ObrelaJOURNAL Dialogues in Clinical Neuroscience & Mental Health

e-ISSN: 2585-2795 • Printed-ISSN: 2654-1432
DOI: 10.26386/obrela.v3i3.166

State-of-the-art deep learning has a carbon emission problem. Can neuromorphic engineering help?

Evangelos Stromatias

Abstract

Deep learning has attracted a lot of attention from both academic, as well as, industrial parties mainly due to its success when working large datasets and its ability to improve performance by scaling up the size of the models. However, the current trends of training state-of-the-art deep learning models are worrisome. Recent data show that training cutting-edge models is vastly energy inefficient and pose a threat to the democratisation of this technology: the resources required to train a model might be accessible only by a few large corporations in the near future. Moreover, executing trained state-of-the-art deep learning models on mobile devices with limited resources is currently not possible due to the large amounts of computations, memory and energy requirements these models need. Neuromorphic engineering is a relatively recent interdisciplinary research field that attempts to simulate neurons and synapses directly on hardware and at a level that is closer to biology. The advantage of this approach is that because neurons are simulated in an asynchronous manner the overall energy consumption is very low since neurons that do not participate in the computations consume nearly zero energy. While a method to train neural networks directly on neuromorphic devices has yet to be discovered it has already been demonstrated that executing trained neural networks on neuromorphic platforms comes with large energy savings and lower prediction latencies.

Key Words: Deep learning, machine learning, artificial intelligence, carbon emissions, neuromorphic engineering, low-power, low-latency

Corresponding Author: Evangelos Stromatias: Deep Learning Specialist at Medis Medical Imaging Systems, email: evangelos.stromatias@gmail.com

DOI: 10.26386/obrela.v3i3.166

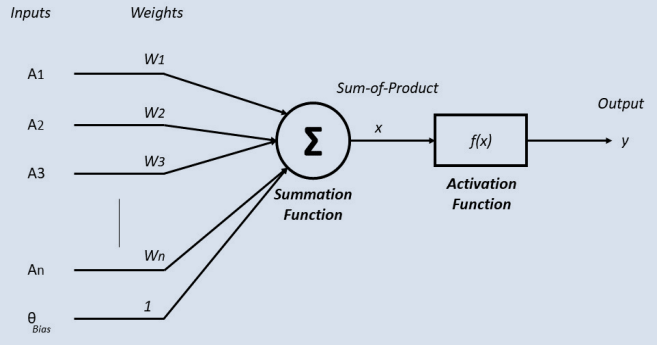
Evangelos Stromatias

1. Introduction

Deep Learning **[1],** part of the broader family of machine learning algorithms that fall under the umbrella of Artificial Intelligence (AI), has attracted the attention of the industry and academic institutions over the past decades. The most popular form of deep learning algorithms is supervised learning in which the goal for a model is to learn a function that maps an input to an output based on example cases. Ultimately, the aim of supervised learning algorithms is to be able to predict the class labels of unseen data correctly. Deep Learning models have surpassed other machine learning methods in virtually all supervised learning tasks and have achieved state-of-the-art results in computer vision tasks (classification, medical imaging, face recognition), speech/audio recognition tasks, machine translation, natural language processing **[2].**

The fundamental building block of Deep learning is the artificial neuron which is loosely based on biological neurons (Figure 1). Artificial neurons when combined together with other artificial neurons, form Artificial Neural Networks (ANNs), while the various ways that artificial neurons can be combined together give rise to the different deep learning architectures. Each artificial neuron receives one or more input through its dendrites, often referred to as weights or model parameters in the literature. These inputs are multiplied with the weights, summed up and passed through a non-linear activation function to become the input to the neurons in the next layer.

Figure 1. The basic components of an artificial neuron. The input signal (A) is multiplied by the neuron's weights (W). The result is added together (x) and then passed through a non-linear activation function (f). Figure recreated from [3].



In supervised learning, an optimization algorithm is used in conjunction with a dataset (each example of the dataset consists of the input data as well as the desired output also known as target label) that is related to the problem that needs to be solved, and a loss function that decides how to update the model parameters. Describing the whole training process in detail is beyond the scope of this article but on a very high level is as follows: The data are presented to the input layer of the network and based on this input the neural network produces an output. Then the loss function is used to compute the error given the prediction of the model and the ground truth (desired output) of that particular input example. The error is then passed to the optimization algorithm that updates the model parameters in a backwards manner, starting from the output layers and moving to the input layers. This process typically takes thousands of iterations until a model has been trained successfully.

Currently there exist a number of Deep Learning architectures for supervised learning and the choice often depends on the type of problem to be solved. Roughly speaking, if the input data are numeric or categorical then it is common to use fully-connected neural networks, in which each neuron in a previous layer is connected to each neuron to the next layer. If the data are images then convolutional neural networks (CNNs) are preferred. CNNs are a popular choice when working with images because they preserve spatial information of the previous layer and because each neuron in the next layer shares weights. This weight-sharing feature results in models with much fewer parameters, and as a consequence becomes easier to scale up to very deep (multiple layers) networks compared to fully-connected networks. Finally, if the data are sequential, for example audio sequences or text manuscripts, it is common to use recurrent neural networks (RNNs). Figure 2 presents some example use-cases of various deep learning architectures applied for cardiovascular image analysis.

