Evaluating SPAN Incremental Learning for Handwritten Digit Recognition

Ammar Mohemmed[1], Guoyu Lu[3], and Nikola Kasabov[1,2]

[1] Knowledge Engineering Discovery Research Institute,
Auckland University of Technology, New Zealand
{amohemme,nkasabov}@aut.ac.nz
http://www.kedri.info
[2] Institute for Neuroinformatics, ETH and University of Zurich
[3] Department of Information Engineering and Computer Science
University of Trento, Italy

Abstract. In a previous work [12, 11], the authors proposed SPAN: a learning algorithm based on temporal coding for Spiking Neural Network (SNN). The algorithm trains a neuron to associate target spike patterns to input spatio-temporal spike patterns. In this paper we present the details of experiment to evaluate the feasibility of SPAN learning on a real-world dataset: classifying images of handwritten digits. As spike encoding is an important issue in using SNN for practical applications, we discuss few methods for image conversion to spike patterns. The experiment yields encouraging results to consider the SPAN learning for practical temporal pattern recognition applications.

Keywords: Spiking Neural Networks, Supervised Learning, Nuerocomputing, Spatiotemporal pattern recognition

1 Introduction

Driven by the emerging need for systems that can behave autonomously and adaptively through learning, research is turning to biological intelligence for better solutions. The knowledge about how the brain is functioning that becomes available due to the discoveries of neuroscience is inspiring researchers to mimic the brain, at different levels, to create more efficient methods and systems. Spiking Neural Networks (SNN) [5, 9], considered the third generation of artificial neural networks, is an important tool to model many functional aspects of the brain. Furthermore, SNN models have been investigated for a number of computer applications including computer vision [18, 3], speech recognition [17], autonomous robots [4, 15] and others.

Most of real-world data is represented as static or dynamic continuous values. In the latter case, data changes in time and space where useful knowledge can be extracted only after a certain time period of observing the data. An example is extracting information from video data, such as human action recognition for human-robotic interaction or security/health surveillance.

SNN internally use spikes to communicate, where information is encoded in the time of the spikes. That makes SNN to be suitable for spatio-temporal data processing. However, for SNN to be applicable for real-world problems, input data needs to be transformed into spikes before it can be processed in SNN.This conversion should be done properly such that the inter-class/intra-class relationships between data categories are preserved, otherwise the recognition task using SNN will be hard to yield accurate results.

In fact how to encode information into spikes is a challanging problem that extends to a deep research field in neuroscience. According to previous studies such as in [6] and recent one [14], temporal coding whereby information is encoded into precise time of the spikes, plays a significant role in the neural code of the brain especially in the visual system.

SPAN [12, 11], is a learning algorithm for spiking neural networks which is based on encoding input information as precise time of spikes (Temporal coding). This is opposite to rate coding where information is coded in the mean firing rate of the neurons. The algorithm was evaluated mainly on two tasks: precise time spike sequence generation, and spike pattern classification [12]. In spike sequence generation task, a spiking neuron is trained to generate any random spike train in response to a recognised pattern of input spike sequences (spike trains). This property is also used for temporal spike pattern classification by training the neuron to associate different spike trains to different input classes. Recently, we have extended the application of the algorithm to train multiple neurons to classify multiple classes of spike patterns generated artificially [10].

In this paper, we investigate SPAN learning on a practical dataset where the task is to classify images of handwritten digits from the MNIST dataset [7]. The first stage in the learning process is to convert the images into spike patterns. The conversion is done using the Virtual Retina [19] -a simulator that transforms image/video into pattern of spike trains. The generated patterns are used to train a single layer of SPANs for classification. The MNIST is a non-linear dataset which guarantees that the different spike pattern classes have more complicated inter/intra- class characteristics, i.e., patterns that belong to the same class are varied and there is overlap between different classes, giving a clear indication of the feasibility of SPAN learning for a practical applications.

