Learning weights with STDP to build prototype images for classification

Ajay Vasudevan, Teresa Serrano-Gotarredona, Bernabe Linares-Barranco
Instituto de Microelectrónica de Sevilla (IMSE-CNM), CSIC and Universidad de Sevilla, Sevilla, Spain
Email: ajay@imse-cnm.csic.es, terese@imse-cnm.csic.es, bernabe@imse-cnm.csic.es,

***Abstract*—The combination of Spike Timing Dependent Plasticity (STDP) and latency coding used in a spiking neural network has been shown to learn hierarchical features. In this paper we propose a new way to classify images using an SVM. Prototype images are built from the weights learned in an unsupervised manner using STDP. The prototype images are cross correlated with the input image and the peak of the cross correlation with each prototype image is used as additional features for an SVM. The network, demonstrated on the MNIST data set, achieves 99.15% testing accuracy which is the best reported accuracy for a SNN with unsupervised training.**

***Index Terms*—STDP, Image Classification**

I. Introduction

Visual object recognition in the primate brain is possible through largely feed forward computations [1]. Methods to develop computational visual object recognition has drawn inspiration from the hierarchical feed forward processing of the primate brain [3]. While the state of the art of such methods shows high performance in terms of recognition accuracy [4] they have some fundamental differences from the primate visual cortex. In order to avoid over fitting, these models need to be presented with a large database of examples which are labelled which is in contrast with primates where learning occurs with fewer examples and mostly unlabeled data. These Deep Convolutional Neural Networks (DCNNs) use back-propagation for training of weights which has no evidence of a biological basis.

The feed forward computations in the human brain are done in the estimated 1.6 – 4 billion neurons of the visual cortex [2]. In spite of the large number of neurons involved in the speed of processing, a short time period of 100-150 ms appears to be enough to process an image [5]. The processing in the human brain has been shown to be accurate better than random chance even when the input images are presented for a very short duration of 13 ms without inter stimulus intervals [6].

In order to understand the remarkable speed and accuracy of the primate visual cortex, it is essential to study spiking neural networks where information between two neurons are passed in the form of spikes. Recent research into spiking neural networks can be classified into 3 types

1) Direct supervised learning
2) Artificial neural network converted to work in the spiking domain
3) Unsupervised learning in the spiking domain (like STDP)

Methods for direct supervised learning(1) in the spiking domain include coding in the time of spike to have a differentiable relationship with a subset of previous spikes and hence compatible with the gradient descent back-propagation rule [7]. Other methods include approximating the spiking activity to be differentiable [8].

Conversion from a pre-trained ANN model (2)- CNN units can be translated into biologically inspired spiking units with leaks and refractory periods [9]. Spiking equivalents of DCNN operations can be used which allow the conversion of any arbitrary DCNN network to the spiking domain [10]. They report the best performance of 0.54% error rate on the MNIST database for a supervised learning SNN.

In this paper we present an SNN with unsupervised learning (3). Unsupervised learning with STDP has been demonstrated with both rate based poisson coding and latency based one spike per neuron coding [11] [12]. The use of biologically plausible neuron and synapse models in a 2 layer network has shown 95% test accuracy on the MNIST data set [11]. Convolution and pooling layers in cascade with the network performing layer by layer learning to learn hierarchical features has shown the best performance reported to date for an unsupervised learning SNN [12] [15]. Once the weights of the neurons are trained using STDP, for classification the learning rate can be set to zero and each neuron assigned to the class which gives highest response [11]. The use of a fully connected (FC) layer at the output has also been demonstrated [15]. Alternatively, after the training is done, the threshold can be set to infinity and the maximum potentials of the neurons in the final layer can be used as a feature vector for a Support Vector Machine (SVM) [12]. In this work, we propose to include cross correlation values in the feature vector along with the final potential values of the neurons of the final layer. We demonstrate that the inclusion of these features increase the network performance.

II. Methods

The network used contains a coding layer, convolutional layers and a pooling layer in cascade. The architecture of the network used is shown in Fig. 1. Learning happens only in the convolutional layers and all potentials are set to zero before every pass of an image.

