Analysis of Neurons' Information in Deep Spiking Neural Networks using Information Theory

Leonardo Capozzi[1,2]
leonardo.g.capozzi@inesctec.pt
Tiago Gonçalves[1,2]
tiago.f.goncalves@inesctec.pt
Jaime S. Cardoso[1,2]
jaime.s.cardoso@inesctec.pt
Ana Rebelo[3]
ana.maria.s.rebelo@accenture.com

[1] Faculdade de Engenharia
Universidade do Porto
Porto, Portugal
[2] INESC TEC
Porto, Portugal
[3] Accenture Portugal
Lisboa, Portugal

Abstract

Deep learning methodologies have been very successful in a large number of tasks, sometimes surpassing human performance. One of the most simple neural network architectures is the multi-layer perceptron (MLP) which tries to mimic the brain in some ways. These methodologies are generally trained using gradient descent as they are differentiable. In recent years, a new methodology called spiking neural networks (SNNs) was proposed. These networks can be seen as a more biologically realistic approach than artificial neural networks (ANNs), as they use discrete spikes to transmit information, instead of continuous values. In this paper, we study these networks, and present results achieved by training these models. We also evaluate several information theory quantities such as entropy and mutual information of these networks to extract the relationship between the inputs and the outputs.

1 Introduction

In recent years deep learning methodologies have proved to be very successful in a large number of problems, namely classification in computer vision tasks, sometimes challenging human performance. One of the most simple neural network architectures is the multi-layer perceptron (MLP). These kinds of networks are trained using the backpropagation algorithm, since the loss function that we are trying to minimize is differentiable, allowing us to calculate the gradients of the weights and optimize them. Neurons in a MLP contain single and continuous activations. Biological brains, on the other hand, use discrete spikes to transmit information. With this in mind, a new methodology called spiking neural networks (SNNs) was proposed. SNNs can be seen as a more biologically realistic approach than artificial neural networks (ANNs). SNNs typically require fewer operations, are more energy-efficient, and also more hardware-efficient, making them very appealing for the future [9]. One of the current problems with these kinds of networks is the difficulty in training them since they are non-differentiable and therefore the backpropagation algorithm cannot be used. SNNs can be trained using supervised or unsupervised approaches. Currently, SNNs are still inferior in terms of accuracy compared to ANNs, but the gap is closing due to advances in recent years. The code related to this paper is available in a public GitHub repository[1].

2 Fundamental Concepts of SNNs

2.1 The Learning Process in SNNs

The spike trains are formally represented as sums of Dirac delta ($\delta(\cdot)$) functions and do not have derivatives. Therefore, gradient-based optimization rules may not be trivial to implement in SNNs. However, similarly to ANNs, the main objective is to learn the best synaptic weights. One of the paradigms in computational neuroscience relies on the concept of spike-timing-dependent plasticity (STDP). The intuition behind this approach is that the weight (synaptic efficacy) connecting a pre- and post-synaptic neuron is adjusted according to their relative spike times within an interval of milliseconds in length [1]. To understand this concept, we need to revisit Biology: if we know that the pre-synaptic neuron fires for a brief time (*e.g.*, $\approx$ 10ms) before the post-synaptic neuron, the weight connecting them is strengthened; on the other hand, if the pre-synaptic neuron fires briefly after the post-synaptic neuron, then the causal relationship between the temporal events is nonsense and the weight connecting these two neurons is weakened.

2.2 Fundamental Concepts of Information Theory

We present the fundamental concepts of information theory, needed to understand the intuition behind our analysis. All the concepts are presented as in [2]. The definition of entropy is the expected information content of random variable X. It is defined as:

$$H(X) = -\sum_{x} p_X(x) log_2[p_X(x)] \tag{1}$$

The joint entropy of two random variables X and Y is defined as:

$$H(X,Y) = -\sum_{x,y} p(x,y) log_2[p(x,y)] \tag{2}$$

Mutual information refers to the amount of information that can be obtained about one random variable by observing another random variable. The mutual information between two random variables X and Y, is related to entropy as follows:

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \tag{3}$$

3 Methodology

We performed experiments with three benchmark data sets: MNIST [7], Fashion-MNIST [10] and CIFAR-10 [6]. We follow the examples provided in [5]. This architecture consists of two layers: 1) an input layer, that contains 28×28 or 32×32 neurons (*i.e.*, one neuron per pixel); 2) a processing layer, composed of a variable, but equal, number of excitatory and inhibitory neurons [3]. Each input image is modeled as a Poisson spike-train and is fed into the excitatory neurons of the second layer. These excitatory neurons are then connected in a one-to-one fashion to inhibitory neurons (*i.e.*, each spike in an excitatory neuron will trigger a spike in its corresponding inhibitory neuron). On the other hand, each of the inhibitory neurons is connected to all excitatory ones, except for the one from which it receives a connection.

