Spiking Neural Networks Vs Convolutional Neural Networks for Supervised Learning

Sahil Lamba
Information Technology,
Bharati Vidyapeeth's College Of Engineering, New Delhi
Email: sahillamba45@yahoo.com

Rishab Lamba
Information Technology,
Bharati Vidyapeeth's College Of Engineering, New Delhi
Email: rishablamba407@gmail.com

***Abstract*—Deep learning has revolutionised the field of machine learning in near years, particularly for computer vision. Two methods are used to view supervised learning, SNN networks with the handwritten Digit Recognition Problem (NOD) and Normalized Normalized Approximate Descent (NORMAD). Experiments show that the identification accuracy of the prototype SNN does not deteriorate by more than 1% relative to the floating-point baseline, even with synaptic weights of 3-bit. In addition, the proposed SNN, which is trained on the basis of accurate spike timing data, outperforms the equivalent non-spiking artificial neural network (ANN) trained with back propagation, especially at low bit precision, and is in line with the convolutionary neural network that is normally used to train these system. Recent work shows the potential to use Spike-Based Data Encoding and learning for applications of the real world for positive neuromorphism.**

1. Introduction

The surpassing computating effectiveness of bio-logical systems has motivated the chase to reverse the mind to create smart computational platforms which can learn to perform a broad range of information analytics and. Inspired by the actual network architecture of the build of the brain, Artificial Neural Networks (ANNs) have appeared as the latest technology for multiple machine learning apps. Neurons in ANNs perform nonlinear memory less synaptic signals as input to their conversion to generate real valued signals as output. [1]

CNNs consist of neurons with learning weights and biases, including neural networks. Each neuron receives different inputs, receives a weighted amount, performs an activation function and responds with an output. The entire network has a missing purpose and on CNNs, we can still use all tips and tricks that we have built for neural networks. [2]

(SNNs) that more tightly imitate natural neural networks are Artificial neural networks. In addition to the neuronal and synaptic state, SNNs incorporate the concept of time in themselves.The idea here is that in every propagation cycle the neurons in the SNN do not fire (as is the case for typical multilayer perceptron network), but only fire when the intrinsical quality of the neuron associated with its electric membrane load is a membrane potential that reaches a certain value. [3] If a neuron fires, it produces a signal to other neurons that improves or reduces their capacity according to this signal.

Optical character recognition (OCR) sometimes fails to acknowledge the handwriten text because the methodology of writing varies from individual to individual. An OCR's primary job is to effectively distinguish and acknowledge printed text. OCR covers the following stages: preprocessing, segmentation, extraction of features and recognition classification. In pattern recognition, ML sight and AI , OCR is a study subject of great significance. OCR is also named Off-line Character Recognition System. [3]

1.1. Dataset:

MNIST

TABLE 1. DATASET PARTITION

Training Images	Testing Images
60,000	10,000

2. Spiking Neural Networks:

Spiking NNs are among the generation three of nueral networks using neuronal models descending from neuronal signaling biological processes. While the spiking process mechanism in biological neurons is dependent on the cell membrane's complicated ion-channel interactions, a computationally easier (leaky integration and fire) LIF model from computing prospective is typically used to simulte SNNs.This reflects a neuron's potential as the voltage across a capacitor linked to a leaky conductance route in parallel and is loaded by "incoming input currents". According to the differential equation, evolution of membrane potention V(t) is as follows:

$$C\,dV(t)/dt = -gL(V(t) - EL) + Isyn(t) \tag{1}$$

When V(t) is VT local, a spike is issued and transferred into the downstream synapses ; after the spike, its EL

ISBN: 978-1-7281-4826-7/19/$31.00 ©2019 IEEE

resting value is reset. In our simulations, we use EL = 70miliV and VT = 20miliV. C = 300 picoF and gL = 30 nanoS, respectively, modelling the capacity and leaking of the membrane. Immediately after a spike is given, biological neurons enter a refractory period "during which another spike can not be issued".This is enforced by keeping the membrane potential at V(t)= EL for a brief tref = 3 ms after a spike has been issued.

The spikes reaching the synapse with force (weight) w generate the following post synaptic current (syn(t)) in the neuron downstream:

$$c(t) = \Sigma i\, \delta(t - ti) * (e - t/\tau 1 - e - t/\tau 2) \quad (2)$$

$$Isyn(t) = w \times c(t) \quad (3)$$

Where ti refers to the moment of problem of the incoming spike and where * is the operator of the convolution. The $\pi 1$=5 miliS and $\pi 1$=1.25 miliS variables model the format of the existing synaptical unit C(t) and show its time constants that fall and grow.Note that the time of issue of LIF neuron spikes depends in a strong non-linear manner on the incoming spike times and the synaptic force due to the weighted summation, integration and reset.