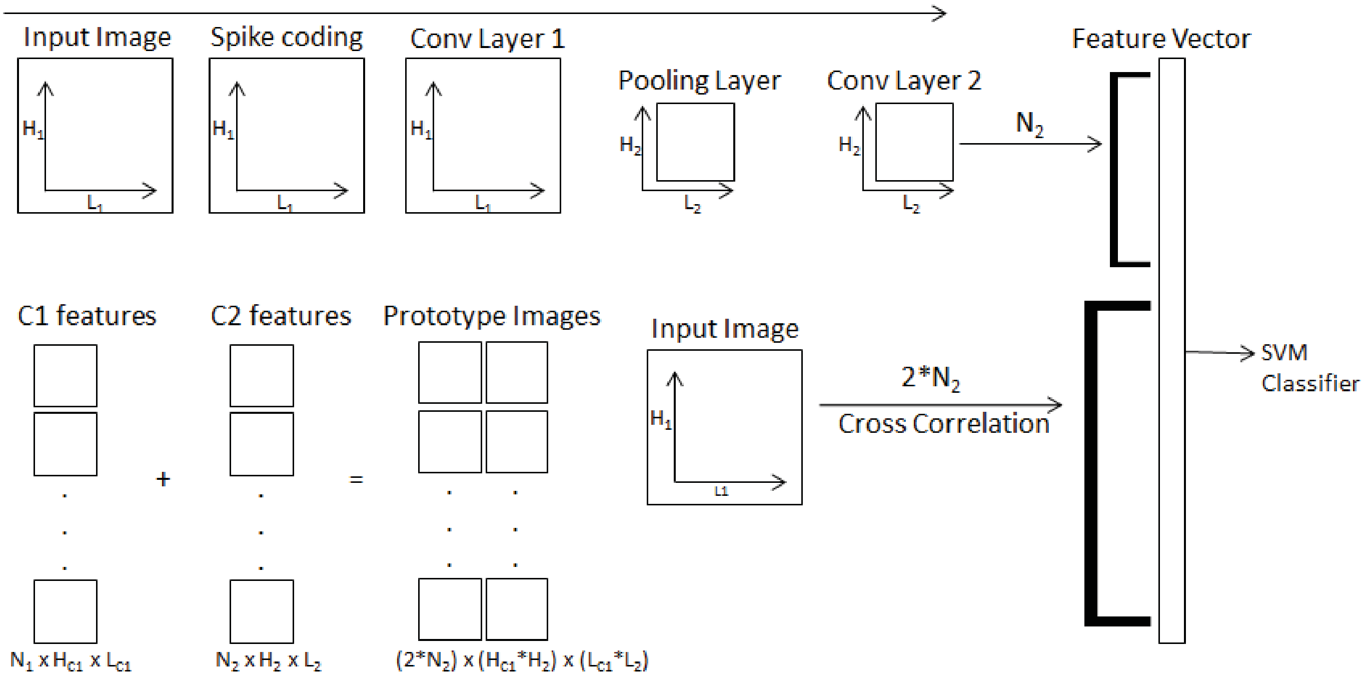

Fig. 1. Network has 2 convolution layers and one pooling layer. The input image is encoded in spikes by the Difference Of Gaussian(DoG) layer and passed to the first convolution layer. Learning happens only in the convolutional layers and layer by layer with Conv Layer 1 learning low level features. The type of feature learned is dictated by the size of the layer weight maps. The size of C1 weight maps ($H_{C1}xL_{C1}$) is smaller than the size of C2 weight maps (H_2xL_2) which is the same size as Conv Layer2. The pooling layer passes along only the first incoming spike in a window of the first convolutional layer thereby compressing information. After the convolution weights of C1 and C2 layers are learnt, for each of the N_2 maps there are 2 prototype images generated. Cross convolution of the input image with the prototype images and the maximum potential of the learned C2 maps together form the feature vector that goes into the SVM for every input image.