3.1 Training

This approach uses the integrate-and-fire (IF) spiking neuron model, which is given by Equation 4:

$$\frac{dv_{mem}(t)}{dt} = \sum_{i} \sum_{s \in S_i} w_i \delta(t-s) \tag{4}$$

where v_{mem} is the membrane voltage, t is the time, w_i is the weight of the i-th incoming synapse, $\delta(\cdot)$ is the Dirac delta function, and $S_i = \{t_i^0, t_i^1, ...\}$ contains the spike times of the i-th presynaptic neuron. A spike is generated if the membrane crosses a given threshold v_{thr}; the membrane voltage is then reset to a reset potential *vres*. As in [4], the continuous-time description of this model is discretised into 1ms time-steps. The synapses from input neurons to excitatory neurons are learned by STDP. The weight dynamics are computed using synaptic traces, as proposed in [8]. This approach ensures that besides the synaptic weight, each synapse keeps track of another value (*i.e.*, the presynaptic trace x_{pre}), which models the recent

[1] https://github.com/TiagoFilipeSousaGoncalves/computational-neuroscience

Table 1: Relationship between the index of the neurons that spiked on the output layer, and the index of the output neurons that have mutual information with the input of the network.

Class	MNIST		Fashion-MNIST		CIFAR-10	
	Output neurons that spiked	Neurons with mutual information	Output neurons that spiked	Neurons with mutual information	Output neurons that spiked	Neurons with mutual information
0	{3, 4, 29, 31, 52}	{3, 4, 29, 31, 52}	{22, 24, 64, 67, 85}	{22, 24, 64, 67, 85}	{1, 18, 20, 90}	{1, 18, 20, 90}
1	{8, 97}	{8, 97}	{7, 11, 16, 65, 67, 72, 87}	{7, 11, 16, 65, 67, 72, 87}	{}	{}
2	{23, 71, 85}	{23, 71, 85}	{4, 12, 29, 32, 66, 71}	{4, 12, 29, 32, 66, 71}	{18}	{18}
3	{2, 7, 14}	{2, 7, 14}	{11, 65, 67, 72, 87}	{11, 65, 67, 72, 87}	{45, 73}	{45, 73}
4	{32, 65, 80, 94}	{32, 65, 80, 94}	{}	{}	{}	{}
5	{32, 37, 67, 75, 78}	{32, 37, 67, 75, 78}	{}	{}	{}	{}
6	{17, 28}	{17, 28}	{2, 15, 19, 39}	{2, 15, 19, 39}	{11, 28}	{11, 28}
7	{18, 26, 92, 99}	{18, 26, 92, 99}	{77}	{77}	{}	{}
8	{11, 33, 35, 90}	{11, 33, 35, 90}	{3, 21, 96, 97}	{3, 21, 96, 97}	{1, 18, 23, 36}	{1, 18, 23, 36}
9	{38, 65, 94}	{38, 65, 94}	{0, 21, 92}	{0, 21, 92}	{1, 14, 18, 23}	{1, 14, 18, 23}

presynaptic spike history. Hence, every time a presynaptic spike arrives at the synapse, the trace is increased by 1 and x_{pre} decays exponentially. The weight change Δw provoked by the arrival of a postsynaptic arrival is computed according to Equation 5:

$$\Delta w = \eta (x_{pre} - x_{tar})(w_{max} - w)^{\mu} \tag{5}$$

where η is the learning-rate, w_{max} is the maximum weight, μ determines the dependence of the update on the previous weight, and x_{tar} is the target value of the presynaptic trace at the moment of a postsynaptic spike [3]. From this result, we may conclude that the higher the target value, the lower the synaptic weight will be. Also, this offset promotes that presynaptic neurons that rarely lead to the firing of the postsynaptic neuron will become more and more disconnected. This is particularly useful if the postsynaptic neuron is only rarely active. The inputs of the network are based on the MNIST, Fashion MNIST, and CIFAR-10 datasets, each containing 10 classes. After training is done we assign a class to each neuron based on the highest average response for each of the 10 classes. This is the only time that the labels are used since they are not used for the training of the synaptic weights. To predict the class of the input, we calculate the average firing rate of the neurons of each class and choose the class with the highest average firing rate.

