STDP-based spiking deep convolutional neural networks for object recognition

Supervisor

Simone Scardapane

Students

Edoardo Ghini

Gianluca Cerilli

Contents

Introduction

Second generation artificial neural networks – mainly characterized by fully connected layers that deal with continuous output values – have allowed us to make enormous progresses in many relevant scientific fields. In the same way, also the third generation of artificial neural networks seems to be very promising and it is expected to bridge the gap between neuroscience and machine learning, by using more realistic *spiking* neuron models. These 3rd generation networks are called "Spiking Neural Networks"and are different from usual networks, since they do not deal with continuous values, but they communicate through spikes, that are discrete events that occur at certain points in time. In fact, when a (membrane of a) neuron reaches a certain potential value, it spikes, so it resets its value. This behaviour is inspired by real biological processes and is expected to better model reality. This may seem not so "innovative", since discrete signals are used instead of continuous ones. Instead, this modelling allows us to handle spatio-temporal data, that are real-world sensory data. These spatial and temporal aspects refer to the fact that neurons are connected to neurons local to them and that temporal information of the spikes (that are lost in binary encoding) are now preserved. Indeed, it has been proven that spiking neurons are computationally more powerful than traditional artificial neurons.

In this report we describe a project in the frame of the "Neural Networks"course of Sapienza Università di Roma, that is inspired from the "STDP-based spiking deep convolutional neural networks for object recognition"paper of Kheradpisheh, Ganjtabesh, Thorpe and Masquelier. In particular, the implementation of a Spiking Convolutional Neural Network for object classification is described.

1 Network architecture

An STDP-based spiking deep neural network (SDNN) with a spike-time neural coding has been implemented. In the very beginning of this network, the input image is passed to the temporal-coding layer,that uses Difference of Gaussians (DoG) filters to convert the image into an asynchronous spike train. This is used as input for the successive layer, that is a convolutional one, in which input spikes are integrated and new spikes are emitted if a certain threshold is exceeded, that is when a visual feature is detected. Learning is done through STDP, that is used to learn visual features, and so the weights of the network. Neurons that fire earlier in this phase prevent the others from firing. The convolutional layer is followed by a pooling one, where the visual information is

compressed. Three convolutional and pooling layers (arranged in an alternate and consecutive order) have been implemented to handle larger and more complex visual features through the network. In the end, in the last pooling layer, a global max pooling is performed for the classification, to train a linear classifier. The results are used to find the proper category to which the input image belongs.
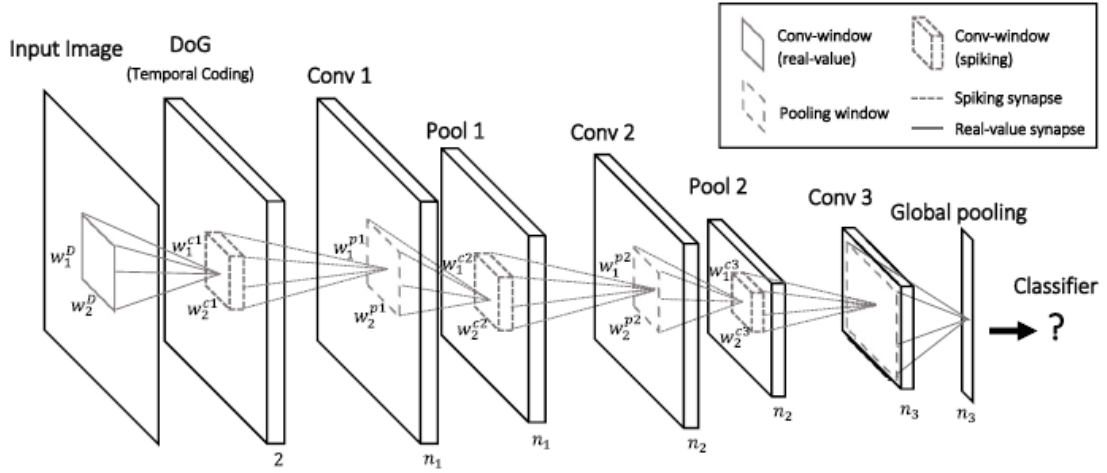

Figure 1: SDNN architecture

1.1 Temporal coding

In the very first part of the neural network, the input signal is encoded into temporal-discrete spike events. A Difference of Gaussians (DoG) filter is used to detect the contrasts in the input image and emit a spike, accordingly. The higher the contrast in a certain cell, the more strongly this one is activated in order to fire. The firing time (i.e. the time when the spike is emitted) of a DoG cell is inversely proportional to its activation value. In particular, if an r output is obtained in a cell after DoG, its corresponding firing time will be $\tau = 1/r$. DoG cells are sensitive to both positive and negative contrasts, depending on their two ON-centre and OFF-centre maps and they emit when their activation values exceed a predefined threshold.

