Pattern Recognition Spiking Neural Network for Classification of Chinese Characters

Pattern Recognition Spiking Neural Network for Classification of Chinese Characters

Nicola Russo[1], Wan Yuzhong[2], Thomas Madsen[1] and Konstantin Nikolic[1,2]

[1]University of West London - School of Computing and Engineering
St Mary's road, W5 5RF, London, UK

[2]Imperial College London - Institute of Biomedical Engineering
South Kensington Campus London SW7 2AZ, UK

Abstract. The Spiking Neural Networks (SNNs) are biologically more realistic than other types of Artificial Neural Networks (ANNs), but they have been much less utilised in applications. When comparing the two types of NNs, the SNNs are considered to be of lower latency, more hardware-friendly and energy-efficient, and suitable for running on portable devices with weak computing performance. In this paper we aim to use an SNN for the task of classifying Chinese character images, and test its performance. The network utilises inhibitory synapses for the purpose of using unsupervised learning. The learning algorithm is a derivative of the traditional Spike-timing-dependent Plasticity (STDP) learning rule. The input images are first pre-processed by traditional methods (OpenCV). Different hyperparameters configurations are tested reaching an optimal configuration and a classification accuracy rate of 93%.

1 Introduction

Spiking Neural Networks (SNNs) are widely used in computer vision, pattern recognition and other related problems [1, 2, 3]. However, implementing SNNs for image classification is more challenging than the traditional Artificial Neural Networks (such as Convolutional NNs) as they rely on a more complicated model of the neuron, which can simulate biological neurons more realistically [4, 5]. Hence, a question remains about the comparsions between the two types of NNs, regarding power efficiency, applicability (say for different lighting conditions), speed, needed computational resources and similar.

In many image classification studies, MNIST dataset is used to verify the classification accuracy of the algorithms [6], and serves as a benchmark for performance of various algorithms in machine learning. The MNIST dataset is a set of handwritten images of Arabic numerals 0-9 created by LeCun et al. [7]. In Diehl et al. work [8], for instance, MNIST dataset is used to train a SNN with (STDP) learning rule and lateral inhibition in an unsupervised way. Therefore we have chosen to use this type of approach and implement it on the case of Chinese characters classification.

In this work we aim to classify a specific set of eight Chinese characters, which have the meanings: up, down, left, right, forward, backward, exit, entrance (see Table 1). We generate our dataset, as described in the next section, and implement image preprocessing pipeline using the OpenCV library. We develop

Chinese Character	上	下	左	右	前	后	出	入
English translation	up	down	left	right	forward	backward	exit	entrance

Table 1: A selection of eight Chinese characters and their English translations.

a SNN, similar to one described in [8], with well defined learning rules as in reference [8], and implement an unsupervised learning process. The network is trained, tested and analysed using different hyperparameter configurations, in order to assess the optimal number of neurons in each layer, and how the number of pixels of the input image is affecting the classification accuracy.

2 Data set creation and Image pre-processing

Since we could not identify a suitable, publicly available dataset of handwritten Chinese characters which would be equivalent to the MNIST dataset, hence we have created a partial dataset. We did that by starting from a set of images of the handwritten characters (Table 1). Then we have performed several image processing tasks, specifically rotate all images for random angles, then perform three different degrees of dilation and erosion operations, and finally zoom to the specified size. In summary, for each Chinese character, we start from 13 font types, then have 20 different rotation angles, and 3 (dilated, eroded and original one) transformations. This is in total 780 images for each character, so for all characters, the whole data set consists of 6,240 (780x8) images.

Regarding the pre-processing part, it serves the purpose of greatly simplifying the data by eliminating irrelevant information in the images and enhancing the detectability of the information so that the SNN can accurately complete the classification task. The pre-processing algorithm pipeline performs median filtering and smoothing on the image to reduce the influence of background noise, then it operates on the pixel values of the image and quantises all the pixel values in the image to two values. The third step is to binarise the image. In the fourth step, the opening operation is performed to eliminate further the texture that may appear on the text pattern. Finally, the removal of irrelevant background and image scaling functions are applied.

