DEEP LEARNING SELF-ORGANIZING MAP OF CONVOLUTIONAL LAYERS

Christos Ferles, Yannis Papanikolaou,
Stylianos P. Savaidis and Stelios A. Mitilineos

Department of Electrical and Electronics Engineering, University of West Attica, Aegaleo, Attica, Greece

ABSTRACT

The Self-Organizing Convolutional Map (SOCOM) combines convolutional neural networks, clustering via self-organizing maps, and learning through gradient backpropagation into a novel unified unsupervised deep architecture. The proposed clustering and training procedures reflect the model's degree of integration and synergy between its constituting modules. The SOCOM prototype is in position to carry out unsupervised classification and clustering tasks based upon the distributed higher level representations that are produced by its underlying convolutional deep architecture, without necessitating target or label information at any stage of its training and inference operations. Due to its convolutional component SOCOM has the intrinsic capability to model signals consisting of one or more channels like grayscale and colored images.

KEYWORDS

Deep Learning, Unsupervised Learning, Convolutional Neural Network (CNN), Self-Organizing Map (SOM), Clustering.

1. INTRODUCTION

Probably the most common bottleneck encountered in many deep learning approaches like Convolutional Neural Networks (CNNs) is the requirement for big labeled datasets. Constructing these datasets is a costly time-consuming procedure that frequently might end up proving infeasible for various reasons. The obvious answer to this problem is devising deep learning models that can be trained with unlabeled/uncategorized data, in other words, invent unsupervised learning algorithms for such deep networks. Aligned with this ongoing research direction one can trace a number of works that combine or hybridize Self-Organizing Maps (SOMs) with CNNs.

The gamut of these approaches –including the present one– is quite widespread, spanning the range from purely unsupervised learning algorithms up to semi (or even full) supervised ones, and from shallow networks up to architectures containing multiple hidden layers; for instance [1], [2], [3] and [4]. Meeting both requirements i.e. building a deep SOM and training it in a purely unsupervised way has proven to be a complex and difficult task. Only a small number of models exist that can be classified as unsupervised beyond any doubt [5], [6]and [7]. Equally few are the approaches that extend beyond the three hidden layer limit [8], [9] and [6].

David C. Wyld et al. (Eds): ACSTY, AIBD, MLSC, CCCIOT, NATP - 2021
pp. 25-32, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110303

The Self-Organizing Convolutional Map (SOCOM) is an attempt to overcome, at a certain extent, the aforementioned limitations. Its key characteristics and contributions are: (1) A deep architecture that is in position to expand beyond the trivial, and not particularity deep, three hidden layer limit. (2) An end-to-end purely unsupervised learning algorithm that does not necessitate the targets/labels of the training samples at any stage.

The organization and structure of the remainder of this paper is as follows. Section 2 presents in detail the SOCOM both architecturally and operationally, and subsequently, analyses the key components of the corresponding feed-forward and backpropagation procedures. Section 3 contains experimental (comparative) results and performance evaluations with different algorithms. Also, (practical) application issues are discussed in this section. In section 4, a summary is given and conclusions are drawn. Finally, section 5 gives hints for future work and suggestions for potential expansions.

2. SOCOM PROTOTYPE

A generic and at the same time characteristic SOCOM architecture consisting of multiple convolutional, pooling, fully-connected and self-organizing layers is illustrated in Figure 1. The basis of the mathematically expressed algorithmic learning procedures is presented in the following subsection. This section describes the main functionality and key methods of the SOCOM from a more macroscopic operational point of view.

