Research on handwritten digits recognition system based on spiking neuron network

Zhao Liu

School of Mechanical Electronic & Information Engineering, China University of Mining & Technology, Beijing, China, 100083

chizkiyahuohayon@gmail.com

Abstract. In the 21st century, deep learning has revolutionized the fields of machine learning and computer science, attaining high accuracy in tasks such as image recognition. More layers and more parameters are stuffed into the network to achieve higher performance, making the network extremely large. A new, radically different approach was proposed to complete the tasks, such as image recognition, using a spiking neural network(SNN). The spiking neural network is event-driven rather than data-driven, which makes it more physiologically realistic and uses a lot less power. This study reviews the development of spiking neural networks and their differences from non-spiking neural networks, as well as the different encoding methods, neuronal models and update rules that have an impact on the performance of the network. It can be concluded that though SNNs can hardly achieve the same accuracy as artificial neural networks(ANN), the gap is narrowing. More strategies from ANN such as back-propagation and convolutional layers have been applied to SNNs, making it more accurate, stable and comprehensive.

Keywords: neuromorphic intelligence, spiking neuron network, artificial neuron network, LIF model, SDTP.

1. Introduction

Neuromorphic intelligence is a computer engineering method that mimics the structure and function of the human brain. It is a complete system containing both electrical circuits and algorithms implemented on the hardware. A spiking neuron network is the most common network model to realize neuromorphic computing, and units of the network are spiking neurons modeled after real biological neurons. In a spiking neural network, the data is transmitted as spike trains, in which the non-differentiability makes back-propagation not feasible, which is the most common training method in ANNs. Several approaches are proposed such as ANN-to-SNN conversion, and surrogate gradient descent. A scenario of different encoding methods, neuron models and learning rules need to be found to achieve the best performance. In this work, a convolutional spiking network(CNN) was implemented to perform the task of digit classification, and high accuracy was achieved.

2. Development

2.1. Development of ANNs

In 1950, Alan Turing published a far-reaching paper called *"Computing machinery and intelligence."* in which he envisioned a machine that could think like human. He believed the only way to realize artificial intelligence was to train a network of artificial neurons, which is exactly the essence of connectionism.

The first generation of an artificial neuron network is based on the perceptron, which was invented by McCulluh and Pitts in 1943 and implemented by Rosenblatt in 1958. Single-layer perceptrons are capable of lineally segregating two classes of pattern but can't do so with many classes of pattern. In 1969, Marvin Minsky showed that perceptron could not deal with the XOR problem, which led to the winter of AI.

The renaissance of neural networks began with the discovery that a multi-layer feedforward neural network may be taught to solve challenging problems like non-separable classification.

Artificial neural networks developed quickly in the twenty-first century as a result of the era of large data and usage of graphics processing units(GPU). The training of neural networks requires a large amount of data, and the GPU may considerably speed up the training process. The rationale is because a significant percentage of computing in ANN training and inference involves matrix computation, which may be performed on a graphics processing unit. As is shown in Figure1 that numerous arithmetic logic units are controlled by a single control unit, enabling parallel computation.

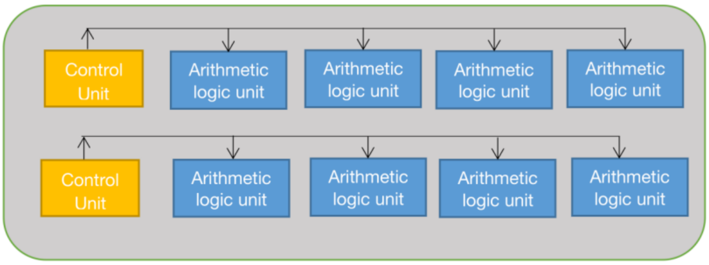

Figure 1. Architecture of GPUs.

2.2. Development of SNNs

The scale of deep neural network is insanely large nowadays, making training difficult and power consumption extremely high. The development of ANN is also hindered by Moore's law and the bottleneck of Von Neumann's architecture.

