes into account all spikes during learning. The approach leverages probability theory to model the probability of neurons responding to different labels, both during the neuron labeling phase and inference phase after training.

During inference, we generate a test example and measure the spike counts s_j of all neurons j. Let T be the total number of spikes generated by all neurons for this test example. We calculate the probability of each neuron responding to each label in the neuron labeling phase using the following formula:

$$p_j{}^i = \frac{s_j{}^i}{\sum_t s_j{}^t},\tag{1}$$

where $p_j{}^i$ is the probability of neuron j responding to label i. This equation computes the probability that a neuron will respond to a given label based on the spike counts of all labels. Thus, each neuron has probability values that reflect the likelihood of it responding to any label. We interpret Equation (1) as containing all the probabilities of a single neuron responding to any label. Here, T is the sum of all test response spike counts generated during inference, and s_j is the test response spike count generated by neuron j during inference. To deduce the final result, we aggregate the probability of all labels, denoted by N, that spiked during testing according to

$$\mathbf{y} = \max_i \sum_{j=0}^{N} \left(p_j^i \times \left(\frac{s_j}{T} \right) \right).\tag{2}$$

This equation works like a weighted voting ensemble in machine learning techniques, where a single neuron makes a decision on all labels and reflects the sum of them in the result.

In the inference phase, each neuron has a proportional influence on the number of spikes in the final result. We accumulate the probability of each label in proportion to the frequency of spike counts occurring in testing. Finally, we select the label with the highest spike probability. Our method achieves improved accuracy compared to other techniques.

Figure 5 presents an improved error in the proposed method, demonstrating its effectiveness in reducing errors for most patterns. Notably, it exhibits a considerable improvement for label 9, which had the highest error rate in the conventional method. However, we observed performance degradation for label 5, which is distinct from the other digits in the dataset in terms of its shape and orientation. This may suggest that the proposed method is less suitable for certain patterns that deviate significantly from the characteristics of the training data. Furthermore, the proposed method has certain limitations and assumptions that may affect its performance on certain patterns, such as the use of unsupervised STDP learning and the assumption of independent probabilities for each label. Future research could explore methods to overcome these limitations and improve the overall performance of the proposed method.
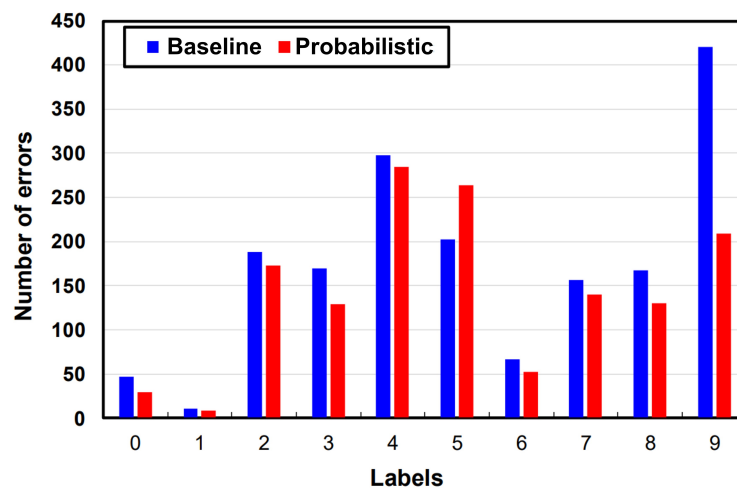

Figure 5. Error improved of each label.

3.2. Classification Using Quantization

The hardware-optimized method is implemented through quantifying the spike count and skipping the insignificant spikes. The method benefits hardware design and exhibits minor accuracy loss. First, among all neuron labeling spike counts, the highest digit position of 1 is found, and the bit becomes the most significant bit (MSB). The same MSB position is applied to other labels. Thereafter, starting with the MSB, we only read the bits up to a specific range of valid digits. Noting that N bits are valid, the count value is normalized from 0 to $2^N - 1$. This circuit design has a mathematical meaning similar to Equation (1). The precision decreases by limiting the effective range, and the number of bits required for the operation can be reduced.