A. *Temporal coding*

The input image is filtered with a difference of gaussian (DoG) filter in order to detect the spatial features at every location of the image. The higher activation value indicates the presence of a stronger contrast in that image location. Similar to [12] we use an inverse relation to generate spikes in the temporal domain so that a neuron with higher activation fires first. There is a threshold set to allow a neuron to fire. A set number of discrete time steps is used. The neurons with activation value above the firing threshold fire one spike each with a spike time between 1 and $N_{timesteps}$. Both ON and OFF center DoG maps are used to get both positive and negative i.e. solid and edge contrasts. This coding scheme ensures that there is only one spike per neuron propagated and the information is encoded in the time of the spike.

B. *First Convolution Layer*

This layer learns from spikes generated from both the ON and OFF center DoG maps in the coding layer. The weight maps are of size 5 x 5 and each map learns the same feature but at different locations due to the use of weight sharing i.e. once a map learns a feature, the learned map is updated for all locations. The neurons in this layer are leaky integrate and fire neurons where at every time step if a pre synaptic neuron fires, the weight of that pre-synaptic neuron is added to the potential of the post synaptic neuron. The leak is equal to one hundredth of its potential and this small leak ensures that random noise does not have an impact and also once the neurons are in an advanced stage of learning, this small leak ensures that the effect of stray spikes is minimal.

$$V_i(t) = 0.99 * V_i(t-1) + \sum_j w_{ij} S_j(t-1) \tag{1}$$

where V_i is the potential of the ith neuron at time step t. w_{ij} is the synaptic weight between the jth presynaptic neuron and the ith post synaptic neuron and $S_j(t\text{-}1)$ takes a value 1 if the jth presynaptic neuron has fired at time t-1.

All neurons in the map are initialised with uniform weights of value 0.8. At every time step, the neurons with potential greater than or equal to the threshold are allowed to fire. In case of multiple neurons with potentials above the threshold, only the neuron with the greatest potential is allowed to fire. In the case that multiple neurons have the exact same potential, the first indexed neuron is allowed to fire to limit the network to one spike per neuron.

Once a neuron crosses the potential threshold and fires, the weights are updated with the simplified STDP rule for pre and post synaptic neuron j and i:

$$\Delta w_{ij} = \begin{cases} c.a^{+} w_{ij}(1-w_{ij}), & \text{if } t_j - t_i \leq 0 \\ c.a^{-} w_{ij}(1-w_{ij}), & \text{if } t_j - t_i > 0 \end{cases} \tag{2}$$

$$c = (N_t - t_i + t_{i,prev} + 1)/N_t \tag{3}$$

where $t_i$and t_j refer to the time of spike of the post and pre synaptic neurons. $t_{i,prev}$ is the previous time step at which neuron *i* spiked. N_t refers to the number of time steps used per image presentation. a^+ and a^- are fixed parameters that scale the positive and negative weight change in the STDP rule.

Once a set of weights is updated, the potentials of all neurons of the same map are reset to zero. The potentials of the firing neuron and 2 neurons surrounding it in all directions, are reset to zero and the weight change shared with all neurons of the same map. The feature weights are allowed to be updated again should any neuron potential cross the firing threshold with the updated weights. In order to differentiate between weight changes that occurs at latter time steps, a constant *c* is used in the STDP formula to scale the weight update in relation to the time step at which it occurs. Supposing a weight update occurs for the first time at the last time step, the weight change is multiplied by a factor of $(N_t$-t_i+0+1)/N_t = 1/N_t. But if the same set of weights had previously been updated at timestep 5, then the weight update at the last timestep is scaled by a factor of $(N_t$-t_i+5+1)/N_t = 6/N_t. The value of a^+ is set to 0.4 and a^- is set to -0.3. By allowing the feature weights to be updated more than once, all information in an image is learnt while the earlier updates have more weightage.

Since each feature learnt has two sets of weights for the ON and OFF center DoG responses, when a feature weight is updated the contributions of the ON and OFF center maps are updated separately. The locations of a pre spike get a positive weight change while other locations irrespective of a post spike or no spike get a negative weight change. Since the temporal coding scheme ensures that only one of the two maps spike per neuron, this means that the location of positive weight change update in one is a negative weight change in the other set of weights of the same map.

Learning occurs layer by layer and only in the convolution layers. The learning in the first convolutional layer is stopped when the convergence value falls below 0.01 [12]. The convergence C at time t is

$$C_t = \sum_f \sum_i w_{f,i}(1 - w_{f,i})/n_w \tag{4}$$

where n_w is the number of weights and there are *i* weights in the *f* features.

