An Efficient FPGA-based Architecture for Contractive Autoencoders

Madis Kerner*, Kalle Tammemäe*, Jaan Raik*, Thomas Hollstein*†
*Tallinn University of Technology, Tallinn, Estonia
†Frankfurt University of Applied Sciences, Frankfurt, Germany
Email: madis.kerner@taltech.ee, kalle.tammemae@taltech.ee, jaan.raik@taltech.ee, hollstein@fb2.fra-uas.de

***Abstract*—Deep learning neural networks have gained much attention in recent research. Excellent results in various domains have proved the usefulness of such algorithms. However, training a deep learning network requires substantial computational effort; therefore, resource-constrained systems like edge devices in the IoT domain still lack full implementations, and training of the network is offloaded to the cloud. Online or unsupervised training of the network, on the other hand, is often a must if the system has to adjust to possible drift of the environment parameters or there is not enough data available initially. This paper proposes the first Xilinx Zynq FPGA (Field Programmable Gate Array) based implementation of the contractive autoencoder (CAE), including training of the network.**

I. Introduction

Deep learning (DL) algorithms have been proved to be useful in various domains: image recognition, natural language translation, human activity recognition, and anomaly detection [1], [2], [3]. However, the current state-of-the-art solutions rely on graphical processing units and other general-purpose hardware accelerators.

The DL algorithms extract the essential features of the input signal automatically; this enables automatic learning and increases the DL modeling capabilities [4].

Before the deployment, DL algorithms need training, which requires substantial computational power. Therefore, the network is either trained offline, or using the cloud [5].

The broader focus of this work is related to the unsupervised DL algorithms and implementations on resource-constrained systems. One class of this kind of methods are autoencoders, which reproduce the input signal to its output. The middle layer of an autoencoder contains compressed features [6], which can be used for different purposes, like data-compression [7].

[8] describes the framework for FPGA based forward pass execution of various DL networks but does not include the training, which has to be carried out separately.

Considering autoencoders, [9] provides the study of an FPGA based sparse stacked autoencoder, but again, it does lack the training.

Using high-level synthesis is another approach found in the literature; [10] provides the solution to train stacked autoencoders. However, the proposed solution lacks the training speed and the contraction term.

The main contribution of this work is to provide the first hardware-based implementation of the Contractive Autoencoder (CAE) [11]. Also, this paper follows proposals to use shared weights on the input and output layers [12] and fixed-point representations for weights and biases [13].

The proposed architecture uses node-level parallelism. For back-propagation, additional parallelism was achieved by maximally reusing the computational results.

The functionality of the solution was verified using the downscaled MNIST dataset [14]. The $38\mu s$ total execution time for a forward pass and training yields to a maximum of 26KS input rate.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 5 2015.
[2] J. Schmidhuber, "Deep Learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.
[3] T. Plotz and Y. Guan, "Deep Learning for Human Activity Recognition in Mobile Computing," *Computer*, vol. 51, no. 5, pp. 50–59, 2018.
[4] H. F. Nweke, Y. W. Teh, M. A. Al-garadi, and U. R. Alo, "Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges," *Expert Systems with Applications*, vol. 105, pp. 233–261, 9 2018.
[5] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A Survey," *Pattern Recognition Letters*, vol. 119, pp. 3–11, 2 2018.
[6] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science (New York, N.Y.)*, vol. 313, no. July, pp. 504–507, 2006.
[7] O. Yildirim, R. S. Tan, and U. R. Acharya, "An efficient compression of ECG signals using deep convolutional autoencoders," *Cognitive Systems Research*, vol. 52, pp. 198–211, 2018.
[8] L. D. Medus, T. Iakymchuk, J. V. Frances-Villora, M. Bataller-Mompean, and A. Rosado-Munoz, "A Novel Systolic Parallel Hardware Architecture for the FPGA Acceleration of Feedforward Neural Networks," *IEEE Access*, vol. 7, pp. 76 084–76 103, 2019.
[9] M. G. Coutinho, M. F. Torquato, and M. A. Fernandes, "Deep neural network hardware implementation based on stacked sparse autoencoder," *IEEE Access*, vol. 7, pp. 40 674–40 694, 2019.
[10] J. Maria, J. Amaro, G. Falcao, and L. A. Alexandre, "Stacked Autoencoders Using Low-Power Accelerated Architectures for Object Recognition in Autonomous Systems," *Neural Processing Letters*, vol. 43, no. 05, pp. 445–458, 2016.
[11] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: explicit invariance during feature extraction," in *Proceedings of The 28th International Conference on Machine Learning (ICML-11)*, no. 1, 2011, pp. 833–840.
[12] A. Suzuki, T. Morie, and H. Tamukoh, "FPGA implementation of autoencoders having shared synapse architecture," in *PLoS One*, vol. 13, no. 03, 2018, pp. 1–22.
[13] J. Jiang, R. Hu, D. Wang, J. Xu, and Y. Dou, "Performance of the fixed-point autoencoder," *Tehnicki vjesnik - Technical Gazette*, vol. 23, no. 02, pp. 77–82, 2016.
[14] Y. LeCun, C. Cortes, and C. J. Burges, "MNIST handwritten digit database," *ATT Labs*, vol. 2, 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/

2576-2621/20/$31.00 ©2020 IEEE
DOI 10.1109/FCCM48280.2020.00062