In 1965, Gordon Moore made the extrapolation that the number of transistors in an integrated circuit doubled about every two years. In the past, the development of software was largely contributed by the development of hardware. However, it is believed by many people that Moore's law is close to its end due to the physical limitation. For example, a transistor smaller than an atom can never be made.

In the Von Neumann architecture, which is the foundation of modern computers, data and instructions are stored and processed separately, in which transferring data largely increases time and energy consumption. Besides, ANNs bear little resemblance to the biological neural network.

Some experts believe that the spiking neuron network, which is more scientifically explicable, is the third generation of neural networks [1]. The computing engineering technique known as the spiking neural network imitates the structure and operation of the human brain. Spiking neurons, a duplicate of actual neurons, are the building blocks of spiking neural networks.

3. Biological and spiking neurons

3.1. Biological neurons

As physicist Feyman said, “What I cannot create, I do not understand”. If we want to devise a machine that can work like a human brain, we must know how a human brain works. So before we dive into the mathematical model of a spiking neuron, we should first have a simple idea of the biology of a biological neuron. Building a circuit and a network that work like a human brain is an important way to explore the treasure of human brains and realize general artificial intelligence. Neurons are the primary functional units of the brain, and our mental experiences, thoughts, and memories are the results of molecular movement across the neural membrane.

A neuron consists mainly of 3 parts: dendrites (input), soma (processing unit) and axon(output). The complicated functions of the brain are realized through the processing and communication of neurons. Neurons receive signals from other neurons through the dendrites, and send signals to the downstream neurons through the axon. A single neuron can receive information from more than 10e4 neighbouring cells. The conjunction of two neurons is the synapse, where the transmitter of the former neuron is released and absorbed by the latter neuron, cause action potential. An action potential is an all-or-nothing event in which the membrane potential rises and falls, which can be easily represented in computers by 0 and 1. It typically has an amplitude of 100mV and a duration of 1-2 ms. After each firing, the sodium channels are closed for a while, during which the neurons can not fire no matter how strong the stimulus is. This is an important property of neurons called the refractory period, which is realized in some spiking neuron models.

3.2. Models of spiking neuron

(1) Hoghkin and Huxley model

H-H model was proposed in 1952, which depicted the three kinds of ion channels: Na+, K+, Cl+ and a membrane storing charge as a capacitor. It can present the properties of biological neurons accurately, but its complexity make it can’t be employed to a large network.

(2) Integrate-and-fire model(LIF)

Integrate-and-fire(IF) and LIF models have simple computation and are easy to implement on hardware, making them most commonly used in SNNs. In IF neorons, the membrane potential of the neuron increases with the arrival of spikes. Once the membrane potential reaches the fixed threshold, the neuron emits a spike. At the same time, the membrane potential drops to the resting potential, waiting for new spikes to come. Figure 2 shows the relationship between spikes and membrane potential in IF models.

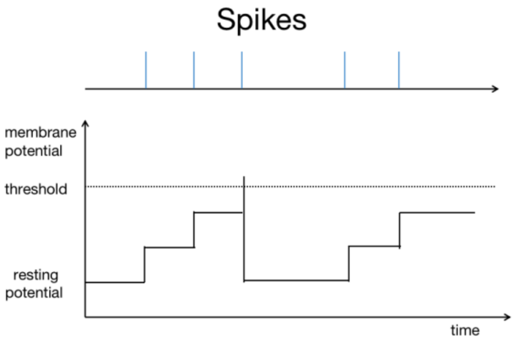

Figure 2. The Integrate-and-Fire model.

(3) Leaky Integrate-and-fire model

Based on LF models, leaky current is introduced to add temporal properties, making the membrane potential decreases expotentials over time The similarities between nerve membranes and RC circuits was observed by Louis Lapicque in 1907. In an resistor-capacitance circuit(RC circuit), the membrane which insulate the conductive saline solution is modelled with a capacitor, and the ion channels which are pathways for charge to flow are modeled with a resistor, as shown in Figure 3. The input current can be divided into Ic, which flows to the capacitor and Ir, which flows to the resistor.