An example of DoG filter output is shown in figure 2. In this picture, a face and a motor (image categories that compose the learning and training dataset) are encoded into discrete spike events.
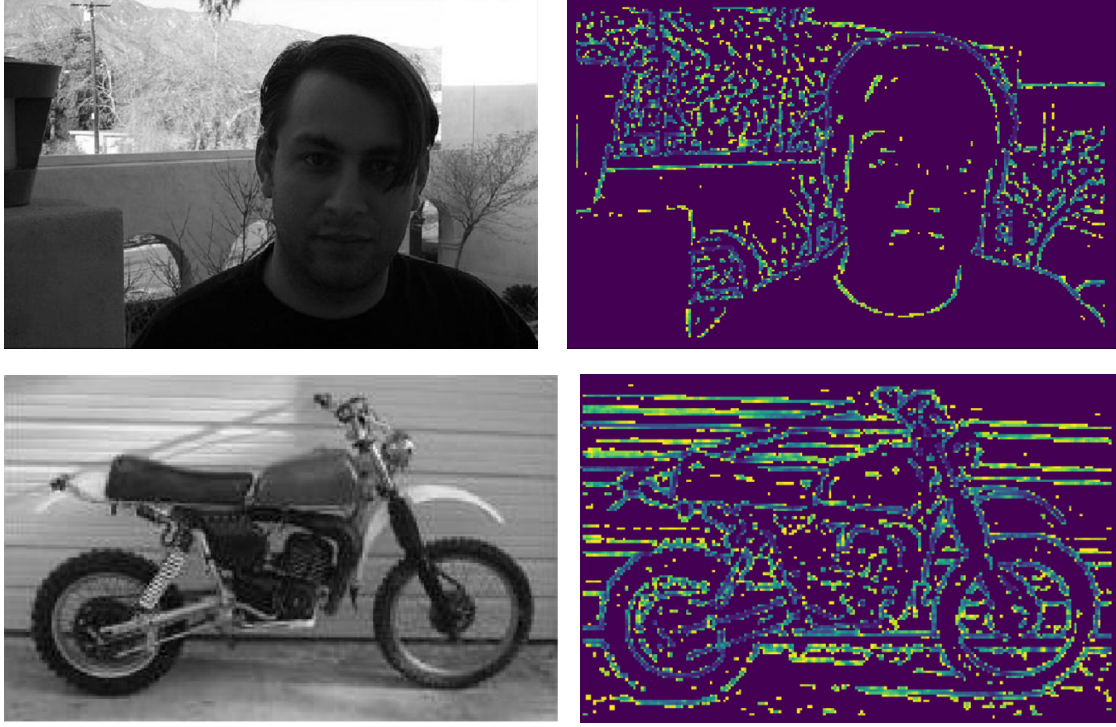
Figure 2: DoG temporal coding

1.2 Convolutional and pooling layers

A convolutional layer is represented by several neuronal maps, whose role is to detect visual features at different locations. To this extent, the synaptic weights assigned to the same neuronal map will be the same. Each neuron in the convolutional layer generates spikes according to what it receives from the pre-synaptic neurons and when its internal membrane potential reaches a certain threshold. So, at each time step t:

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

the internal potential $V_i(t)$ of the *i-th* convolutional neuron is updated by adding the spike train $S_j(t-1)$ of the *j-th* pre-synaptic neuron, multiplied to the synaptic weight $W_{j,i}$ between the *j-th* pre-synaptic neuron and *i-th* convolutional neuron. In particular, the potential $S_j(t-1)$ can be equal to 1 if the neuron has fired at previous time, or 0 otherwise. If the neuron potential $V_i(t)$ exceeds the threshold V_{th}, then it emits a spike ($S_i(t) = 1$) and it resets its potential to zero. After it fires, the neuron can not fire again and it also inhibits other neurons in its same location (but belonging to other neuronal

maps). In this way it is easier to detect a visual feature in a certain location.