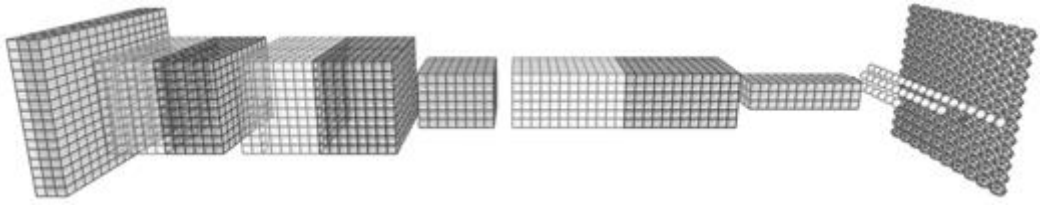

Figure 1. Detailed architecture of a SOCOM paradigm consisting of the following layers: input → convolutional→ReLU → convolutional → ReLU→pooling → convolutional → ReLU → pooling → fully-connected → fully-connected → fully-connected →output neural map.

The input layer of the SOCOM accepts any type of numerical data arranged in vectors, matrices (e.g. grayscale images) or volumes (e.g. colored images or successive images that exhibit a spatiotemporal correlation). The explicit assumption of CNNs that the inputs are images, something that makes the information propagation more efficient to implement and hugely reduces the network's parameter count, still holds in the SOCOM paradigm but does not a priori exclude all other types of input data.

As can be seen, a SOCOM comprises of a sequence of different layers with adjustable parameters. Each respective layer transforms one volume of activations to another via a differentiable function, thus facilitating the use of backpropagation during training. Stacking these layers in series eventually forms a full SOCOM architecture (Figure 1).

Similarly to other CNNs the convolutional layer consists of a set (or bank) of tunable filters/kernels. Despite of its usually small size, every filter extends through the full depth of the input volume. During the forward propagation each filter slides along the width and height of the input volume by performing convolutions. Strictly mathematically speaking the convolution operation carried out here is the same as cross-correlation except that the kernel is rotated by 180^0. In the long run this procedure yields a two-dimensional activation map that contains the responses of the respective filter at each spatial position. The hypothesis (which is currently

backed up by several experimental findings in the literature) is that the network will tune its filters so that they activate when they trace some type of visual feature, edge, or pattern. Stacking the activation maps generated by the respective layer's bank of filters produces the activation volume (or feature map) that is fed to the following (hidden) layer. As has been discussed, the units in a layer are only connected to a small region of the layer before it. This underlying weight sharing strategy, which is the aftereffect of using small filters, ends up reducing the overall number of trainable weights hence introducing sparsity, and at the same time, making the architecture suitable for manipulating images.

Neural networks' essential characteristic of nonlinearity (frequently in the form of the sigmoidal or hyperbolic tangent functions) is retained in CNNs by applying element-wise a non-saturating function to the activation volume of the preceding convolutional layer. The norm, that also the SOCOM adheres to, is to apply the rectified linear unit (ReLU) function to each individual activation produced by the convolutional layer. It has been shown, that such nonlinearities result in richer and more elaborate representations along the network architecture.

At certain points in the convolutional-ReLU layer hierarchy a pooling layer is inserted. Essentially, pooling performs a downsampling operation solely along the width and height spatial dimensions of the input volume. The reduction of such blocks of activations to just a single value has several positive aftereffects: (1) the number of parameters and related computations is reduced, (2) sparseness is introduced, and (3) overfitting is avoided.

After several convolutional and pooling layers, it is common to transition to fully-connected layers where the high-level abstract representations are formed. These densely connected layers are identical to the layers of the standard multilayer neural network. The first fully-connected layer decomposes the activations of its input volume into a one-dimensional vector and connects them to every unit it has. Subsequent layers consist of units which receive all the activations from the previous layer and perform a dot product followed by a nonlinearity. Fully-connected layers are not spatially arranged anymore something that prohibits the use of convolutional layers after a fully-connected layer.

Finally, a SOM lattice of topologically arranged neurons acts as the output layer. Each of its neurons receives the activations of every unit in the last fully-connected layer. The magnitude of each neuron's activation is based on a distance metric between the input activations and its codebook parameters. The neural mapping of the input image coincides with the position of the neuron that produces the optimal fit with respect to the computed activations and the neighborhood kernel (which has been defined over the topology of the neural grid). Apart from mapping this particular type of nonlinear projection can be further exploited for data clustering and visualization.