4. Experimental Results

Figure 6 shows an example of 3-bit range neuron labeling. When three different labels are assigned, "A" spike counts are reduced from 412 to 6, "B" spike counts are reduced from 160 to 2, and "C" spike counts are reduced from 81 to 1. There is a large deviation in the assigned spike count for each neuron. This approximation technique minimizes the deviation of set spike counts through normalizing all neurons to uniform range spike counts. Spike counts that are sufficiently small for exclusion from consideration are ignored as zeros in this process.
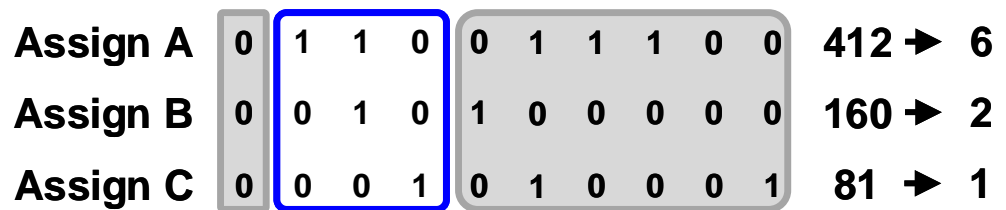

Figure 6. Normalization example of 3 label's spike counts.

Figure 7 demonstrates typical hardware architecture. The neuron labeling circuit diagram shows counters, one OR gate, and one leading one position detectors (LOPD) circuit [13]. In the hardware circuit design, we implement the proposed method by performing OR gate to spike counts of all labels. Subsequently, we find the highest digit position of 1 through LOPD. Note that the LOPD circuit is used to determine the MSB with the highest position of numbers. This circuit determines the number corresponding to the highest position of the MSB in all labels' assignments. As a result of the LOPD, certain shifters shift the assignment counters to the left. After the shift, the circuit only reads the specific part of the output from the MSB to N-bit. Only the range from the number of digits found by LOPD to the number of significant digits is applied effectively. Figure 8 shows how to acquire three significant digits from a 10-bit counter. Since each neuron has the same range of count values, it prevents a specific neuron from intervening in the inference of the result. This process prevents the neuron from being responsible for inferring when the spike count is excessively high.
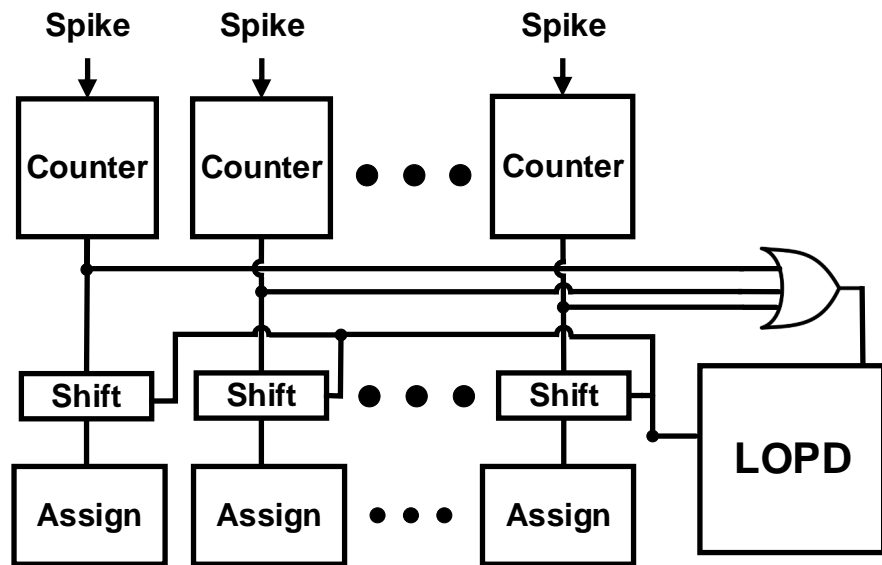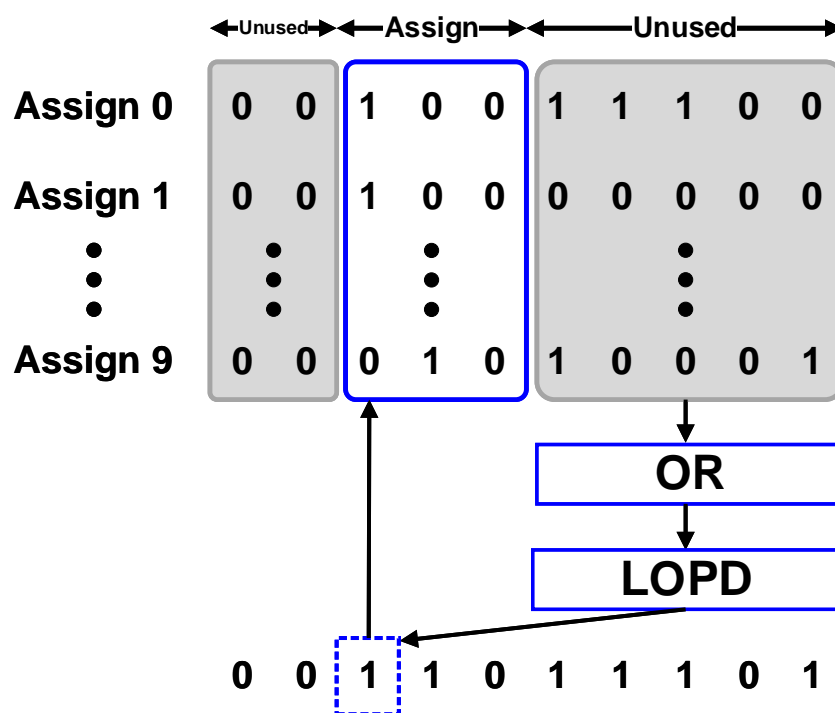
Figure 7. Hardware architecture.

