FEMa: A Finite Element Machine for Fast Learning

Danillo Roberto Pereira, João Paulo Papa, *Member, IEEE,* and Hojjat Adeli, *Member, IEEE*

***Abstract*—**

***Index Terms*—Finite Element Method, Classification, Regression.**

I. Introduction

THE "Big Data" era has flooded researchers and the whole community with tons of data daily. Multimedia-based applications are in charge of generating an unsurmountable amount of data, which end up at the screens of mobile phones and tablets. Home-made videos are usually referred as the bottleneck of any network traffic analyzer, since they are uploaded to cloud-driven servers as soon as they are generated or forwarded by someone else via the so-called social networks.

The huge amount of data requires to be processed and mined efficiently either. Former versions of well-known machine learning techniques such as Support Vector Machines (SVMs) [1] and Artificial Neural Networks (ANNs) [2] are now being implemented in General-Purpose Computing on Graphics Processing Units (GPGPU) to cope with streams of data that need to be analyzed daily.

Active learning is another research area that needs fast techniques for learning and classification. One very usual example concerns interactive and semi-supervised learning tools for image classification and annotation. Suppose a physician wants to classify a Magnetic Resonance image of the brain, which may contain hundreds of thousands of pixels. The user shall mark a few positive and negative samples (pixels) that will be used to train the classifier, which then classifies the remaining image. Further, the user shall refine the results by marking some misclassified regions for training once more. Notice the whole process should take a few seconds/iterations. In this context, the user feedback is crucial to obtain a concise/reliable labeled image.

Considering the aforementioned situation, traditional machine learning techniques may not be appropriate to be employed, since they can hardly handle the problem of updating the model learned previously when new training samples come to the problem. Support Vector Machines are known to be costly, since they require a fine-tuning parameter step, which turns out to be the bottleneck for efficient implementations. Although different variations and GPU-based implementations are published monthly, there is always a trick to use them, which makes them far from being user-friendly. Additionally, SVM training step is quadratic with respect to the number of training examples.

Deep learning techniques have arisen as the hallmark in the last years [3], since they can learn features from images/signals without label information. Although such approaches have obtained outstanding results in a number of applications, they usually overfit under small training sets. Also, some architectures require hundreds of parameters to fine-tune, being quite costly for training either. Recently, Extreme Learning Machines (ELMs) [4], [5] have been in the spotlight due to their simple architecture, which leads to a fast learning phase, and with promising results in several applications. However, ELMs may require a higher number of hidden neurons due to the random initialization of the weights between the input and the hidden layer. Additionally, ELMs can get trapped from local optima [6].

Graph-based pattern recognition techniques took their place in the scientific community as well. Some years ago, Papa et al. [7], [8], [9] proposed the Optimum-Path Forest (OPF), which is a framework to the design of classifiers based on the optimum-path forest. OPF has obtained promising results in a number of applications, being much faster than SVM for training, since its former version is parameterless [8], [9] and does not require fine-tuning parameters. However, OPF-based classifiers are usually affected under high-dimensional spaces, which are always a shortcoming for distance-based classifiers.

Artificial Neural Networks have been reinvented in the last decades. From the original Backpropagation learning algorithm [10] to faster approaches such as the Levenberg-Marquardt [11], the reader can refer to a number of variants that somehow try to deal with the problem of avoiding getting trapped from local optima during training, as well as to make their convergence step faster [12].Mais algumas referencias aqui. Depois falar de PNNs e EPNNs.

II. Finite Element Machine Classifier

A. Complexity Analysis

III. Conclusion

The conclusion goes here.

Acknowledgment

The authors are grateful to FAPESP grants #2014/16250-9 and #2015/50319-9, as well as CNPq grant #306166/2014-3.

D. Pareira and J. Papa are with the Department of Computing, São Paulo State University, Bauru, SP, 17033-360 Brazil e-mail: dpereira@ic.unicamp.br, papa@fc.unesp.br.

H. Adeli is with the Department of Civil, Environmental and Geodetic Engineering, The Ohio State University, Columbus, OH 43210 USA e-mail: adeli.1@osu.edu

Manuscript received April 19, 2005; revised August 26, 2015.

References

[1] C. Cortes and V. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[2] S. Haykin, *Neural Networks: A Comprehensive Foundation (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2007.

[3] Y. LeCun, Y.Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[4] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1759–1763, pp. 489–501, 2006, neural NetworksSelected Papers from the 7th Brazilian Symposium on Neural Networks.

[5] G. B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 513–529, 2012.

[6] Q.-Y. Zhu, A. K. Qin, P. N. Suganthan, and G.-B. Huang, "Evolutionary extreme learning machine," *Pattern Recognition*, vol. 38, no. 10, pp. 1759–1763, 2005.

[7] J. P. Papa and A. X. Falcão, "A new variant of the optimum-path forest classifier," in *Advances in Visual Computing*, ser. Lecture Notes in Computer Science, G. Bebis, R. Boyle, B. Parvin, D. Koracin, P. Remagnino, F. Porikli, J. Peters, J. Klosowski, L. Arns, Y. Chun, T.-M. Rhyne, and L. Monroe, Eds. Springer Berlin Heidelberg, 2008, vol. 5358, pp. 935–944.

[8] J. P. Papa, A. X. Falcão, and C. T. N. Suzuki, "Supervised pattern classification based on optimum-path forest," *International Journal of Imaging Systems and Technology*, vol. 19, no. 2, pp. 120–131, 2009.

[9] J. P. Papa, A. X. Falcão, V. H. C. Albuquerque, and J. M. R. S. Tavares, "Efficient supervised optimum-path forest classification for large datasets," *Pattern Recognition*, vol. 45, no. 1, pp. 512–520, 2012.

[10] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

[11] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the marquardt algorithm," *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, 1994.

[12] Q. Liu and J. Wang, "A one-layer recurrent neural network for constrained nonsmooth optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 5, pp. 1323–1333, 2011.