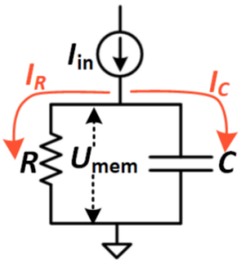

Figure 3. The RC circuit(Jason K. Eshraghian, 2021).

The linear ordinary differential equation describes the RC circuit is:

$$RC\frac{dU}{dt} = -U + RI_{in} \tag{1}$$

In another form:

$$U = I_{in}R + (U_0 - I_{in}R)e^{\frac{-t}{RC}} \tag{2}$$

Where R is the resistance, C is the capacitance, and U0 is the initialized membrane potential.

When the input current is low, the action potential is always below the threshold, so the spiking neuron will not fire. When the input current exceed the boundary, the firing rate increases as the input current increases, which shows the same property as biological neurons..

In the LIF model, the membrane potential increases instantaneously when the spike comes, and decreases exponentially to the resting potential as shown in Figure 4. In order to control the firing rate, the threshold will increment a little after each spike in the adaptive leaky model, which is a variation of the LIF model.

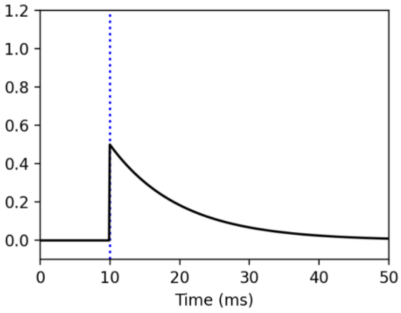

Figure 4. The Leaky Integrate-and-Fire model(Jason K. Eshraghian, 2021).

4. Topologies and encoding methods

4.1. Three kinds of topologies

Similar to structures in ANN, there are three types of topologies in SNN: feed-forward, convolutional, and recurrent, and they all include an input layer, hidden layers, and an output layer. A feed-forward neural network is composed of multi-layer perceptrons with randomly set weights, which will be updated during the process of learning. In CNNs, convolutional kernels are used for feature extraction, which are inspired by receptive fields in animals' retinas. Temporal information is included in the recurrent network, which means the input of a spiking neuron is not only the output of the former layer, but also also its own state a moment ago. This formula shows the temporal property of a recurrent neural network:

$$V_i(t) = V_i(t-1) + \sum_j W_{i,j} S_j(t-1) \tag{3}$$

$V_i(t)$: electric potential in time t

$W_{i,j}$: weight of jth neuron in former layer to ith neuron in this layer

4.2. Encoding method

The vixels, for instance, can be converted into spiking trains using diferent encoding methods. The simplest method is rate coding, in which the value of each pixel is represented by the firing rate. The advantages of being noise-resistant outweigh the disadvantage of high power consumption, which rises as the number of spikes does.

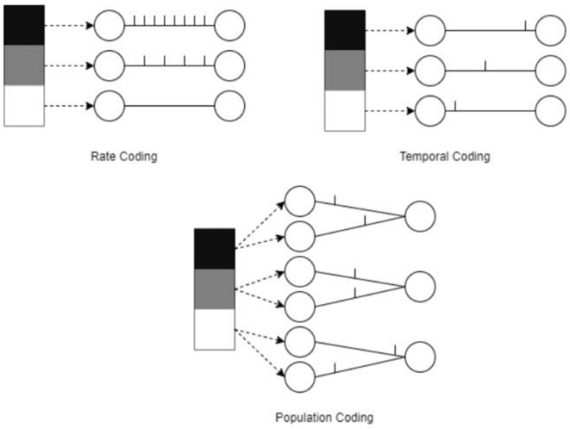

Figure 5. Three encoding method(Osaze Shears, 2020).

In temporal encoding, the neuron only fires once in a time step, and the neuron with higher intensity fires earlier. It consumes much less energy but it is sensitive to noise.

While population encoding appears to be a balancing of the previous two encoding techniques, more neurons are required to represent the input data in this method. Delta modulation is another approach which uses spikes to represent change of value, showing great temporal properties.