Figure 8. The algorithmic mechanism of the proposed structure involves approximating the unused part and utilizing only the relevant portion for inference, if the assigned label structure is adopted.

The accuracy and energy efficiency for storing numbers in counter circuits are some significant advantages of this process. When implementing hardware, we do not need to calculate $\frac{s_j}{T}$ as per the Bayesian theory. Multiplying the number of test response spike counts has a mathematical meaning identical to Equation (2).

The proposed methods are adopted in a two-layer spiking network [9] designed to classify the MNIST dataset. Within the hardware structure, general leaky integrate and fire neurons were used and arranged in a parallel layer. Unlike hardware circuits, the neurons

are not parallel and updated one by one in this simulation. The network's input layer includes $28 \times 28 = 784$ neurons, and the output layer consists of 400 excitatory and inhibitory neurons, respectively. This network trains the dataset using the conventional STDP. The excitatory layer and inhibitory layers are connected in a one-to-one manner. The inhibitory layer performs the WTA mechanism through lateral inhibition. Each comparison has the same training phase and inference phase. We compare using different operations only in the neuron labeling phase. Theoretically, each method has a different meaning. In the inference phase, however, the same circuit can be reused and implemented. In the neuron labeling phase, different operations are used to compare the accuracy. Therefore, this study obtained the results using four different neural labeling methods based on the same training network. The accuracy measurement was achieved via applying different inferring methods in the same trained network. Table 1 shows the average accuracy of 20 tests using each inferring method. The accuracies of the probabilistic assignment method, the inferring method presented in [9], and the bit neuron labeling method with a valid range of 2 and 3 bits are compared. The probabilistic classification method with 16-digit floating point precision exhibits the highest average accuracy (85.13%). Conversely, 3 bit neuron labeling, which is divided into 8 integer-type levels with a valid range of 3 bits, presents a 0.63% lower accuracy than that of the probabilistic classification method. Although the bit neuron labeling method displays accuracy loss, it exhibits considerable efficiency for maintaining a 16-digit floating point number. Moreover, even if the valid range is lowered to 2 bits, i.e., 4 integer-type levels, there is no significant loss of accuracy. Thus, sufficient learning can be achieved by classifying neurons into four levels(most actively responding, moderately responding, lukewarm responding, and not responding). However, research was not conducted when the limit was set to 1 bit (2 levels).