3.2 Applying information theory to SNNs

After training the model, we generate predictions for the test set. We record samples of the inputs (*i.e.*, one image per label) and samples of the outputs (*i.e.*, the spike train generated by the output neurons for a given input image). With this data, we performed simple tests based on information theory:

1. **Computation of the entropy of the input spike train and output spike train:** we have the distribution of spikes per neuron along time. Hence, we can compute, per neuron, the probability of spiking/ not spiking for input and output neurons.

2. **Computation of the mutual information between the spike trains of the input and output neurons:** from the previous point, we obtain the marginal distributions of the input and output neurons. To compute the mutual information between the input and output neurons, we need to compute the joint distribution $P(i,o)$ which is necessary for the computation of the joint entropy $H(i,o)$. We compute the mutual information between a single output neuron and a single input neuron. If the mutual information between a pair of neurons is greater than zero, it means that those neurons have information in common. Therefore, we sum all the single contributions from the input neurons related to an output neuron, thus, obtaining a final value for that output neuron.

4 Results and Discussion

Table 1 presents the set of indices of the neurons that spiked on the output layer and the set of indices of the output neurons that have mutual information with the input of the network. The SNN obtained an accuracy of 72% on MNIST, 46% on Fashion-MNIST and 9% on CIFAR-10. An extended analysis of the results is publicly available in our extended report[2].

[2] https://github.com/TiagoFilipeSousaGoncalves/computational-neuroscience/blob/master/report.pdf

5 Conclusions and Future Work

This work presented an exploratory study with the SNN model proposed by [4] in three benchmark data sets. We computed the entropy of the input neurons, the entropy of the output neurons, and the mutual information between the input neurons and the output neurons, for a sample of a given input. Results suggest that the output neurons that present higher values of mutual information related to the input neurons are the ones that fire when given an image of a certain class, which confirms our initial intuition. Further work should be developed to the improvement of the accuracy performances and the analysis of the impact of this metric on the information theory analysis quantities.

Acknowledgements

This work is co-financed by Component 5 - Capitalization and Business Innovation, integrated in the Resilience Dimension of the Recovery and Resilience Plan within the scope of the Recovery and Resilience Mechanism (MRR) of the European Union (EU), framed in the Next Generation EU, for the period 2021 - 2026, within project HfPT, with reference 11, and by FCT – Fundação para a Ciência e a Tecnologia within the PhD grants "2020.06434.BD" and "2021.06945.BD".

References

[1] Natalia Caporale and Yang Dan. Spike timing–dependent plasticity: a hebbian learning rule. *Annu. Rev. Neurosci.*, 31:25–46, 2008.

[2] Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

[3] Peter Diehl and Matthew Cook. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in Computational Neuroscience*, 9:99, 2015. ISSN 1662-5188. doi: 10.3389/fncom.2015.00099. URL https://www.frontiersin.org/article/10.3389/fncom.2015.00099.

[4] Peter U. Diehl, Daniel Neil, Jonathan Binas, Matthew Cook, Shih-Chii Liu, and Michael Pfeiffer. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2015. doi: 10.1109/IJCNN.2015.7280696.

[5] Guy Hadash, Einat Kermany, Boaz Carmeli, Ofer Lavi, George Kour, and Alon Jacovi. Estimate and replace: A novel approach to integrating deep neural networks with existing applications. *arXiv preprint arXiv:1804.09028*, 2018.

[6] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images, 2009.

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

[8] Abigail Morrison, Ad Aertsen, and Markus Diesmann. Spike-timing-dependent plasticity in balanced random networks. *Neural computation*, 19(6):1437–1467, 2007.

[9] Amirhossein Tavanaei, Masoud Ghodrati, Saeed Reza Kheradpisheh, Timothée Masquelier, and Anthony S. Maida. Deep learning in spiking neural networks. *CoRR*, abs/1804.08150, 2018. URL http://arxiv.org/abs/1804.08150.

[10] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.