It is also interesting to note that the proposed SOCOM architecture is in position to incorporate any number of layers (from the previous types) in any permutation. There are only two limitations: (1) after the first fully-connected layer convolutional layers cannot be used, (2) the output layer needs to be a SOM grid.

2.1. Forward Propagation

As has been demonstrated a generic SOCOM architecture consists of an input layer, L hidden layers (convolutional, ReLU, pooling and fully-connected ones) and an output layer (viz. lattice of ordered neurons). The novel component of the SOCOM is its neural output map and in particular the different from the norm energy function that is associated with it.

The output layer that consists of G topologically arranged neurons performs a mapping of its input representations onto its neural map. More specifically, the projection of an input representation on the SOCOM plane is defined as the neuron yielding the lowest weighted squared Euclidean distance between the last hidden layer's outputs $o_i^L$and its corresponding codebook parameters $u_{g,i}$ where weighting refers to the neighborhood kernel/function $h_{e,g}$ defined over the topology of the neural grid. Frequently, this neuron (denoted as c) is referred to as "winner". Algorithmically, this best-matching winner neuron is given by:

$$c = arg\min_{e} \sum_{g=0}^{G-1} h_{e,g} \sum_{i=0}^{P-1} \left(o_i^L - u_{g,i}\right)^2 \quad \text{(eq. 1)}$$

where$P$ is the total number of units in the last hidden layer L. Additionally, this particular type of nonlinear projection can be further exploited for data clustering and visualization procedures.

2.2. Backpropagation

The purpose of being in positon to compute an error or loss function is dual. First, a definite quantification/assessment of the network's performance is obtained. Second, learning takes place via the optimization of the network's weights to minimize this specific error. This error function can be a number of different things, such as binary cross-entropy or sum of squared residuals. Differently from supervised approaches, learning in the case of SOCOM does not necessitate any type of desired or target values at any stage; thus giving rise to a pure unsupervised deep learning algorithm. The corresponding error/cost/loss function (or alternatively, the penalty term) is symbolized as E and is defined as:

$$E = \sum_{c=0}^{G-1} N(c) \sum_{d=0}^{G-1} h_{c,d} \frac{1}{2} \sum_{i=0}^{P-1} \left(o_i^L - u_{d,i}\right)^2 \quad \text{(eq. 2)}$$

Where

$$N(c) = \begin{cases} 1, & c = arg\min_{e} \sum_{d=0}^{G-1} h_{e,d} \sum_{i=0}^{P-1} \left(o_i^L - u_{d,i}\right)^2 \\ 0, & otherwise. \end{cases} \quad \text{(eq. 3)}$$

For gradient descent backpropagation the updates that need to be performed are for the weights, the biases and the deltas. The utilized energy formula by the SOCOM is in accordance with the variation proposed in [10]and has been applied in a number of hybrid SOM networks [11], [12].

3. EXPERIMENTS

The experimental investigation strategy that has been followed serves a dual purpose. First, a (mainly quantitative) comparison against a comprehensive series of similar/related deep SOMs is achieved. These models cover the full range of SOMs that extend beyond the mainstream two layer architecture (a single input layer connected to an output neural map) by employing at least one intermediate hidden layer between their inputs and outputs. Second, the conducted experiments act as a proof of concept for the proposed network by tangibly demonstrating/verifying its capabilities and clustering performance, given the modelling problem under consideration.

Following the justified requirement of comparing the SOCOM approach with an as wide as possible gamut of likewise SOM approaches the MNIST benchmark choice was unavoidable since: (1) the landslide of published deep SOMs report results (frequently, exclusively only) on the MNIST dataset, (2) it is traditionally the entry point dataset of experimental investigation when it comes to testing deep learning algorithms. The MNIST benchmark used in the current experimental setup is Yann LeCun's version [13] which contains handwritten numerical digits that have been size-normalized and centered in a fixed-size image. The dataset consists of 60000 training examples and 10000 testing examples; it is an almost balanced collection where the highest deviating category (in terms of sample size) is the handwritten digit "1" (approximately 11.2% in the train set and 11.4% in the test set instead of the expected 10%).