Table 1. Hyperparameters used in the experiment.

Params	
Iteration	20,000
Hidden neuron	400
Membrane potential decay rate	0.1
Leakage	0.0025
Spike threshold	0.005
Ltd learning rate	0.001
Ltp learning rate	0.16
Initial membrane potential	0

Table 2 shows the result of synthesizing the proposed circuit. This table compares the area of the hardware chip, the delay between input and output, the energy consumption of the chip, and energy improvement accuracy. The synthesis is conducted using Samsung's 32 nm chip, and the synthesis tool used is Synopsys. As a result, a remarkable performance gap between the circuit area of determining the maximum value index and the area of the LOPD and shifter is observed. LOPD, the core circuit of the proposed method, is suitably optimized and can operate in a minimum area with less energy consumption [13].

Table 2. Accuracy comparison table of four methods with standard deviations and *p*-values.

Method	Accuracy	Standard Deviation	*p*-Value
Baseline	81.92%	0.15	0.01
Probabilistic inferring	85.13%	0.18	0.01
Bit(2) inferring	84.40%	0.33	0.01
Bit(3) inferring	84.50%	0.21	0.01

For each method, we compared the circuit for the neuron labeling phase. The baseline method needs a counter to count each neuron labeling spike, a circuit to compare

the largest neuron, and a 3-bit flip flop to assign the final neuron label. The method of bit neuron labeling show high efficiency in terms of energy and reduced delay since the real-time bit neuron labeling is different from the baseline method seeking the most significant spiking counts. Table 3 presents that the area of the proposed method is lower than the area of the baseline method. Although there is a certain level of parity in the power consumption values, the proposed method indicates enhanced energy efficiency due to the difference in the response speed. Hence, this method achieved a considerable performance improvement with respect to the circuit area and delay time because the LOPD and shift operate as a combination circuit. Consequently, the energy consumption per accuracy of the two proposed methods is approximately half that of the baseline method. The difference in accuracy between the 2- and 3-bit valid ranges is not significant. However, the 2-bit neuron labeling method is the most efficient method in terms of its energy efficiency. Hence, in the MNIST dataset, the 2-bit neuron labeling method is proved to be the most advantageous method.

Table 3. Performance summary of baseline and proposed methods.

Design	Area (μm^2)	Delay (*ns*)	Power (μW)	Energy (%)	Energy/Acc (10^{-3})
Baseline	3948.48	4.47	165.2	738.444	9.0138
Bit(2) inferring	3438.08	1.88	162.8	306.064	3.6262
Bit(3) inferring	3787.20	2.52	163.2	411.264	4.8669

5. Conclusions

This study presented a novel probability inferring method for spiking neural networks that is aligned with the principles of neuromorphic computing. Our results demonstrate that this method improves classification accuracy and energy efficiency while maintaining probability factors for various labels. The potential impact of this work is significant for advancing the field of neuromorphic computing and could lead to the development of low-computational-resource networks. Further research in this area could lead to the development of more efficient and powerful neuromorphic computing systems. This study presented a probability inferring model for SNN to achieve improved efficiency with respect to accuracy, circuit area, delay, power and energy. To demonstrate the performance of the spiking network while using the proposed method, a two-layer SNN designed to classify the MNIST handwritten digits using 1584 neurons was employed. The network was trained using the unsupervised STDP learning rule. The simulation results showed that the proposed method can improve the classification accuracy and energy. In addition, neurons maintained the probabilities for various labels. Therefore, since a new network trains each of these probability factors, the network can be established with low computational resources.