3 SNN Model

The SNN model used in this study was proposed by Diehl et al. [8] and it is based on unsupervised learning using STDP. They proposed one network architecture and four learning rules performing classification training and result evaluation on the network using the MNIST data set. Due to the similarity of task types, this study adapts this network's topology, but then investigates a range of hyperparameters (such as different numbers of neurons), and then performs training for the Chinese character data set, and finally test the accuracy of the classification results.

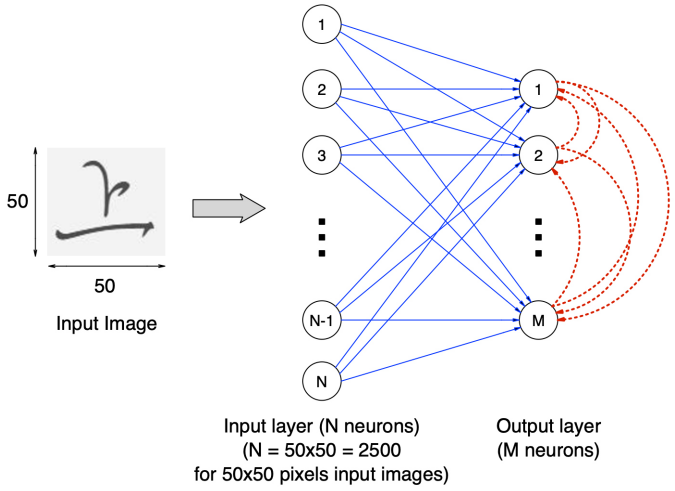


Fig. 1: The SNN topology. The full (blue) lines represent excitatory feed forward connections, and broken (red) lines are inhibitory recurrent connections.

3.1 Topology

The SNN used here has the characteristics of a competitive network (CN) where neurons in the output layer can laterally inhibit each other, and they are in constant competition. Most of the learning methods of CNs can discover the salient features of the input sample using unsupervised or self-organized learning, which are in general closer to *the brain network model* than supervised learning. The configuration of the SNN consists of two layers, see Fig. 1. The first one is the input layer with a number of neurons corresponding to the grayscale input image's pixels (N neurons for N pixels in the input image, e.g. $N = 2500$ for 50×50 pixel image). Each neuron encodes a pixel into a Poisson type fixed spike sequence (Details in Results Section), which simulates the generation mode of the neuron's spiking as the input information for the next layer. The rate of the Poisson spike sequence is proportional to the gray level of the pixel. The second layer is the output layer, consisting of M neurons, where one or more neurons will eventually represent each character.

The connectivity map is the following: (a) input layer to output layer - all to all (Fig. 1, blue lines), excitatory synapses, (b) output layer to output layer - all to all (without self-connections, Fig. 1, red lines) with inhibitory synapses. This kind of connection provides the effect of lateral inhibition generating competition between output neurons. The neuron model in this network is the Leaky Integrate-and-Fire (LIF), and the model for the synapses is the Conductance Based (COBA) model, described in [9]. The training process applies only to the synapses between input and output layer (blue/full lines in Fig. 1). The weights of the inhibitory synapses (red/broken lines) are fixed, and do not change in the training process. The maximum conductance of the inhibitory synapses is set to 10 nS to ensure that the lateral inhibition is neither too weak nor too strong. The value of 10 nS was chosen empirically.

3.2 The Learning Rule

The network combines synaptic plasticity and threshold plasticity to achieve competitive learning. For the synaptic plasticity we use a variation of the STDP learning rule. A synapse S_{ji} between presynaptic neuron i and postsynaptic neuron j updates whenever any of these two neurons spike. When the postsynaptic neuron (j) spikes at the moment t, the weight update formula is [8, 10]:

$$\Delta w = \eta_{post}(x_{pre} - x_{tar})(w_{max} - w)^{\mu} \tag{1}$$

where η_{post} is the learning rate, w is the synaptic weight, w_{max} is the maximum weigh, and μ determines the degree of the weight dependence [8]. The term x_{pre} is the spike timing factor for the presynaptic neuron i spikes before t. Whenever i spikes at t_i^{new}, we update the value $K_i \leftarrow K_i \cdot \exp(-(t_i^{\text{new}} - t_i^{\text{old}}))/\tau) + 1$, where t_i^{old} is the time of the previous spike of the neuron i (we also update $t_i^{\text{old}} \leftarrow t_i^{\text{new}}$) [10]. When the postsynaptic neuron spikes at t, we have $x_{pre} = K_i \cdot \exp(-(t - t_i^{\text{old}})/\tau)$. The x_{tar} is some target presynaptic trace value, which lowers the synaptic weight and consequently reduces the impact of weak synapses.