Before moving further an important point should be made. An end-to-end purely unsupervised learning algorithm that does not necessitate the targets/labels of the training samples at any stage. If these are provided they can potentially be used, but, typically, an unsupervised model should be is in position to function even when these are absent or missing. Nevertheless, there is merely a handful of approaches that adhere to strict unsupervised training criteria [5], [6], [7]and the ones reporting results on the MNIST database are specifically indicated in Table 1. Frequently, in the literature, an "unsupervised" model with a supervised or self/semi-supervised training procedure is proposed. Apart from the fact that this defeats the purpose and it is deluding, it is practically of questionable use. If the targets/labels of the input data are utilized during training then why resort in clustering results (which are intrinsically of coarser/qualitative nature) when the alternative of classification results (which are more detailed/informative) is on the table.

Evaluating the quality of a clustering output and, in particular in the case of SOMs, of a mapping output is a non-trivial task that has been tackled by introducing various internal and external criteria. Internal criteria are more qualitative in the sense that they evaluate clustering results indirectly (e.g. by means of organization, compactness/sparseness, isolation and preservation), whereas external are more quantitative since by measuring the match between clustering and external (e.g. human-based) categorizations they are in position to provide more precise assessments. In the related literature, the most widely used external criterion, in particular for clustering tasks, is purity:

$$PUR = \frac{1}{S}\sum_{p=1}^{P}\max_{1\le t\le T}\left|s_p \cap s_t\right|. \qquad \text{(eq. 4)}$$

The subscript p denotes the partitioning of a set of S samples into P distinct clusters (a posteriori estimated by the model); similarly, the subscript t denotes the assignment of these samples into T categories (a priori defined in the dataset). As expected its resulting values lie in the $[0, 1]$ interval. Obviously purity identifies with accuracy given that the majority voting principle is utilized for labeling each individual cluster. Although purity intuitively is rather straightforward/precise it tends to favor small (in sample numbers) clusters like singletons.

On a related note, a distinction should be drawn between obtaining accuracies with a posterior labeling of neurons (based on data labels) and obtaining accuracies with the addition of a supervised model/layer (like MLP, SVM or fully-connected Softmax network). Obviously, the latter approaches' results are misleading since the unsupervised networks' outputs are treated as input features to a supervised network (which is obviously trained in a supervised manner). This type of experimental testing does reveal characteristics of the unsupervised module's output feature space but is by no means indicative of the network's clustering capabilities and performance.

The SOCOM architecture that has been utilized in the present series of experiments closely follows that of resnet18 [14] upon which appropriate modifications have been carried out. Specifically, the first hidden 2D convolutional layer has been replaced by a 2D convolutional layer that accepts single channel signals/images, the last fully-connected layer has been removed, an output layer implementing the neural map has been added, followed by an 1D pooling layer for facilitating the backpropagation optimization algorithm. Standard stochastic gradient descent backpropagation with momentum [15] is used for training the network. Transfer learning [16]is also utilized for obtaining the initial weight/parameter values of the hidden layers that are shared with the resnet18 architecture. The codebook parameters have been initialized according to the methodology described in [17], using a uniform distribution. The lower and upper limits of the value ranges used for the learning rate and momentum hyper-parameters have been estimated according to the technique described in [18]. Output neurons are arranged onto a 2D hexagonal grid; the Gaussian neighborhood kernels' standard deviations start with a value equal to half the largest dimension on the grid and decrease linearly to one map unit, during training. The performance (in terms of accuracy) and main characteristics of a list of indicative deep SOMs including SOCOM are summarized in Table 1.

Table 1. The architectural/algorithmic characteristics of various deep SOMs and their respective accuracies on the MNIST dataset.