3. Convolutional Networks:

In order to achieve the issue of classification and recognition, conventional neural networks use three fundamental variables. The job of taking input was done in the local receiving fields that are fully connected as a vertical pixel intensity column. We see it in a convolutionary net as 28 by 28 square neuron matrix that correspunds to the image input.In this case we will not connect all input layer pixels in the hidden layer to every other neuron, but rather interact in small, located image fields. Let us say, for example, a 5 to 5 region that equals 25 input pixels. So we could have neuron-like connections.

The region in the input picture is called the hidden layer neuron local receptive field. Every relationship is gaining a weight. [4]

In the subsequent segment of common weights and partialities the main objective is to have a preference and weight connected to your local receptive area. One thing is that for every 24 hidden neurons we will use the same weights and tendencies. All the neurons in the first hidden layer will simply detect the very same function at different locations of the output image as the input is shifted by a local receptive field. This forecast can help in other areas of the picture to make it reasonable to assume weights and harms that hidden layers can predict the vertical edge in a particular receptive local environment. Neural networks Convolution is well-known in practice for the invariance of images. This mapping is sometimes referred to as the output layer function map for the hidden layer. For knowing the feature, we define weights as shared weights and bias as the shared bias, both often referred to as kernel or filter. [5]

The next processing phase used is called Pooling layers, which are also component of the concealed layer and present after the Convolutional layer. Pooling layers simplify data from convolution layer production and create a very thin and condensed feature map that can predict thinner and finer details for extraction of features. These characteristics can be anticipated again at any picture location. "Each unit of the pooling layer can predict a 2 by 2 neurons from the convolutionary layer in practical terms. The pooling method is coined as Max-pooling". [6]

4. Architecture

4.1. SNN

Us authors intended a straightforward MNIST database 3-layered SNN for handwritten digit clasification. Since MNIST pictures are 28 serv28 pixels, there are 784 neurons in the input layer of our network and 10 neurons in the output layer, each corresponding to a specific digit.The input layer neurons bind to 8112 cache neuron layer neurons by 12 kernels of 3 and 3 sizes which have been previously set. synapses connecting the cache layer to the outputlayer are shown using the NormAD algorithm. [7]

4.1.1. Encoding input. For the encoding of real-world data by computerized time encoding machines, for translating small input signal into a spike domain, the use of complex transformations like speed encoding, spike time coding and population coding are biological sensory nerve settings. Several recent projects using Gaussian reception zones and Poisson encoding are also available to directly convert valued data into time . For static images, each gray pixel value is converted into streams which can be used for inputs in the [0, 255] spiking neurons array. Therefore, each pixel value k is translated into a permanent LIF neuron input stream

$$i(k) = I0 + (k \times Ip) \quad (4)$$

In which the scaling factor is Ip= 101,2 pA and in which the LIF neuron spike in eq(1) is the highest steady amplitude present of I0= 2700 pA. The input layer of the LIF neuron therefore produces pikes which are spaced in time in uniformity with a frequency which is sub-linearly proportional with the "magnitude of the input current".

4.1.2. Feature Extraction. The convoluting laying of this network utilizes the fixed value of weights for the various characteristic mappings and further detects the main characteristics of the picture. The filters are constant curves, as illustrated in Fig.2(L) and include excitatory as well as inhibitory links. Our kernels are only 3x 3 pixels and have been motivated by biological research suggesting that small, visually receptive areas are the first layers of the visual cortex.

With all the spike trains(28/28) arriving from the input layer neurons over a simulation time period T, the filter kernels are spatial compatible with 1 stroke resulting in 26

/ 26 character maps. The kernels have a network greater inhibition than excitement, because it helped to better delete spikes in the respective functional map from unwanted corners of the input digit picture. Previously, the first layer of a highly complex neural network used for fixed weights, utilizing Gabor filters, increases the reliability of the MNIST dataset compared with the initial LeNet-5. Comparably simpler rim detection filters are used in the hidden layer of our network. [8]

These synaptic weight-kernel spikes from the input layer neurons create currents for cache layer neurons. The magnitude of the current entering the concealed layer neurons is reduced to a limit of 10 Hz on average. Picture. 2(right) displays in the 2D function maps showing the amount of spikes generated by a cache neuron when an exemplary picture of Digit 9 is presented from MNIST data set to the network in T=100ms. Effectively encoding edges and input image features in the spike domain is possible with the different kernels.

