Overview of analog implementations of neural networks

Christoph Schröter

Abstract

probleme: viele implementierungsmöglichkeiten für analoge nns -> verschiedene vor/nachteile -> vorstellen und vergleichen

- vorteile nachteile neuronale netze analog digital
- implementierungen die neuronale netze analog verwenden finden und beschreiben - vergleichen, metriken für auswahl für implementierung finden

1 Introduction

Neuronal networks (NNs) are becoming increasingly relevant for industry and research. Their power stems from being able to approximate an arbitrary function from just input and output values through training and backpropagation. Therefore they are heavily used already today, for example in image recognition which can be used in medical applications or for autonomous systems such as automotives or roboters. Even in manufacturing they can be used productively, let it be for product design or quality inspection.
In recent research, analog neural networks (ANNs) are occurring more and more frequently, as further improvement in gemeral purpose processors slows down, while the demand for powerful NNs increases, slowly forcing research away from the traditional digital ones.
Even though over 30 years ago research has been conducted already in this topic [5, 6, 13, 16], new developments and improvements are still being made with great success. Therefore, this paper presents some recent work put into ANNs and compares their architectures against each other. As a result, recommendations can be given on which architecture to use based on metrics of a problem or an already developed neural network which should be transferred to the analog design space.

2 Neural Network Structure

Since for implementing a NN whether its digital or analog the structure is crucial, this section provides a brief summary about basic NN components. Because NNs try to recreate the structure of a brain, there are similarities between the two, however, these are not relevant for implementation and therefore not explained in this paper.

2.1 Neuron

The smallest piece of a neural network is a *neuron* (also called *perceptron*), whose structure is shown in Figure 1.

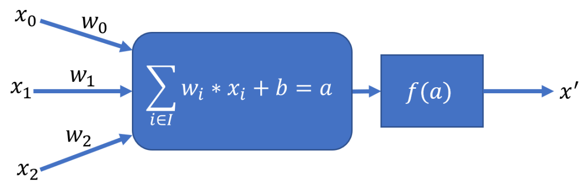

Figure 1. Structure of a neuron. The inputs x_i are multiplied with their corresponding w_i, after that, the bias b is added. Lastly an activation function is applied to determine the output x'.

It sums up an arbitrary number of input values x_i multiplied with their individual weights w_i and adds a bias b to it. The resulting value is called the *activation a* and gets passed to the next neuron (or the output) after applying the *activation function f*. This function plays a huge role in the networks performance, can be selected almost arbitrary and is a research topic on its own. However, simpler activation functions tend to outperform more complex ones, presumably because of a more difficulty training process (see). Currently the most widespread function is the *Rectified Linear Unit* which is defined as $ReLU(x) = max(0, x)$. [10]

2.2 Network

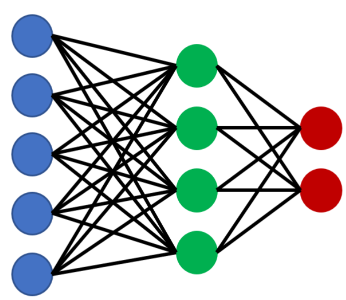

Figure 2. Visualization of a fully connected neural network. Each input (blue) and output (red) is connected to every neuron in the hidden layer (green).

3 Conclusion

References

[1] Stefano Ambrogio, Pritish Narayanan, Hsinyu Tsai, Robert M Shelby, Irem Boybat, Carmelo Di Nolfo, Severin Sidler, Massimo Giordano, Martina Bodini, Nathan CP Farinha, et al.

Equivalent-accuracy accelerated neural-network training using analogue memory. *Nature*, 558(7708):60–67, 2018.

[2] Maxence Bouvier, Alexandre Valentian, Thomas Mesquida, Francois Rummens, Marina Reyboz, Elisa Vianello, and Edith Beigne. Spiking neural networks hardware implementations and challenges: A survey. *J. Emerg. Technol. Comput. Syst.*, 15(2), apr 2019. ISSN 1550-4832. doi: 10.1145/3304103. URL https://doi.org/10.1145/3304103.

[3] Maurizio Capra, Beatrice Bussolino, Alberto Marchisio, Guido Masera, Maurizio Martina, and Muhammad Shafique. Hardware and software optimizations for accelerating deep neural networks: Survey of current trends, challenges, and the road ahead. *IEEE Access*, 8:225134–225180, 2020. doi: 10.1109/ACCESS.2020.3039858.

[4] Lei Deng, Guoqi Li, Song Han, Luping Shi, and Yuan Xie. Model compression and hardware acceleration for neural networks: A comprehensive survey. *Proceedings of the IEEE*, 108 (4):485–532, 2020. doi: 10.1109/JPROC.2020.2976475.

[5] H.P. Graf and L.D. Jackel. Analog electronic neural network circuits. *IEEE Circuits and Devices Magazine*, 5(4):44–49, 1989. doi: 10.1109/101.29902.

[6] H. Harrer, J.A. Nossek, and R. Stelzl. An analog implementation of discrete-time cellular neural networks. *IEEE Transactions on Neural Networks*, 3(3):466–476, 1992. doi: 10.1109/72.129419.

[7] Olga Krestinskaya, Khaled Nabil Salama, and Alex Pappachen James. Analog backpropagation learning circuits for memristive crossbar neural networks. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2018. doi: 10.1109/ISCAS.2018.8351344.

[8] Youjie Li, Jongse Park, Mohammad Alian, Yifan Yuan, Zheng Qu, Peitian Pan, Ren Wang, Alexander Schwing, Hadi Esmaeilzadeh, and Nam Sung Kim. A network-centric hardware/algorithm co-design to accelerate distributed training of deep neural networks. In *2018 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 175–188, 2018. doi: 10.1109/MICRO.2018.00023.

[9] Liqiang Lu, Jiaming Xie, Ruirui Huang, Jiansong Zhang, Wei Lin, and Yun Liang. An efficient hardware accelerator for sparse convolutional neural networks on fpgas. In *2019 IEEE 27th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, pages 17–25, 2019. doi: 10.1109/FCCM.2019.00013.

[10] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions, 2017. URL https://arxiv.org/abs/1710.05941.

[11] Johannes Schemmel, Johannes Fieres, and Karlheinz Meier. Wafer-scale integration of analog neural networks. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 431–438, 2008. doi: 10.1109/IJCNN.2008.4633828.

[12] D.B. Schwartz, R.E. Howard, and W.E. Hubbard. A programmable analog neural network chip. *IEEE Journal of Solid-State Circuits*, 24(2):313–319, 1989. doi: 10.1109/4.18590.

[13] E.A. Vittoz. Analog vlsi implementation of neural networks. In *IEEE International Symposium on Circuits and Systems*, pages 2524–2527 vol.4, 1990. doi: 10.1109/ISCAS.1990.112524.

[14] Peng Yao, Huaqiang Wu, Bin Gao, Jianshi Tang, Qingtian Zhang, Wenqiang Zhang, J Joshua Yang, and He Qian. Fully hardware-implemented memristor convolutional neural network. *Nature*, 577(7792):641–646, 2020.

[15] Huo Yingge, Imran Ali, and Kang-Yoon Lee. Deep neural networks on chip - a survey. In *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 589–592, 2020. doi: 10.1109/BigComp48618.2020.00016.

[16] J.M. Zurada. Analog implementation of neural networks. *IEEE Circuits and Devices Magazine*, 8(5):36–41, 1992. doi: 10.1109/101.158511.