Model/Network	Accuracy (%)	End-to-End Unsupervised Learning	Number of Layers
(Aly, 2020)[4]	99.43		3
(Braga, 2020)[19]	98.36		4
SOCOM	97.35	●	20
(Wang, 2017)[9]	96.7		8
(Liu, 2015)[2]	96.17		3
(Friedlander, 2018)[5]	87.7	●	3
(Wickramasinghe, 2019)[3]	87.12		2
(Wickramasinghe, 2018)[20]	84.87		2

As can be seen the proposed SOCOM outperforms the majority of previous approaches by utilizing a purely unsupervised learning algorithm which is capable of handling (through the backpropagation of gradients) all the necessary computations needed for adjusting the underlying deep architecture. All the rest of the approaches, apart from [5], use extensively label/target information throughout their training procedures for reaching the reported accuracy rates. This observation further demonstrates the capabilities of the SOCOM since it is in position to perform better against (or almost at par with) models that access richer information like the label/class information of input images of handwritten digits. It should be noted that by taking into consideration the other purely unsupervised deep SOM the SOCOM achieves nearly 10% improved accuracy. Last, it is also important to reiterate that algorithmically the SOCOM model is not restricted to single channel input signals (like the grayscale MNIST images) but it is capable of incorporating three channel inputs (i.e. colored images) or input volumes of higher dimensions. This can be accomplished in a straightforward way by not replacing the first hidden convolutional layer's filter shape with the downscaled one used in these experiments.

4. SUMMARY

One of the central dogmas in the field of machine learning (which differently from dogmas in other domains, is continuously being backup up experimentally) is that the stratification of

several levels of nonlinearity is the key to tackle complex recognition tasks, infer higher-level correlations between variables and representations of data, and, in general, mimic and model the way human perception and ingenuity function. SOCOM aligns with the ongoing research towards combining nonlinearities of neurons into networks for modelling highly complex and increasingly varying functions. It is doing this by trying to remain loyal to the unsupervised learning guidelines of necessitating as less label information as possible.

It has been shown, both algorithmically (i.e. in theory) and experimentally (i.e. in practice), that this first working SOCOM prototype is in position to incorporate a deep architecture (evidently deeper in comparison to the deep SOMs reported in the literature) which is trained with a gradient backpropagation algorithm tailored to meet the requirements of the architecture's complexity, depth and parameter size. As has been discussed previously, the proposed algorithm not only is along the lines of the optimization methods which are proven to work with deep networks but also keeps the required label/target information to a minimum. Further, due to the fact that the first hidden layers of SOCOM's architecture are convolutional, the data that can be modelled are not restricted to grayscale images (i.e. single channel ones) but instead can consist of an arbitrary number of channels e.g. colored images (i.e. three channels) or even sequences of images/signals; such data rarely can be processed by the currently published deep SOMs.

5. FUTURE WORK

It is reasonable that this proof-of-concept study of the SOCOM prototype could give rise to a number of closely-related research directions pointing towards expanding and enriching the model, and towards making full use of its clustering capabilities in real-world complex problems. More specifically, an omnidirectional research plan could involve: (1) The construction of deeper SOCOMs based for instance on the resnet34, resnet50 and resnet152 architectures [14]. (2) Gradually utilizing the backpropagation flow of gradients in adjusting/tuning layers further deep down the architecture. (3) Incorporating diverse deep network configurations that are based upon other well-known paradigms like Alexnet[21], VGG [22], and GoogLeNet[23]. (4) In depth and in detail analysis and evaluation of the various optimization methods provided by the Pytorch framework. (5) Using existing deep learning visualization techniques up to the last hidden representation layer, and, subsequently, treating visualizations as "inputs" to the ordered neuron output array. The final objective in this case is having either a visualization of what the map models/clusters [24] or a projection of the achieved higher-level representations onto the output map.

ACKNOWLEDGEMENTS

This research is co-financed by Greece and the European Union (European Social Fund-ESF) through the Operational Programme "Human Resources Development, Education and Lifelong Learning 2014-2020" in the context of the project "Self-Organizing Convolutional Maps" (MIS 5050185).

The authors would like to thank the anonymous reviewers for their constructive comments and insightful remarks.