5. Learning algorithm

5.1. Spike Time Dependent Plasticity(STDP) algorithm

The hardware and the algorithm are two distinct concepts in computer science. While the algorithm is the hardware in biology. The topologies of spiking neural networks play a significant role in the algorithm. Generally speaking, supervised learning and unsupervised learning are two categories of learning algorithms.

SDTP is the typical of local learning, which was first found in neuroscience. According to Hebbian rule: the neurons wire together, fire together. If the post-synapse neuron fires immediately after the pre-synapse neuron, we can assume that the firing of the pre-synapse neuron may cause the post-synapse neuron to fire, and we increase the weight of the synapse between the two neurons. On the contrary, if the pre-synapse neuron fires immediately after the firing of the post-synapse neuron, there can't be a correlation, so we decrement the weight of the synapse.

Update the parameter between the presynaptic neuron and postsynaptic neuron:

$$\Delta \mathrm{w} = \begin{cases} \mathrm{A}_{+}\exp\left(\frac{\Delta \mathrm{t}}{\tau_{+}}\right), \Delta \mathrm{t} < 0, \\ \mathrm{A}_{-}\exp\left(\frac{\Delta \mathrm{t}}{\tau_{-}}\right), \Delta \mathrm{t} > 0. \end{cases} \tag{4}$$

The time difference $\Delta t = t_{pre} - t_{post}$ between firing of presynaptic neur0n and postsynaptic neuron determines the change of w, which is the weight of the synapse.

5.2. Surrogate gradient descent

In conventional deep neural networks, the gradient descent algorithm is the most fundamental method to train the network. After initializing the parameters, including weights and biases, the loss is calculated by measuring the difference between predicted outputs and actual outputs(labels) using the established loss function L. The back-propagation algorithm calculated the partial derivatives according to the chain rule, tweaking all parameters as shown in the formula below.

$$\theta = \theta - \alpha \frac{\mathrm{dL}}{\mathrm{d}\theta} \tag{5}$$

After performing numerous iterations, the minimum of the loss function and associated weights are determined. The estimated value and target value in ANNs are normally vectors, whereas in SNNs they are spike trains, which is the difference between gradient descent in ANNs and SNNs. For instance, in image recognition, the pixels of the image are first fed into a spiking neural network as spike trains encoded using a particular encoding method. Therefore, the key distinction is that, in contrast to ANNs, data is transferred as spike trains between spiking neurons. It became difficult to achieve back-propagation since the discrete spike is non-differentiable Back-propagation was made possible by softening the output of the non-differentiable system, which is called the surrogate gradient approach.

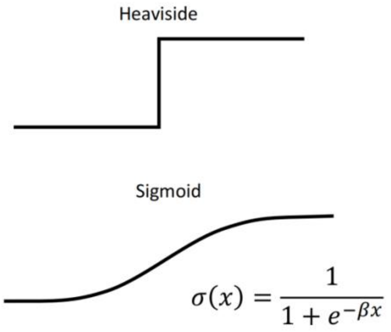

Figure 6. Surrogate gradient descent approach.

As shown in Figure 5, the activation function of a spiking neuron is non-differentiable. The derivative of spiking with respect to membrane potential is zero(no spiking) or positive infinite(spiking). In the forward pass, the spiking of the neuron is controlled by the shifted heavyside function as before. However, in the backward pass, the derivative is calculated with smoothing functions such as the sigmoid function. This approach works well on some tasks, although the speed is limited.

6. Implementatin

Nengo, a Python library for creating spiking neural networks, was used to construct a handwriting digits recognition system. This work made use of the packages nengo, nengo_dl, tensorflow, and numpy. The MNIST datasets, which contain both images and labels, are initially divided into training set and test set. The test set includes images that have been preprosessed and flattened. Below are photos that are gray.

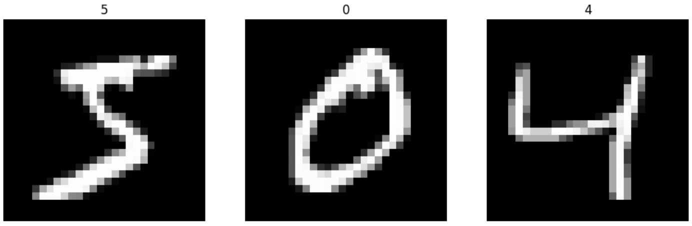

Figure 7. Grayscale images of handwritten digits.