Author Contributions: Conceptualization and methodology, validation and formal analysis, resources and data curation, M.S.; writing—original draft preparation, investigation, J.K.; writing—review and editing, J.-M.K.; supervision, J.-M.K. All authors have read and agreed to the published version of the manuscript.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: MNIST dataset can be found at https://yann.lecun.com/exdb/mnist/.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhao, Z.Q.; Zheng, P.; Xu, S.T.; Wu, X. Object detection with deep learning: A review. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3212–3232. [CrossRef] [PubMed]
2. Zhang, Z.; Geiger, J.; Pohjalainen, J.; Mousa, A.E.D.; Jin, W.; Schuller, B. Deep learning for environmentally robust speech recognition: An overview of recent developments. *ACM Trans. Intell. Syst. Technol. (TIST)* **2018**, *9*, 1–28. [CrossRef]
3. Kang, Y.; Hauswald, J.; Gao, C.; Rovinski, A.; Mudge, T.; Mars, J.; Tang, L. Neurosurgeon: Collaborative intelligence between the cloud and mobile edge. *ACM SIGARCH Comput. Archit. News* **2017**, *45*, 615–629. [CrossRef]
4. Yu, S.; Wu, Y.; Wong, H.S.P. Investigating the switching dynamics and multilevel capability of bipolar metal oxide resistive switching memory. *Appl. Phys. Lett.* **2011,** *98*, 103514. [CrossRef]
5. Zhao, L.; Hong, Q.; Wang, X. Novel designs of spiking neuron circuit and STDP learning circuit based on memristor. *Neurocomputing* **2018**, *314*, 207–214. [CrossRef]
6. Yang, L.; Sun, Q.; Zhang, N.; Li, Y. Indirect multi-energy transactions of energy internet with deep reinforcement learning approach. *IEEE Trans. Power Syst.* **2022**, *37*, 4067–4077. [CrossRef]
7. John, R.A.; Ko, J.; Kulkarni, M.R.; Tiwari, N.; Chien, N.A.; Ing, N.G.; Leong, W.L.; Mathews, N. Flexible ionic-electronic hybrid oxide synaptic TFTs with programmable dynamic plasticity for brain-inspired neuromorphic computing. *Small* **2017**, *13*, 1701193. [CrossRef] [PubMed]
8. Zhong, G.; Zi, M.; Ren, C.; Xiao, Q.; Tang, M.; Wei, L.; An, F.; Xie, S.; Wang, J.; Zhong, X.; et al. Flexible electronic synapse enabled by ferroelectric field effect transistor for robust neuromorphic computing. *Appl. Phys. Lett.* **2020**, *117*, 92903. [CrossRef]
9. Diehl, P.U.; Cook, M. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Front. Comput. Neurosci.* **2015**, *9*, 99. [CrossRef] [PubMed]
10. Kheradpisheh, S.R.; Ganjtabesh, M.; Thorpe, S.J.; Masquelier, T. STDP-based spiking deep convolutional neural networks for object recognition. *Neural Netw.* **2018**, *99*, 56–67. [CrossRef] [PubMed]
11. Ferré, P.; Mamalet, F.; Thorpe, S.J. Unsupervised feature learning with winner-takes-all based stdp. *Front. Comput. Neurosci.* **2018**, *12*, 24. [CrossRef] [PubMed]
12. Yu, S.; Gao, B.; Fang, Z.; Yu, H.; Kang, J.; Wong, H.S.P. Stochastic learning in oxide binary synaptic device for neuromorphic computing. *Front. Neurosci.* **2013**, *7*, 186. [CrossRef] [PubMed]
13. Kunaraj, K.; Seshasayanan, R. Leading one detectors and leading one position detectors-an evolutionary design methodology. *Can. J. Electr. Comput. Eng.* **2013**, *36*, 103–110. [CrossRef]