Improving Image Classification Accuracy using Hybrid Systems of Support Vector Machines and Convolutional Neural Networks

David Macêdo, *Member, IEEE*

***Abstract*—Both Support Vector Machines (SVM) and Convolutional Neural Networks (CNN) techniques have been used to construct images classifiers, but both methods have some weakness. In this paper, we propose a hybrid system that composes the mentioned techniques. We show that the proposed hybrid system presents better performance than the isolated stand-alone systems.**

***Index Terms*—Support Vector Machines, Hybrid Systems, Convolutional Neural Networks, Deep Learning, Computer Vision, Pattern Recognition, Image Classification, MNIST.**

I. Introduction

THE last two decades has brought many advances in machine learning research. Of particular interest, SVM classifiers have been shown to be very successful in many recognition tasks. The mentioned approach have some very attractive characteristic such as a convex loss function and a maximum margin criterion, which provides good generalization capabilities[1].

Despite all the very attractive qualities, SVM presents some drawbacks when treating image data in classification tasks. First of all, SVM is a shallow architecture and therefore it presents some troubles in learning deeper hierarchical features commonly presented by image datasets. Moreover, the SVM is unable to explore the prior knowledge presented by the two-dimensional geometry of image data.

Yet another problem may arise when dealing with SVM to image recognition. Since an image is a set of a large number of raw pixels, the large quantity of the attributes and number of image samples may implicate in scalability problems.

On the other hand, Convolutional Neural Networks (CNN) are a special kind of neural networks designed specially to explore the geometry of images in order to accommodate this prior knowledge to the model[2]. CNNs are built using a hierarchical architecture in order to be able to learn deeper features presented by the image data set and scalability is one of the main advantages presented by these models. Nevertheless, convolutional neural networks have some drawbacks. Since the loss surface is non-convex, a global minimal is not assured in this case. Moreover, the last layers of CNNs are traditional fully-connected linear layers, which is a classifier less efficient than an SVM (the softmax layer do not chance in essence the generalization power of the previews fully-connected linear layers).

In this paper, we investigate and propose a hybrid system in order to overcome the drawbacks of the techniques presented while combining the best characteristic of each system. The main idea is beginning with a convolutional neural network trained using backpropagation in order to extract features. After this, the feature extracted are used in a second step in order to train an SVM classifier. We show that the hybrid system proposed presents a better accuracy than both SVM or CNN used separately. In order to demonstrate the accuracy of our hybrid model, we will be working with the very known MNIST dataset[2].

We have used TensorFlow framework[3] in order to construct the CNN to extract high-level features from the images and Python Scikit-learn library[4] to process the SVM classifications and related activities. The source code for this project is freely available at https://github.com/dlmacedo/SVM-CNN.

II. Proposed Hybrid System

A. Convolutional Neural Networks

Convolutional neural networks are feed forward neural networks specialized mainly in treating image data. Taking into consideration that images are symmetrical by a shift in position, weight sharing and selective fields techniques are used in order to create filter banks that extract geometrically related features from the image data set. The process is composed hierarchically over many layers in order to obtain higher level features after each layer. Fully connected layers are added at the top of the architecture in order to function in a similar way of perceptron classifiers. The network is normally trained using gradient backpropagation techniques.

The filter banks used in CNNs have an attribute kernel size, which is related to the number of weights used to extract the features of a patch of the image. In the proposed hybrid model, we are using 5x5 kernel sizes. The first convolutional layer was defined to have 32 kernels and the second one to have 64 filters. The third layer has 128 nodes in a fully connected architecture. The fourth layer is composed of 10 nodes also fully connected. The last layer is a softmax operation.

B. SVM and Kernels

The Support Vector Machine[5] algorithm is used as the final classifier in the proposed system. This is a very successful and mathematical elegant method commonly used by machine learning community. It is shown to provide an optimization problem that presents a convex error surface. It is normally

Manuscript received May 30, 2016; revised June 15, 2016.

TABLE I
TEST PERCENT ACCURACY COMPARISON

Experiment	CNN	SVM_{CNN}
1	97.52	98.55
2	97.59	98.66
3	97.58	98.59
4	97.25	98.51
5	97.29	98.49
6	97.44	98.62
7	97.03	98.44
8	97.49	98.62
9	97.26	98.48
10	97.16	98.32

used in conjunction with kernel methods in order to enable the classification of non-linearly separate data set.

C. *Hybridization*

The first step in order to build the hybrid model is to train the CNN using backpropagation algorithm. The ADAM method[6] was used in order to optimize the model that makes use of a cross-entropy loss function.