2 Image to Spike Coding

The main function of the biological retina is converting input image stimulus to patterns of spikes. How different features of the stimulus relate to the structure of the generated spike pattern is not clear. A number of research works have proposed simplified software and hardware tools to model the retina and other parts of the visual system. Rank Order Coding (ROC) is one of the earliest coding scheme for image coding is based on the biological principle of fast processing of image stimulus in the brain [16]. Information is encoded in the order of spike firing across a population of neurons in which each neuron fires at most a single spike. Based on this coding, a face identification system is proposed [2] and also

an evolving SNN (eSNN) architecture for integrated audio-visual information processing and pattern recognition [20].

A piece of hardware referred to Silicon Retina(SR) was made to convert input streams of image frames into spike patterns in a format referred to Address Encoding Representation (AER) [8]. The pixels of the SR respond to events that represent relative changes in intensity by computing the difference in pixel intensity between two successive frames (temporal contrast) and generating spikes if this difference exceeds a threshold value. Although the concept is simple, the hardware implementation provides fast computation power necessary for certain vision application.

The Virtual Retina (VR) is a software simulator that models more complicated aspects of the biological retina. It transforms a video input into spike patterns [19]. The VR consists of three stages of processing layers that correspond to different layers of the retina, namely the Outer Plexiform Layer (OPL), Contrast Gain Control (CGC) and the Ganglion layer (GL). Particularly, the GL layer is responsible of converting continuous current into spike trains. The conversion is performed using a Noisy Leaky Integrate and Fire (LIF) neuron, where the time delay of the generated spikes is proportional to the input current. The noise is represented as a random value added to the membrane potential of the LIF neuron in order to reproduce the variability found in trail-to-trail spike recording of real ganglion cells. Because the conversion is temporal, the VR is suitable to use for SPAN learning which is also based on temporal coding. We use a basic configuration of the VR, that consists of OPL and a single GL, as an encoder to convert digit images into spike patterns.

3 SPAN Learning Method and Network Topology

In this section, we describe briefly SPAN learning rule and the SNN network architecture. More details can be found in previous publications [10–12]. SPAN rule is a supervised learning method to associate input spike pattern to a target spike train by adjusting the weights of the input synapses according to the following formula:

$$\Delta w_i = \lambda \left(\frac{e}{2}\right)^2 \left[\sum_g \sum_f (|t_i^f - t_d^g| + \tau_s) e^{-\frac{|t_i^f - t_d^g|}{\tau_s}} - \sum_h \sum_f (|t_i^f - t_a^h| + \tau_s) e^{-\frac{|t_i^f - t_a^h|}{\tau_s}} \right] \quad (1)$$

where τ_s is the kernel function time constant, e is the exponential constant, t_i, t_a and t_d are the times of the input, actual output and target spikes respectively,.

According to this rule, the synaptic weight w is adjusted based on the precise time of the input, output and target spikes. Fig. 1 shows the configuration of the SNN network, for two classes, trained by the above rule.

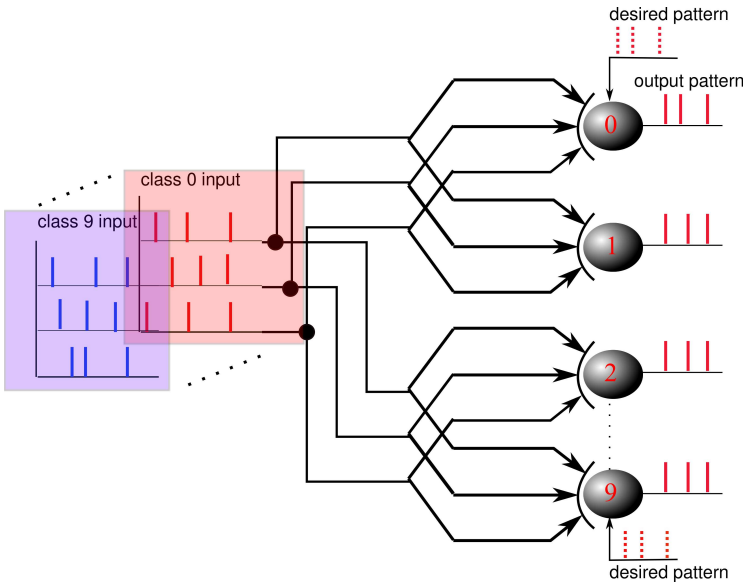


Fig. 1: The SNN topology used in this work. It is a single layer of spiking neurons, each neuron is trained to recognize a single digit.