4.1.3. Learning Layer. With the NormAD algorithm, the synaptic weights connected to the clock layer of the output classifier are trained. At the start of the training, weights are initialized to zero. At the end of the image display that takes place at the interval T, in this fully linked network layer, the weights of all 8112/10 synapses are changed. The synaptic weights aligned with the clock layer of the output classificator were learned using the NormAD algorithm. At the start of the training, weights are initialized to zero.Weights were set to zero at the end of practice. Update of all 8112/10 synapses of this fully connected network layer at the end of each photo which takes place over a T-interval. [9]

$$w(n+1) = w(n) + \Delta w. \quad (5)$$

The weights updated are only calculated when the spike times of the required weight updates (sd(t)) are different from those observed (So(t)) Spike Trains, e(t)= Sd(t) Sd(t). So(t). The cost function can be established by identifying the mistake in the value between required neuron membrane potentials (Vdes (t)) and (V(w, t)) as

$$J(w) = 1\,2 \,' \; T\,0\, |e(t)|(V\,des(t) - V(w, t))2dt \quad (6)$$

You can use the gradient descent on the Instant Cost J(w, t) that you get by restricting the integral limit in Eq for an instant weight change. (6) at a time interval infinitesimally small, t as,

$$\Delta w(t) = \eta\,(t)\nabla w\,J(w, t) \quad (7)$$

Along with

$$\nabla w\,J(w, t) = |e(t)|(V\,des(t) - V(w, t))\nabla wV(w, t) \quad (8)$$

The η(t) in Eq(7) is a time-dependent constant of proportionality. By normalizing and approximating the weight reliance of the membrane potential, a closed form connection can be obtained for weight updating as

$$\Delta w = r\,'\,{}_0^T \; e(t)\frac{\mathrm{d}(t)}{||\,\mathrm{d}(t)||}\,dt \quad (9)$$

The ideal Sd(t), which is a slightly higher spike than the 3ms LIF refractory duration, is a standardized spike train of frequency 285 Hz equal to one spike per 3.5 meters. For all other neurons, there are no spikes in Sd(t).

4.1.4. Output layer lateral inhibition. Including feed-forwards from the convolution neuron node, each output layer neuron is also supplied with lateral inhibitory inputs from the other nine output neurons that are analogous to Lee et al. (2016) applying the dynamics of winner-take-all (WTA). The outgoing WTA synapses of the neuron spikes induce detrimental current in other neurons and block their peaks as shown in the figure. 3.

4.2. CNN(LeNet)

In our Convolutional Networks software, which is based primarily on local receptive fields and kernels and sub-sampling or pooling, we will use LeNet technology to ensure multiple distortions are invariant. A standard digit recognition and identification neural convolution network as shown in Figure 5. The input layer receives images that can be downloaded from the accessible libraries from the MNIST dataset. They were organized and uniform. For regional receptive fields like endpoints, directed sides, lines, neurons can acquire distinct vision characteristics. Those features are more reliably determined from subsequent surfaces. All units in a function map are extracted to perform the same procedure at different locations of the image. A detailed network of convolution consists of various feature mappings with distinct weights and biases so that different features can be extracted at once and added to each part of the image. Figure 5 displays the first layer of our LeNet architecture as a practical example. Units in the first secret layer of LeNet are grouped in the 6 feature mappings of our network [10] . Can unit takes a local receptive file or kernel5-by-5 from the output surface. Every neuron in the hidden layer therefore has 25 separate trainable parameter equations which can be learned while implementing the learning algorithm and can be improved using cost function and activation function. It also has a trainable version bias that is special to any form of learning. The corresponding neighboring units in the output layer form the adjacent receptive fields. TMany layer structure maps may have different weights and priorities and may be useful to forecast many features. Six distinct types of features are obtained in interface maps in our CNN's LeNet architecture at each output layer by six units at the same location. "A step-by-step execution of the function map would scan the specified input picture with a single unit having a local receptive field and store the unit information in the function maps at the appropriate places. This procedure is identical to a convolution, followed by an extra bias and activation function so that the name convolutional neural network. The kernel we use in the network is set of weights used by units in mathematical strategy to convolution in feature mapping. An amazing feature is that the amount by which the input image is shifted, the same amount is shifted to the feature map output. This property guarantees