After training the CNN, in order to construct the proposed hybrid model, the softmax layer and the last 10 nodes layer is dropped and the activations of the 128 nodes fully connected previous layer are used as the features extracted from a given image from the MNIST data set. Therefore, instead of feeding the SVM classifier with the row 784 pixels values, the proposed hybrid system, labeled as SVM_{CNN}, is built by feeding an SVM classifier with the 128 higher level features extracted from the previews trained CNN.

After the previous procedure, the regular maximum margin criterion is used in order to optimize the resulting model. Therefore, the hybrid system has a convex error surface that uses high-level features extracted from the convent as input data.

III. RESULTS AND DISCUSSIONS

The MNIST is a dataset of handwritten digits. In our experiments, we used 55 thousand images for training and 10 thousand for test. Each example consists of a grayscale 28x28 pixels image. In the following performance analysis, we define SVM_{L} as the experiment that consists of classifying the SVM using linear kernel directly into a flat array of 784 features representing each pixel of the MNIST images. The experiment that consists of training and testing SVM in the flat array of 784 features representing an MNIST image using gaussian kernels is called SVM_{G}. In our experiments, the test accuracy of SVM_{L} and SVM_{G} were 93.93% and 94.39%, respectively.

In this article, the original convolutional neural network used to extract the feature is called CNN and our proposed hybrid system that consists of using an SVM classifier on the 128 high-level features extracted by the conventional neural network is called SVM_{CNN}. We executed each of the mentioned models one hundred times. The Table I shows the test accuracy of the first ten experiments realizations of CNN and SVM_{CNN}. We observe that both models are consistently more accurate than both SVM_{L} and SVM_{G}. More important, we see that SVM_{CNN} consistently improves the accuracy of the CNN. Therefore, our proposed hybrid system is shown to be a better classifier than both the original individual systems.

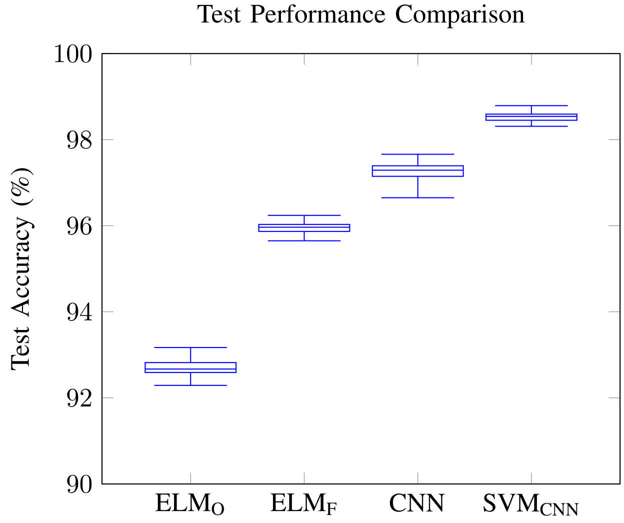

Fig. 1. Box plot of the distribution of test percent accuracy of one hundred of experiments of each model on MNIST data set.

In order to evaluate how accurate and fast is the proposed hybrid system when compared to other class of classifiers, we trained and test two Extreme Learning Machines (ELM) models. The first one with one thousand hidden nodes called ELM_{O} and the second one with four thousand nodes called ELM_{F}. For this purpose, we also executed each of the ELM one hundred times.

The Fig. 1 summarizes in box plots the statistics of the distribution of the test accuracy after the execution of one hundred experiments of the ELM_{O}, ELM_{F}, CNN, SVM_{CNN} models. In the mentioned figure, we can see that the ELM_{O} presents very poor performance. In fact, its performance is even worse that the both SVM_{L} and SVM_{G} models. We also observe that ELM_{F}, which used four thousand hidden layer nodes, achieves performance better that the original both SVM classifiers, but it is still poor when compared with both CNN, SVM_{CNN} models. We believe that this poor performance is mainly because the fact this is a shallow architecture.

Moreover, as we will see below, in order to achieve this accuracy near the CNN model, a number of hidden layers nodes make this ELM model presents training time much higher than the competitor CNN model. As expected, the CNN model presents a good performance, but the proposed SVM_{CNN} model is still better. Again, the experiments suggest that learn high-level feature indeed improves the accuracy of a posterior classifier. The results indicate that the random hidden layer features of the ELM models make it difficult for them to achieve good performance.