The network could be used to classify multi-class spike patterns or to generate different spike trains. In [10], we have evaluated the network in classifying multiple categories of spike patterns generated artificially. The dataset was created by generating a number of template spike patterns, based on a uniform distribution. Then, using these templates many samples were generated by adding time delay jitters to the spikes of the templates. Therefore, it is likely this procedure will lead to a dataset that is linearly separable. Hence, in next section we evaluate SPAN on a more realistic dataset.

4 Learning Handwritten Digits Using SPAN

4.1 Description of the data

The MNIST handwritten digits, available from [7], is a well-known dataset used by many researchers to evaluate pattern recognition methods. Each image is 28×28 pixels. We use 200 sample images per digit (a total of 2000 images) for training and the network is tested on different 200 images per digit. Each sample digit is converted into spike pattern using the Virtual Retina [19]. The produced spike pattern consists of 784 spike trains. Fig.2 shows an example of the generated spike patterns for four digits. It can be noted that it is quite difficult to recognize the digit by only looking at the spike pattern.

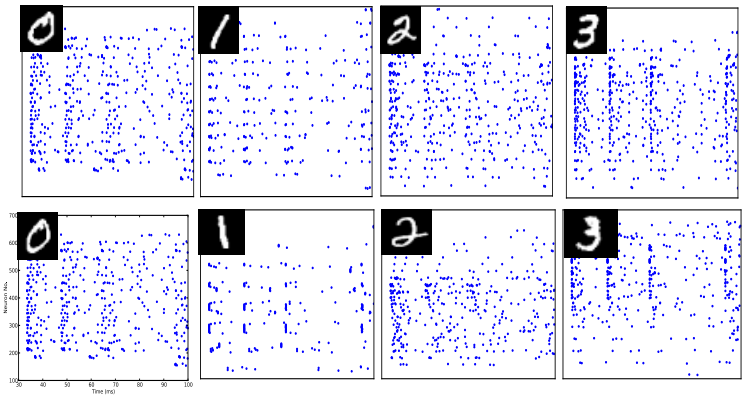


Fig. 2: Raster plots of the generated spike patterns by the Virtual Retina for the first four digits.

4.2 Experimental Setup

SPAN rule is used to train the network of Fig. 1 to learn the animated digit images. The network consists of ten neurons, each for one digit class. The weights are initialised randomly in the range [0.0, 5.0]. Each neuron has 784 synapses corresponding to (28×28) pixels of the input image. Therefore, there are 7840 synapses to be trained (784 synapse $\times$ 10 classes).

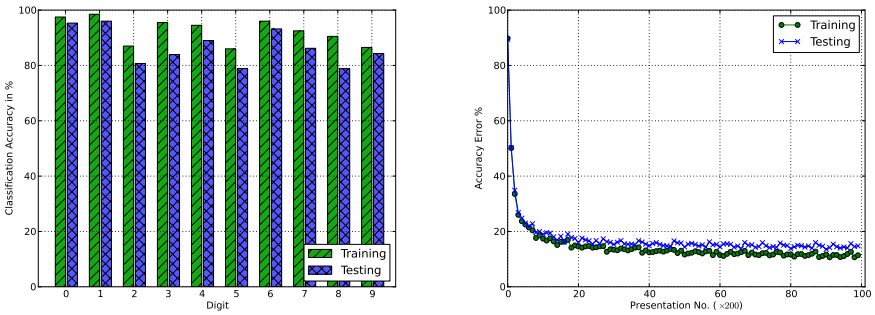
The neurons are Leaky Integrate-and-Fire (LIF) described by the following differential equation:

$$\tau_m \frac{du_i}{dt} = -u_i(t) + R\ I_i^{\text{syn}}(t) \tag{2}$$

where I_i^{syn} is the input signal current. The constant $\tau_m = RC$ is called the membrane time constant of the neuron and fixed to 10ms. Whenever the membrane potential u_i crosses a threshold $\vartheta = 20$mv, the neuron fires a spike and its potential is reset to a reset potential $u_{\text{reset}} = 0$mv. The learning rate (λ) in Eq. 1 is fixed to 0.01.