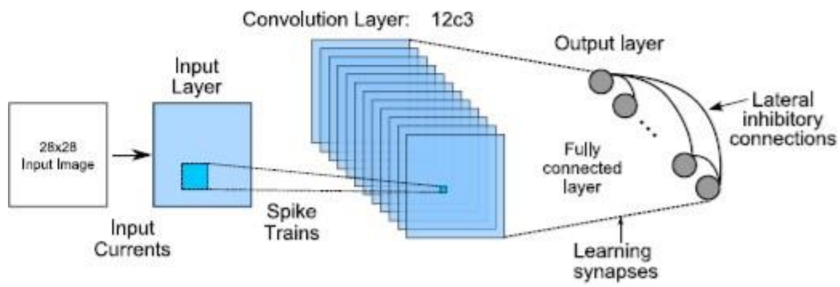

Figure 1. The suggested spiking architecture of the neural network for handwritten digit classification.

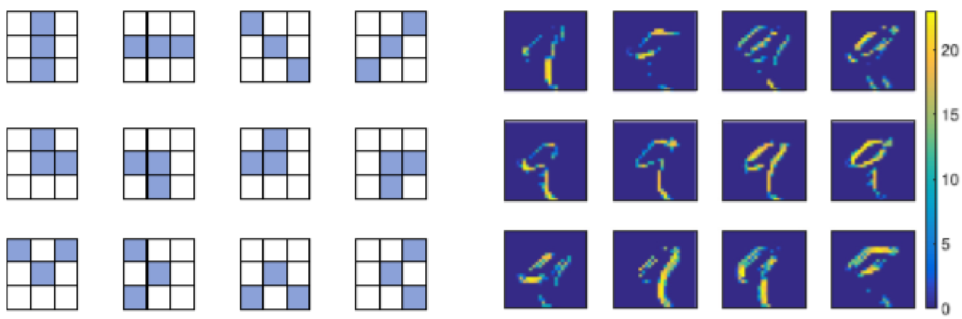

Figure 2. Convolution filters are 3x 3 pixels in our SNN. The blue pixels are the values of excitement, white pixels are the inhibition.

that input changes and disturbances are invariant to our network." [11]

Once the function is extracted, it does not need its precise position in the image, only the approximate positioning is necessary relative to other features extracted. In that analysis, for example, researchers have come to understand that the output image provides an endpoint for a horizontal line segment in the upper left field. In the reduced part of the image, they can assume that the endpoint of a vertical line segment is a seventh. But these exact positions are important for each of their characteristics, since the positions the vary in individual input digit situations. The only way to solve this problem is by increasing the spatial resolution of the characteristic map. It comes with sub-sample layers that conduct a local average and sub-sample, thus increasing the resolution of the map. The second secret layer of LeNet is a network of sub-samples. This layer replaces one feature map with 6 different feature maps in the previous layer. The receptive field in the past layer function map for each unit compromises 2 by 2 kernels. Every system performs a standard procedure like convolution, which combines its four outputs and is weighted by the trainable weight matrix, adds a trainable bias and goes through the activation function like the sigmoid function. We will use the feature ReLu activation used in rectifiers in our implementation. As the maps in our convolutionary layers had, there are half rows and columns in a sub-sampling layer function graph. The coefficient and discrimination control the effect of the activation function. When the sampling layer is low and only blurs the picture the sub-sampling layer, this system runs in a near linear mode. One after another layers of convolution and sampling result in a two way pyramid at each layer when done on the image. With this consecutive decrease in the spatial resolution of feature maps, a greater degree of invariance can be achieved and also allows us to predict and detect finer information and boost the representation of richer information. It makes it easy for the network to identify features at a very rapid rate, making it easier to use CNN's preference. [12]

Since all weights are learned using learning algorithms, in this situation authors have used back propagation, it is possible to view convolutionary networks as a self-extractor of their characteristics. The technique of weight sharing has limited storage memory and time for calculating these weights.

5. Result

TABLE 2. RESULT

Network	Accuracy	Epochs
SNN	98.14	30
CNN	99.24	30

6. Conclusion

Network quality is dependent on many factors, such as limited memory constraints, high latency and improved accuracy, although this article focuses solely on improving