The Fig. 2 presents the mean training time of the models used in the article. We can observe that the SVM model used to directly classify the flat 784 features of the MNIST images are really slow, especially the one that uses Gaussian kernel.

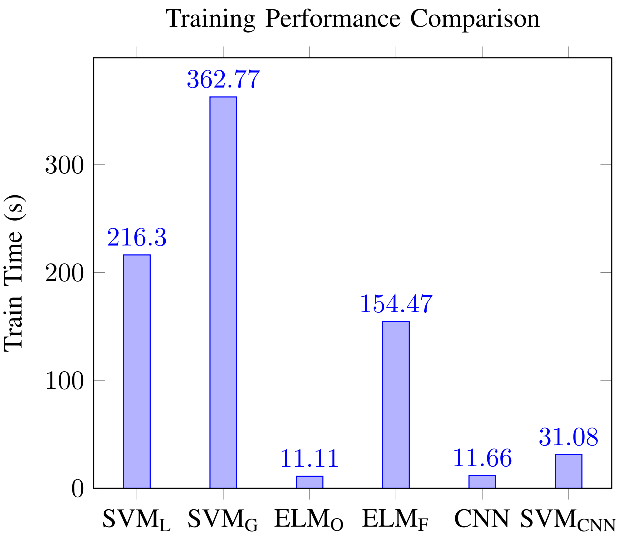

Fig. 2. Mean training time in seconds of one hundred of experiments of each model, with exception of SVM$_{\text{L}}$ and SVM$_{\text{G}}$, that were executed just once since they have an almost deterministic training time.

The figure also shows that ELM$_{\text{O}}$ model is indeed very fast, as expected. However, we recall that this model presented very poor accuracy. Indeed, its accuracy was even worse than the original SVMs models. Naturally, in order to improve the poor accuracy presented by the ELM$_{\text{O}}$ model, we trained the ELM$_{\text{F}}$. We recall that it indeed improved the performance, that now is at least better than the SVMs ones. However, despite it still presents worse test performance than both CNN and SVM$_{\text{CNN}}$, its training time skyrocket. In fact, while we multiplied the number of hidden nodes by four, the training time improved almost fifteen times, showing how not scalable is this kind of models.

The analysis of the Fig. 2 also shows how fast the CNN model is. It can be a little surprise since it is often believed that deep models are slow to train. However, we have to emphasize those deep architectures, like all other neural networks models, are extremely parallelable, with dramatically reduces the training time when graphics processing unit (GPU) or any other parallel computing hardware are used. We have also to mention that CNN has an architecture that makes these techniques have orders of magnitude fewer weights than the traditional multilayer perceptron, which also contribute to reducing the training time. This characteristic is one of the most singular features of deep learning models since others models like SVM and ELM are extremely less parallelable.

Therefore, despite to be as fast as the ELM$_{\text{O}}$, the CNN is much more accurate (Fig. 1). Moreover, while the ELM$_{\text{F}}$ presents test performance not much worse than CNN, its training time is already much higher. Our proposed hybrid model SVM$_{\text{CNN}}$ has trained five times better than the one presented by ELM$_{\text{F}}$ while providing much better test performance. It should be noticed that SVM$_{\text{CNN}}$ training time much better than the original SVMs models, while improved drastically the test accuracy. In other words, training an SVM on a reduced number of high-level features is a good approach.

IV. Conclusion

We have shown that a hybrid system composed of an SVM classifier trained on high-level features extracted from a CNN not only improves consistently improves significantly the test accuracy when compared with both original systems but also reduces the training time when compared with the original SVM classifier.

It was also showed that, while presenting much better test accuracy, the hybrid system is still competitive in training times with ELM models. We observed that the parallel computing capabilities of deep architectures make this model very competitive in training time when using the full power of parallel hardware like the recent and relatively cheap GPUs.

References

[1] V. N. Vapnik, *Statistical learning theory*. Wiley, 1998.

[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.

[3] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, and G. Research, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems.”

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and . Duchesnay, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[5] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, 9 1995. [Online]. Available: http://link.springer.com/10.1007/BF00994018

[6] D. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” 12 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

David Macêdo received the bachelor electronic engineer degree *summa cum laude* from the Federal University of Pernambuco, Brazil, where he is currently pursuing the M.S. degree with the Center of Informatics. He is a Senior Consultant at Recife Center for Advanced Studies and Systems (C.E.S.A.R.). His current research interests include Deep Learning, Convolutional Neural Networks, Recurrent Neural Networks and applications in Pattern Recognition and Computer Vision.