C. Pooling Layer

The pooling layer performs an operation where only the first incoming spike in a window is propagated. The pooling layer effectively compresses the data to be presented to the next convolution layer and introduces a degree of translation invariance.

D. Second Convolution Layer

This is the final convolutional layer consisting of features which are the same size as the pooling layer output and learning happens by the same simplified STDP rule used for the first convolution layer. But, in this layer only one feature is allowed to learn per input presentation and only once. In this way, there is competition to learn the total representation of the image.

E. Classification

Classification is done by SVM. The feature vector used is made up of two components

1) The threshold of the second convolution layer is set to infinity and the maximum of the final potential reached as a fraction of the sum of weights of each feature map is used as the feature value of the map. There will be N_2 number of these features.
2) The peak values of cross correlation of input image with prototype images are used as additional features. There will be 2*N_2 number of these features.

For each feature map in the last convolution layer, 2 prototype images are generated solely from the learned weights of the two convolution layers.

1) *prototype_max:* For each N_2 neuronal map, at each location there are N_1 weights. This image is formed by multiplying the maximum of these weights at each location with the corresponding C_1 weight map.
2) *prototype_sum:* For each N_2 map at each location multiplying and summing all N_1 weights with its corresponding C_1 weight map.

So, for a feature of size H_2*L_2, the prototype images will be of size (H_2*H_{C1}) x (L_2*L_{C1}) where H_{C1} and L_{C1} are the sizes of the features of the first layer. The *prototype_sum* images will lie in a greater range because of the summing of weights, while *prototype_max* images will lie in the range 0 to 1. Each image in the training set will have 3*N_2 number of features and this N_{imgs} X 3*N_2 vector is trained by the SVM for classification.

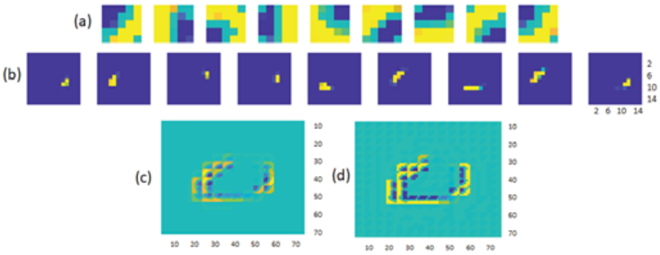

Fig. 2. Illustrating prototype images. (a) Learnt L1 weights of size 5x5 for all the C1 feature maps (b) Learnt L2 weights of size 14x14 for a particular C2 feature map (c) Corresponding 'Prototype-max' image (d) Corresponding 'Prototype-sum' image.

III. Results

We applied the network on the full MNIST dataset which contains 60,000 images in the training set and 10,000 images in the test set [14]. We obtained an accuracy of 99.15% for the test set. The network used was 28x28x2 - 9C5 - 2P - 200C14 where C describes a convolution layer and P is a pooling layer.

A. DoG layer and first convolutional layer

We used both ON and OFF center DoG to generate spikes. We used a spiking threshold of 1/10th of the maximum activation. The spike times which have an inverse correlation to the activation value were scaled between 1 and 15 timesteps. The number of spikes at each timestep is not fixed and it depends on the number of activation values corresponding to that discrete time step. At most one spike is generated per pixel.

For the first convolutional layer we use 9 features of size 5x5. The firing threshold used is 10. As described earlier, each feature is allowed to learn more than once per image presentation. The learned features indicate that at his level the edge features are learnt. Fig 2. shows the learned features with the contribution of the ON and OFF center DoG spikes.

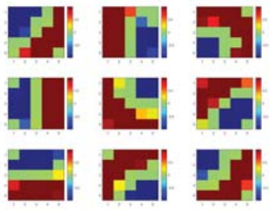

Fig. 3. C1 features showing the learned weights of both the ON and OFF center DoG spikes.

B. Pooling layer and second convolutional layer

The output of the first convolutional layer is spikes from 28x28 neurons. The pooling layer used is of size 2x2 and stride 2. It effectively halves each input dimension to the second convolutional layer by passing along only the first incoming spike in a 2x2 window.