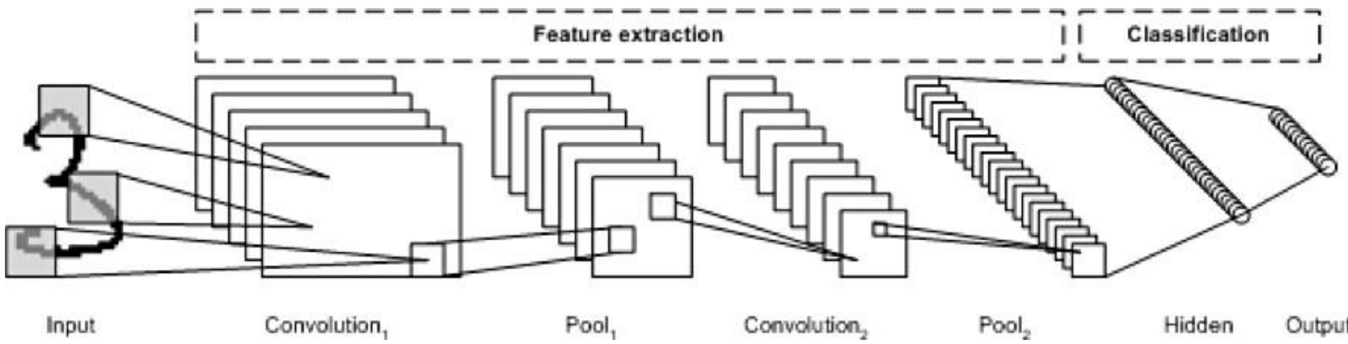

Figure 3. LeNet Architecture

classification accuracy. Previously Artificial Neurons was more effective, but right now the CV division mostly depends on profound learning technologies such as Convolutionary Neural Networks and Spiking Neural Networks. We provided an extremely compact and effective 3-layer spiking neural network to identify handwritten numbers, using the NormAD learning algorithm to achieve a precision of 99.24 percent on the NIST dataset. Alldata on the network is encoded and stored in the spike domain at sporadic biological spike levels. In artificial intelligence, the main motivation for computer vision is to create a network that is good for any output assessment and provides outcomes for all kinds of data sets that can be learned or educated and understood.

References

[1] B. Illing, W. Gerstner, and J. Brea, "Biologically plausible deep learning - But how far can we go with shallow networks?" *Neural Networks*, vol. 118, 2019.

[2] J. Pradeep, E. Srinivasan, and S. Himavathi, "Diagonal Based Feature Extraction For Handwritten Alphabets Recognition System Using Neural Network," *Electronics Computer Technology (ICECT), 2011 3rd International Conference on*, vol. 4, 2011.

[3] M. S. Alkoffash, M. J. Bawaneh, Hasanmuaidi, and M. Shihadehalqrainy, "A Survey of Digital Image Processing Techniques in Character Recognition"," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 14, no. 3, pp. 65–65, 2014.

[4] A. Sharma, R. Dipti, and Chaudhary, "Character Recognition Using Neural Network"," *International Journal of Engineering Trends and Technology*, pp. 662–667, 2013, 4.4.

[5] M. Anshul, C. Shrivastava, and Mahanta, "Offline handwritten character recognition using neural network." Computer Applications and Electronics (ICCAIE)," *IEEE International Conference on*, 2011.

[6] M. Mozafari, M. Ganjtabesh, A. Nowzari-Dalini, S. J. Thorpe, and T. Masquelier, "Bio-inspired digit recognition using reward-modulated spike-timing-dependent plasticity in deep convolutional networks," *Pattern Recognition*, vol. 94, 2019.

[7] F. Kimura, N. Kayahara, Y. Miyake, and M. Shridhar, "Machine And Human Recognition Of Segmented Characters From Handwritten Words," *4th International Conference On Document Analysis And Recognition (ICDAR '97)*, vol. 2, pp. 866–869, 1997.

[8] Y. Lecun, B. Baser, J. S. Denker, D. Henderson, R. E. Howard, R. Hubbard, and L. D. ., "Handwritten digit recognition with a back-propogation network in D.Tourezky," *Advances in Neural Information Processing Systems*, vol. 2, 1990.

[9] R. Shruti, B. Kulkarni, and Rajendran, "Spiking neural networks for handwritten digit recognition-Supervised learning and network optimization," *Neural Networks*, 2018.

[10] H. Yamada and Y. Nakano, "Cursive Handwritten Word Recognition Using Multiple Segmentation Determined by Contour Analysis," *IEICE Transactions on Information and Systems*, vol. 79, pp. 464–470, 1996.

[11] "Corima Cartes and Vladimir Vapnik The Soft Margin Classifier," *Machine Learning to appear*, 1995.

[12] S. Singh and M. Hewitt, "Cursive Digit And Character Recognition on Cedar Database," *Proceedings. 15th international conference on*, vol. 2, 2000, Pattern Recognition.