The next step is to construct a convolutional spiking neural network with LIF-modeled neurons for each node. In the input layer, each of the 28×28 neurons in the input layer represents a pixel in the image. The information is subsequently sent in a to three convolutional layers. NengoDL.layer is used to create the layers. One notable difference is that time must be taken into account while analyzing the data, necessitating various timesteps to gather spike data throughout time. Ten units make up the output layer, one unit for each category. The accuracy of this technique ranges from 98% to 99%.

7. Conclusion

This essay is a review of the spiking neural network, and its encoding methods, neuron model and learning rules. And a convolutional spiking neural network used for image classification was implemented in Python.

There has been much progress in the training of spiking networks, via both supervised and unsupervised learning methods. For example, the conversion of back-propagation trained ANNs to SNNs has been a success. Although most tests on SNNs are benchmarked with simple datasets such as MNIST and CIFAR datasets, there has been success in training SNNs on ImageNet, which is a significant advance. It has been pointed out that the static datasets don't correspond with the dynamic properties of SNNs, and the resulting gap between the accuracy of SNNs and that of ANNs is narrowing.

Using local learning methods such as SDTP to train large-scale networks remains a challenge. And few attempts have been made to connect recurrent networks to their spiking counterparts. How to apply different learning methods and network topologies will continue to be a research direction in the future.

References

[1] Maass W. Networks of spiking neurons: The third generation of neural network models. Neural Networks. 1997;10(9):1659–1671.

[2] Diehl, P. U., and Cook, M. (2015). Unsupervised learning of digit recognition using spike-timing-dependent plasticity. Front. Comput. Neurosci. 9:99. doi:10.3389/fncom.2015.00099

[3] G. Li, L. Deng, Y. Chua, P. Li, E. O. Neftci, and H. Li, "Spiking neural network learning, benchmarking, programming and executing," Frontiers in Neuroscience, vol. 14, 2020.

[4] Jason K. Eshraghian, Max Ward, Emre Neftci, Xinxin Wang, Gregor Lenz, Girish Dwivedi, Mohammed Bennamoun, Doo Seok Jeong, and Wei D. Lu. "Training Spiking Neural Networks Using Lessons From Deep Learning". arXiv preprint arXiv:2109.12894, September 2021.

[5] E. M. Izhikevich, "Which model to use for cortical spiking neurons?" IEEE transactions on neural networks, vol. 15, no. 5, pp. 1063–1070,2004.

[6] A. M. Turing, "Computing machinery and intelligence," Mind, vol. 59, no. 236, pp. 433–460, 1950.

[7] E. M. Izhikevich, "Simple model of spiking neurons," IEEE Transactions on neural networks, vol. 14, no. 6, pp. 1569–1572, 2003.

[8] Anthony N Burkitt. 2006. A review of the integrate-and-fire neuron model: I. Homogeneous synaptic input. Biological cybernetics 95, 1 (2006), 1–19.

[9] Diehl, P. U., Neil, D., Binas, J., Cook, M., Liu, S.-C., and Pfeiffer, M. (2015). “Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing,” in International Joint Conference on Neural Networks (IJCNN) (Anchorag, AK), 1–8. doi: 10.1109/ijcnn.2015.7280696

[10] Indiveri, G., Corradi, F., and Qiao, N. (2015). “Neuromorphic architectures for spiking deep neural networks,” in 2015 IEEE International Electron Devices Meeting (IEDM) (Washington, DC: IEEE), 1–4. doi: 10.1109/iedm.2015.7409623

[11] Sengupta, A., Ye, Y., Wang, R., Liu, C., & Roy, K. (2018). Going deeper in spiking networks: VGG and residual architectures, arXiv [Preprint]. arXiv:1802.02627v3.

[12] Bi G-Q, Poo M-M. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. J. Neurosci. 1998;18(24):10464.