In the second convolutional layer there are 200 N_2 features that take input spikes from a 14x14x9 input. The weights in this layer are also initialised with a uniform value of 0.8. In this layer since each convolution window is of size H_2*L_2=14x14, each feature is learning on the entire image as opposed to learning any feature in a window of an image such as edges which were learnt in the first convolution layer, hence each feature learns only once per image and only one feature learns per image presentation. Similar to the first convolution layer, learning is finished when the convergence value is less than 0.01.

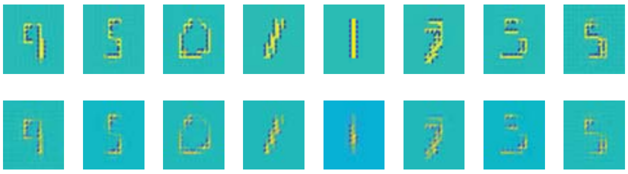

Fig. 4. Examples of Prototype images

C. Classification

After the weights are learnt in an unsupervised way with STDP. We pass each image of the training set through the network to make features to use for the SVM. Since SVM is inherently two class, for our multi class problem we found that the *one vs all* method worked better than the *one vs one* classification. For each image we generate 600 features as described in the methods section. For the cross correlation features, we find the cross correlation of the positive and negative contrasts separately and sum them, from which the peak value is stored as the feature. For cross correlation, the input images are scaled to match the size of the prototype images.

Our preprocessing steps include

1) Each maximum final potential feature is divided by the sum of its weights so that this feature indicates the final potential reached as a fraction of the maximum possible value for that feature.
2) Feature dropout. For each digit label, 15 features are dropped from the total 600 features. This is split as 5 features from the maximum final potential features and 5 each from the cross correlation features.
3) The 585 features for an image are standardized to have a zero mean and unit variance separately for the maximum final potential features and the cross correlation features.
4) The data is then standardized across features to lie in a distribution with zero mean and unit variance.

For each digit the features are activated differently and from the range of feature values of all examples of a digit, the features with the lowest maximum activation values for each digit are dropped. This is to ensure that the features that are not activated to a high level do not contribute to the classification for that digit. The data scaling first across instance and then across dimension means that data values roughly lie in the same range and no feature will dominate the classification because features with large dynamic range will dominate the separating hyperplane.

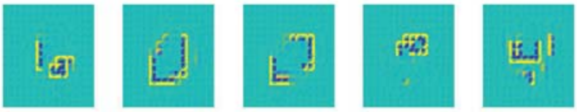

Fig. 5. Examples of features dropped for classifying digit 1.

We used the *fitcsvm* function in MATLAB with optimized hyper-parameters *kernel* ='rbf' *KernelScale* = 8 and *BoxConstraint* = 26. The SVM training takes approximately 190 seconds on an Intel i7 processor with 6 cores. For the test dataset, the second part of scaling that is scaling across features is done with the mean and standard deviation of that feature in the training dataset [13]. This is to ensure that the range of data values is with respect to the training data.

IV. Analysis

Our testing accuracy of 99.15% for the full MNIST dataset is the best reported accuracy for a spiking neural network with unsupervised learning of weights with STDP. The better classification accuracy with the cross correlation features indicate that the translation invariance in the network is improved and the building of prototype images with the final weights shows good generalization of images in the dataset. Cross correlation which is correlation at different locations of the image can be easily implemented in hardware.

We found that the number of features used in the second convolution layer has a greater impact on performance than the number of features in the first layer. This indicates that there is greater variance in the higher level features than the lower level features which are edges. The performance improvement using dropout is noticed when dropping only a minimal number of features. This shows that only a few features learnt are non selective.

TABLE I
EFFECT OF DROPOUT ON ACCURACY

Dropout Level	Accuracy
No Dropout	99.09%
15 features	99.15%
30 features	99.08%

Since our network is limited to maximum one spike per neuron, it is very energy efficient. Our network generates on average 600 spikes per input pass.