In the pooling layer instead, each neuron performs a max pooling operation over a window in the corresponding neuronal map of the previous layer. Here no learning occurs and, as before, each neuron can emit only one spike. In this phase the visual information is compressed. These convolution-pooling phases are repeated for three times, to handle larger and always more complex data.

1.3 STDP-based learning

Spike-timing-dependent plasticity (STDP) is actually a biological process used by brain to modify its own synapses. This same idea has been also used in neural networks, so as to have stronger synapses in case they contribute to the firing of a post-synaptic neuron and weaker synapses if they do not give any contribute to the firing.

This process is easy to understand by looking at figure 3. In this case, four different
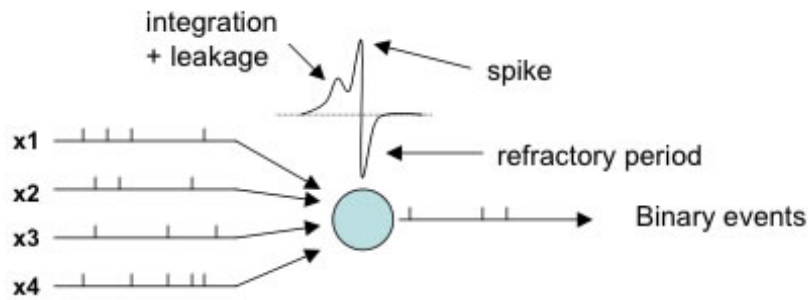

Figure 3: Contributon to the firing of a post-synaptic neuron

pre-synaptic neurons are connected to one neuron (the blue one) by synapses. Each of them is characterized by its own firing rate. These spikes are sent forward by the corresponding synapses increasing the membrane potential of the post-synaptic neuron. When this spikes, it is possible to see which pre-synaptic neuron has contributed and its membrane potential is increased. In the same way, the potential of the neurons that have not contributed to the firing of the post-synaptic one, is weakened.

As already mentioned, learning occurs only in convolutional layers, so that neurons compete to fire earlier than others and use STDP to learn input patterns. This competition is crucial to encourage neurons of different neuronal maps to learn different features. In this case, STDP is represented as:

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

where i and j refer to the post-synaptic and pre-synaptic neurons, respectively, Δw_{ij} is the synaptic weight modification and a^+ and a^- are the learning rate parameters. The term $w_{ij}(1 - w_{ij})$ is needed to bound the weight values between 0 and 1. The best choice for the learning rate parameters is with a^+ greater than a^-, but with small values for both. This, to prevent that neurons learn more than one pattern and/or that do not reach the predefined threshold to fire.

It could happen that different neurons fire at the same time, since the time is discrete. In that case, the neuron with the highest potential is selected, since it should define a higher similarity between its learned feature and input pattern.

Weights are randomly initiated with mean 0.8 and standard deviation 0.05 and the learning convergence of the *l-th* convolutional layer is represented by:

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

where n_w is the total number of synaptic weights in that layer and $w_{f,i}$ is the *i-th* synaptic weight of the *f-th* feature.

1.4 Classification

In regards of the images classification section, a dataset of labeled features is built with the output of the global max pooling layer, which in this case gives a vector of twenty elements that represent the greatest potentials reached by the neuronal maps in the third convolutional layer.

In fact the activation threshold of the neurons belonging to the last layer has been set to an infinite positive value, since has been choosen not to incurr in the reset of the potentials due to neurons spiking.

The resulting dataset then is standardised and splitted in training and testing sets in order to be fed in a linear SVM.

2 Results

3 Conclusions

References

[1] S. R. Kheradpisheh, M. Ganjtabesh, S. J. Thorpe, and T. Masquelier, "Stdp-based spiking deep convolutional neural networks for object recognition," Neural Networks, vol. 99, pp. 56–67, 2017.

[2] Spiking Neural Networks, the Next Generation of Machine Learning, https://towardsdatascience.com/spiking-neural-networks-the-next-generation-of-machine-learning-84e167f4eb2b

[3] Spiking-Neural-Network, https://github.com/Shikhargupta/Spiking-Neural-Network