REFERENCES

[1] Lawrence, S., Giles, C. L., Tsoi, A. C., & Back, A. D. (1997). Face recognition: A convolutional neural-network approach. IEEE transactions on neural networks, 8(1), 98-113.

[2] Liu, N., Wang, J., & Gong, Y. (2015, July). Deep self-organizing map for visual classification. In 2015 international joint conference on neural networks (IJCNN) (pp. 1-6). IEEE.

[3] Wickramasinghe, C. S., Amarasinghe, K., & Manic, M. (2019). Deep self-organizing maps for unsupervised image classification. IEEE Transactions on Industrial Informatics, 15(11), 5837-5845.
[4] Aly, S., &Almotairi, S. (2020). Deep Convolutional Self-Organizing Map Network for Robust Handwritten Digit Recognition. IEEE Access, 8, 107035-107045.
[5] Friedlander, D. (2018). Pattern Analysis with Layered Self-Organizing Maps. arXiv preprint arXiv:1803.08996.
[6] Pesteie, M., Abolmaesumi, P., &Rohling, R. (2018). Deep neural maps. arXiv preprint arXiv:1810.07291. maps.
[7] Stuhr, B., &Brauer, J. (2019, December). CSNNs: Unsupervised, Backpropagation-free Convolutional Neural Networks for Representation Learning. In 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA) (pp. 1613-1620). IEEE.
[8] Part, J. L., & Lemon, O. (2016, October). Incremental on-line learning of object classes using a combination of self-organizing incremental neural networks and deep convolutional neural networks. In Workshop on Bio-inspired Social Robot Learning in Home Scenarios (IROS), Daejeon, Korea.
[9] Wang, M., Zhou, W., Tian, Q., Pu, J., & Li, H. (2017, October). Deep supervised quantization by self-organizing map. In Proceedings of the 25th ACM international conference on Multimedia (pp. 1707-1715).
[10] Heskes, T. (1999). Energy functions for self-organizing maps. In Kohonen maps (pp. 303-315). Elsevier Science BV.
[11] Ferles, C., Papanikolaou, Y., & Naidoo, K. J. (2018). Denoisingautoencoder self-organizing map (DASOM). Neural Networks, 105, 112-131.
[12] Ferles, C., &Stafylopatis, A. (2013). Self-organizing hidden markov model map (SOHMMM). Neural networks, 48, 133-147.
[13] LeCun, Y., Bottou, L., Bengio, Y., &Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.
[14] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
[15] Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (2013, May). On the importance of initialization and momentum in deep learning. In International conference on machine learning (pp. 1139-1147). PMLR.
[16] Sharif Razavian, A., Azizpour, H., Sullivan, J., &Carlsson, S. (2014). CNN features off-the-shelf: an astounding baseline for recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops (pp. 806-813).
[17] Glorot, X., &Bengio, Y. (2010, March). Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth international conference on artificial intelligence and statistics (pp. 249-256). JMLR Workshop and Conference Proceedings.
[18] Smith, L. N. (2017, March). Cyclical learning rates for training neural networks. In 2017 IEEE winter conference on applications of computer vision (WACV) (pp. 464-472). IEEE.
[19] Braga, P. H., Medeiros, H. R., &Bassani, H. F. (2020, July). Deep Categorization with Semi-Supervised Self-Organizing Maps. In 2020 International Joint Conference on Neural Networks (IJCNN) (pp. 1-7). IEEE.
[20] Wickramasinghe, C. S., Amarasinghe, K., Marino, D., & Manic, M. (2018, July). Deep self-organizing maps for visual data mining. In 2018 11th International Conference on Human System Interaction (HSI) (pp. 304-310).
[21] Krizhevsky, A. (2014). One weird trick for parallelizing convolutional neural networks. arXiv preprint arXiv:1404.5997.
[22] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
[23] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).
[24] Ferles, C., Beaufort, W. S., &Ferle, V. (2017). Self-Organizing Hidden Markov Model Map (SOHMMM): biological sequence clustering and cluster visualization. In Hidden Markov Models (pp. 83-101). Humana Press, New York, NY.