When the neuron i spikes at t, the weight update formula is [8, 10]:

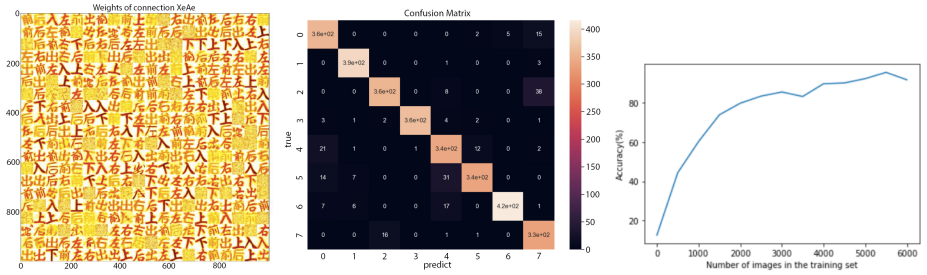
$$\Delta w = -\eta_{pre} x_{post} w^{\mu} \ , \tag{2}$$

where η_{pre} is the learning rate for the synaptic depression, and x_{post} is the postsynaptic neuron j trace, generated similarly as explained for x_{pre}.

In addition, we also use threshold voltage plasticity learning. Specifically, the membrane potential threshold of each output neuron depends not only on v_{thres}, but also on a factor θ. θ is similar to the concept of the trace introduced above. Each time a neuron emits a spike, θ is increased by a fixed value, and decays exponentially during the rest of time. Therefore, the more one neuron fires, the higher its threshold, making it more difficult for the neuron to fire the next spike in quick succession. This mechanism imitates neuron's adaptation to the sustained input current, and effectively limits the firing frequency.

4 Results

The network is trained with 50x50 pixel resolution pre-processed images. For the output layer, the number of neurons varies from 100 to 625. Considering 400 neurons for the output layer, the network has 1 million excitatory synapses linking the input with the output layer, and 159,600 inhibitory synapses linking each output neuron with each other (400x399=159,600). Each image is presented to the network for 350 ms with a delay of 150 ms between two samples. At the beginning of training, the number of spikes emitted by each output layer neuron is irregular (random in time). As the training progresses, each neuron's receptive field gradually learns a fixed input pattern, showing a certain regularity. Only a few specific neurons emit dense spikes, and the remaining neurons hardly emit any spikes. This phenomenon can be established by observing the synaptic weights distribution after training, shown in Fig. 2a. In the weight



(a) (b) (c)

Fig. 2: (a) Weight matrix for $N = 400$ output neurons, presented in $20 \times 20 = 400$ blocks, each block has $50 \times 50 = 2500$ weights for the synapses between the input neurons and one output neuron. Some weights blocks resemble input characters. Training set: 5000 images, (b) Confusion matrix for predictions. (c) Accuracy with different number of images in the training set.

matrix obtained by using 5,000 pictures to train the network, each square represents connections to an output layer neuron, and it is composed of the weights of the synapses to 2,500 input neurons.

To test the network, each output neuron needs to be assigned a label. This is an intermediate step, which could be classified as a "supervised" process and comes after the unsupervised learning phase has created distinctive outputs for each input image. In the process of assigning labels the weights and adaptive thresholds generated after training are fixed and the network is set, then the testing set images are sequentially input to the network.