Each neuron is trained to produce a target spike train={25.,35.,45.,55.,65., 75.,85.,95.}ms selected randomly when a spike pattern from the assigned class is presented at the input and not to spike when patterns from other classes presented. In principle, different output spike patterns can be used for different digits.

The training is performed in 200 epochs, in each epoch the samples of each class (digit) are presented in random order. After each presentation of a training pattern, the synaptic weights of the neurons are updated according to Eq. 1. Thus, the training is performed in incremental mode [10] rather than batch mode, in which synapses are updated only after presenting the all training samples.



(a) (b)

Fig. 3: (**a**) The average accuracy obtained in the training and testing phase for the ten digits. (**b**) Evolution of the classification error, computed using 200 training and testing patterns, after each training sample presentation.

4.3 Results

We report the ability of the network to learn and classify the digit dataset in terms of classification accuracy. The classification accuracy on each digit class is defined as the number of patterns classified correctly over the total number of training(testing) patterns for that digit. Fig. 3a reports the obtained accuracy for the ten digits.

The network was able to learn to recognize the ten digits with an average accuracy of 92% in the training set and 86.6% in the testing set. Digit 8 has the minimum accuracy of 78.8% while the highest accuracy was obtained for digit 1 with a value of 96%, i.e., digit 1 has the most distinctive spike patterns. The obtained results confirm the ability of the network to classify the digit dataset with a good efficiency. We note that we have obtained the same generalization(testing) accuracy of 86% when the trained network was evaluated on more testing samples (10000 testing samples).

To understand the network better, the evolving of the network performance during training is investigated. During the training phase, the misclassification accuracy(error), computed on 200 images from the training set and another 200 images from the testing set, are recorded after each input presentation. There are 2000 training samples and 200 epochs which leads to 2000×200 presentations. However, we report the testing and training misclassification error in a step of 200 and up to 40000 presentations as shown in Fig. 3b. The figure shows the testing error curve is following closely the training curve. After about 4000 presentation , which is equivalent to two training epochs, the error curves start to flat and show very slow change. Thus, it is possible to train the network with less than 20 epochs to obtain good results. In fact, there should be a balance between the number of training epochs and number of training samples. Sufficient number of

training samples with few tens of epochs training are required for satisfactory training for this experiment.

5 Conclusion and Future Work

In this paper we have demonstrated the application of SNN trained with SPAN [10–12] on learning and classifying images of handwritten digits. One crucial factor in using SNN for real-world computer application is properly encoding the information into spike patterns. SPAN learning method is based on temporal coding, i.e., information is coded into the precise time of the spikes. Using the VR [19], it was possible to spike encode the digit images and classify the generated spike patterns efficiently. It is noted that the neurons are trained to produce the desired spike sequence in response to their class and not to fire for other classes. After training, the synaptic weights take positive (excitatory) and negative (inhibitory) values This means the neuron is learning its class and also learning to reject other classes through weights adjustment. It might be more desirable to design a mechanism that is based on inhibition to suppress the neuron firing, for example using a similar mechanism to "winner-takes-all" as it is in [1]. In addition, the used network consists of a single layer of spiking neurons and there are no specific features extracted for classification. Therefore, there is a space for more investigation to enhance the architecture of the network to achieve better results. Although the digit images are static data where other conventional methods can perform better, however after spike encoding the generated spike patterns are temporal data. The video data will take a similar form after spike conversion, indicating that the proposed method has also the potential to be applied for video signals with little modification to the algorithm, which will be our future work to investigate. Furthermore, SPAN learning will be enhanced for online learning and classification. We are also investigating the feasibility of SPAN implementation on a SNN chip scuh as the SRAM- based chip [13]. Such implementation will make it possible to use SPAN for a broad range of engineering applications based on the principle of embedded systems.

Acknowledgement

This work is supported by the Knowledge Engineering and Discovery Research Institute (www.kedri.info) of Auckland University of Technology. GL is also supported by an EU exchange grant from the University of Trento. NK is also supported by EU Marie-Curie IIF funded project 'EvoSpike' (http://ncs.ethz.ch/projects/evospike/.