While the spike times in the network are used for STDP, after training the time of threshold crossing in the final layer remains unused. It is possible to use a latency coding scheme and built a network to apply back propagation and stochastic gradient descent [7]. We attempted to apply that network using the time when the threshold is crossed in the last convolution layer encoded as the spike times of the input layer neurons. Since there is not a big separation in the times of spikes and the spike times are non binary, we could not improve the performance using this training scheme.

TABLE II
COMPARISION OF ACCURACY ON MNIST WITH SNNS USING UNSUPERVISED LEARNING OF WEIGHTS

	Network	Classification	Accuracy
Diehl et al 2015	2layer FC	Heuristic	95%
Kheradpisheh et al., 2016	DCNN	SVM	98.40%
Panda et al., 2016	DCNN	FC	99.08%
This work	DCNN	SVM	99.15%

V. ACKNOLEDGEMENTS

This work was funded by EU H2020 grants 687299 ”NEURAM3” and 824164 ”HERMES”, and by Spanish grant from the Ministry of Economy and Competitivity TEC2015-63884-C2-1-P (COGNET) (with support from the European Regional Development Fund).

REFERENCES

[1] James J DiCarlo, Davide Zoccolan, and Nicole C Rust. How does the brain solve visual object recognition? Neuron, 73(3):415–434, 2012.

[2] Leuba G; Kraftsik R. Changes in volume, surface estimate, three-dimensional shape and total number of neurons of the human primary visual cortex from midgestation until old age, Anatomy and Embryology, 351-366, 190,1994

[3] Y LeCun and Y Bengio. Convolutional networks for images, speech, and time series. In The Handbook of Brain Theory and Neural Networks, pages 255–258. Cambridge, MA: MIT Press, 1998

[4] Alex Krizhevsky, I.Sutskever, and G.E. Hinton. Imagenet classification with deep convolutional neural networks. In Neural Information Processing Systems (NIPS), pages 1–9, Lake Tahoe, Nevada, USA, 2012.

[5] Simon Thorpe, Denis Fize, Catherine Marlot, et al. Speed of processing in the human visual system. Nature, 381(6582):520–522, 1996.

[6] Potter, Mary C., Brad Wyble, Carl Erick Hagmann, and Emily S. McCourt. “Detecting Meaning in RSVP at 13 Ms Per Picture.” Attention, Perception, Psychophysics 76 no. 2 (December 28, 2013): 270–279.

[7] H. Mostafa, ”Supervised Learning Based on Temporal Coding in Spiking Neural Networks,” in IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 7, pp. 3227-3235, July 2018.

[8] Wu Yujie, Deng Lei, Li Guoqi, Zhu Jun, Shi Luping. Spatio-Temporal Backpropagation for Training High-Performance Spiking Neural Networks. Frontiers in Neuroscience, 12, 2018 331

[9] Pérez-Carrasco, José Antonio, Bo Zhao, Carmen Serrano, Begoña Acha, Teresa Serrano-Gotarredona, Shoushun Chen and Bernabe Linares-Barranco. “Mapping from Frame-Driven to Frame-Free Event-Driven Vision Systems by Low-Rate Rate Coding and Coincidence Processing–Application to Feedforward ConvNets.” IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (2013): 2706-2719.

[10] Rueckauer Bodo, Lungu Iulia-Alexandra, Hu Yuhuang, Pfeiffer Michael, Liu Shih-Chii. Conversion of Continuous-Valued Deep Networks to Efficient Event-Driven Networks for Image Classification. Frontiers in Neuroscience, 11, 682, 2017

[11] Diehl Peter, Cook Matthew. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. Frontiers in Computational Neuroscience. 9, 99, 2015

[12] Kheradpisheh, S.R., Ganjtabesh, M., Thorpe, S.J., Masquelier, T., STDP-based spiking deep convolutional neural networks for object recognition. Neural Networks (2017).

[13] Hsu, C.W., Chang, C.C., Lin, C.J., ’A Practical Guide to Support Vector Classification’ , Technical report, Department of Computer Science, National Taiwan University, 2003.

[14] Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998

[15] Panda, P., and Roy, K. Unsupervised regenerative learning of hierarchical features in Spiking Deep Networks for object recognition. 2016 International Joint Conference on Neural Networks (IJCNN), 299-306, 2016.