Assuming the number of training samples is 5,000, a 5000x400 spike matrix can be obtained. Each column represents the firing response of the corresponding neuron to the training images, and each image represents one of the eight Chinese characters. Therefore, by averaging the number of spikes generated by each type of Chinese character, the category with the most significant average value represents the category of the neuron. After each output neuron was assigned with one of the eight labels, we tested the network on a set of 1,000 symbols not previously seen by the network. We have achieved 92.85% accuracy when the training set had 5,000 symbols.

The confusion matrix of the prediction result is shown in Fig. 2b. Index 0 to 7 in the matrix correspond to the 8 Chinese characters. Analysing the results, the characters '左' and '右' (indices 3 and 4), are similar. The character '后' also has a certain similarity with the character '左' and '右'. We can observe that the character 4 has been correctly predicted in 89% of cases, but in 6.2% of cases it is labelled as character 0 and 3.6% as the character 5. For character 5, it is correctly predicted in 85% of cases, wrongly predicted in 4.2% as character 0 and 9.1% as character 4. Therefore, these characters are prone to some small classification errors due to similar shapes, shown in the confusion matrix. After increasing the number of images in the training set, we found that this mismatch rate dropped sharply, indicating that for some neurons that generate strong firing response for

Neurons number	100	225	324	400	625
Accuracy(%)	87	89.8	91.7	92.85	93.04

Table 2: Accuracy with different output layer sizes

the '上' image, their weight matrices are not trained enough. Fig. 2c shows how the accuracy of classification task increases as the number of training set images increases. We have observed that when the number of images is above 5,000, the classification accuracy of the network tends to a stable level, which is around 93%. We have also investigated the effect of the network hyperparameters on the classification results. Table 2 shows how the number of output neurons affects the accuracy of the classification. For $M = 100$ output neurons we already have accuracy of 87%, and increasing the number of output neurons (and synapses) six fold (i.e. to $M = 625$), the accuracy increases for about 6%.

5 Conclusions

This paper introduces a SNN model for the Chinese character image classification task, based on unsupervised learning STDP. The process of developing the model consisted of three main tasks: creating a Chinese character dataset, designing a pre-processing pipeline, and building, training and testing an SNN. The Network hyperparameters (e.g. number of neurons) were tested, reaching an optimal configuration with 50x50 input size images and a 400-neuron output layer. The maximun performance was achieved when the network is trained with minimum 5,000 samples (or about 600-700 per character), reaching an accuracy of ∼93%.

References

[1] Schliebs S, Kasabov N. "Computational Modeling with Spiking Neural Networks", Springer Handbook of Bio-Neuroinformatics. Springer Berlin Heidelberg, 2014:625-646.

[2] Russo N, Huang H, and Nikolic K "Live Demonstration: Neuromorphic Robot Goalie Controlled by Spiking Neural Network", IEEE Biomedical Circuits and Systems Conference (BioCAS), 2022: 249-249.

[3] Russo N, Huang H, Donati E, Madsen T, and Nikolic K. "An interface platform for robotic neuromorphic systems", Chips, 2023, 2(1): 20-30.

[4] Peter O, Daniel N, Liu S C, et al. "Real-time classification and sensor fusion with a spiking deep belief network", Frontiers in Neuroscience, 2013, 7(7):178.

[5] Diehl P U, Neil D, Binas J, et al. "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing", International Joint Conference on Neural Networks. IEEE, 2015:1-8.

[6] Deng L. "The MNIST database of handwritten digit images for machine learning research", IEEE Signal Processing Magazine. 2012, 29(6):141–2.

[7] LeCun Y, Bottou L, Bengio Y, and Haffner P. "Gradient-based learning applied to document recognition." Proceedings of the IEEE, 86(11):2278-2324, November 1998.

[8] Diehl P U, Cook M. "Unsupervised learning of digit recognition using spike-timing-dependent plasticity", Frontiers in Computational Neuroscience, 2015, 9:99.

[9] Vogels T P and Abbott L F. "Signal Propagation and Logic Gating in Networks of Integrate-and-Fire Neurons", The Journal of Neuroscience. 2005.

[10] Morrison A, Aertsen A, Diesmann M. "Spike-Timing-Dependent Plasticity in Balanced Random Networks", Neural Computation (2007) 19 (6): 1437–1467.