References

1. Brader, J.M., Senn, W., Fusi, S.: Learning real-world stimuli in a neural network with spike-driven synaptic dynamics. Neural Comput. 19(11), 2881–2912 (Nov 2007)

2. Delorme, A., Gautrais, J., van Rullen, R., Thorpe, S.: Spikenet: A simulator for modeling large networks of integrate and fire neurons. Neurocomputing 26-27(0), 989 – 996 (1999)
3. Delorme, A., Thorpe, S.J.: Face identification using one spike per neuron: resistance to image degradations. Neural Networks 14(6-7), 795–803 (2001)
4. Floreano, D., Epars, Y., Zufferey, J.C., Mattiussi, C.: Evolution of spiking neural circuits in autonomous mobile robots: Research articles. Int. J. Intell. Syst. 21(9), 1005–1024 (2006)
5. Gerstner, W., Kistler, W.M.: Spiking Neuron Models: Single Neurons, Populations, Plasticity. Cambridge University Press, Cambridge, MA (2002)
6. Hopfield, J.: Pattern recognition computation using action potential timing for stimulus representation. Nature 376, 33–36 (1995)
7. LeCun, Y., Cortes, C.: The mnist database of handwritten digits (1998), `http://yann.lecun.com/exdb/mnist/`
8. Lichtsteiner, P., Posch, C., Delbruck, T.: A 128 x 128 120db 30mw asynchronous vision sensor that responds to relative intensity change. In: Solid-State Circuits Conference, 2006. ISSCC 2006. Digest of Technical Papers. IEEE International. pp. 2060 –2069 (2006)
9. Maass, W.: Networks of spiking neurons: The third generation of neural network models. Neural Networks 10(9), 1659 – 1671 (1997)
10. Mohemmed, A., Kasabov, N.: Incremental learning algorithm for spatio-temporal spike pattern classification. In: IEEE World Congress on Computational Intelligence , WCCI 2012. pp. 1227–1232. Brisbane, Australia (2012)
11. Mohemmed, A., Schliebs, S., Matsuda, S., Kasabov, N.: Method for training a spiking neuron to associate input-output spike trains. In: Engineering Applications of Neural Networks, EANN 2011. pp. 219–228. Corfu, Greece (2011)
12. Mohemmed, A., Schliebs, S., Matsuda, S., Kasabov, N.: Span: Spike pattern association neuron for learning spatio-temporal spike patterns. International Journal of Neural Systems 22(04), 1250012 (2012)
13. Moradi, S., Indiveri, G.: A vlsi network of spiking neurons with an asynchronous static random access memory. In: Biomedical Circuits and Systems Conference (BioCAS), 2011 IEEE. pp. 277–280 (2011)
14. Nikolic, D., Hausler, S., Singer, W., Maass, W.: Distributed fading memory for stimulus properties in the primary visual cortex. PLoS Biol 7(12), e1000260 (12 2009)
15. Panchev, C., Wermter, S.: Temporal sequence detection with spiking neurons: Towards recognizing robot language. Instruction, Connection Science 18, 1–22 (2006)
16. Thorpe, S.J.: Spike arrival times: A highly efficient coding scheme for neural networks. In: Eckmiller, R., Hartmann, G., Hauske, G. (eds.) Parallel Processing in Neural Systems, International Conference on. pp. 91–94. North-Holland: Elsevier (1990)
17. Uysal, I., Sathyendra, H., Harris, J.: Towards spike-based speech processing: A biologically plausible approach to simple acoustic classification. Int. J. Appl. Math. Comput. Sci. 18, 129–137 (June 2008)
18. Van Rullen, R., Gautrais, J., Delorme, A., Thorpe, S.: Face processing using one spike per neurone. Biosystems 48(1-3), 229–239 (November 1998)
19. Wohrer, A., Kornprobst, P.: Virtual retina: A biological retina model and simulator, with contrast gain control. Journal of Computational Neuroscience 26, 219–249 (2009)
20. Wysoski, S.G., Benuskova, L., Kasabov, N.: Evolving spiking neural networks for audiovisual information processing. Neural Networks 23(7), 819 – 835 (2010)