As briefly mentioned in a previous paragraph, the learning

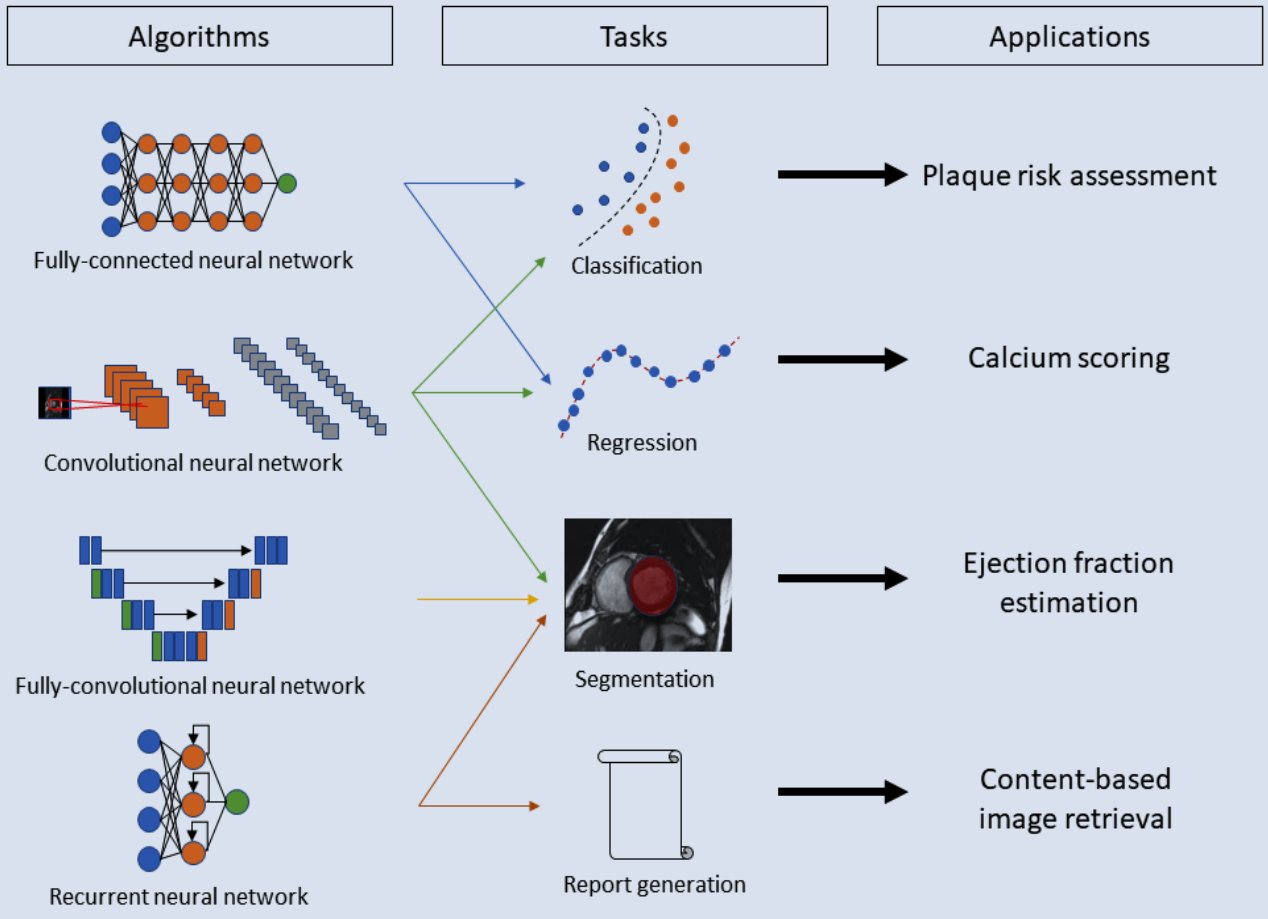
DOI: 10.26386/obrela.v3i3.166

process of deep learning requires thousands of iterations and this number is not known a priori as it depends on the amount of data and the complexity of the task. On top of that, researchers are faced with additional parameters that affect the learning performance of a model. These parameters, referred to as hyperparameters in the literature, include the learning rate (how much the optimisation algorithm will change the model parameters each iteration), total number of layers of neurons, number of neurons per layer, and activation functions of the neurons to name a few. The key point is that a lot of experimentation and empirical effort is required in order to find the optimal combination of hyperparameters that might lead to a successfully trained model.

1.1. Current technologies for accelerating state-of-the-art deep learning research and the potential hazards

To speed up this experimentation process researchers utilise specialised computing hardware called Graphics Processing Units (GPUs), designed for efficient matrix multiplications and convolutions, which constitute the majority of

Figure 2. Example use-cases of various deep learning architectures applied for cardiovascular image analysis. The column "Algorithms" shows some popular deep learning architectures. The middle column "Tasks" presents some category of problems and finally the last column shows some applications where deep learning is currently been used. Figure recreated from [4].



DOI: 10.26386/obrela.v3i3.166

mathematical operations in deep learning. GPUs however, dissipate large amounts of power. For example, Shoeybi et al. 2019 **[5]** trained a very deep network that currently achieves state-of-the-art results on a variety of natural language processing tasks. This model has 8.3 billion trainable parameters and for the training process the authors utilised 512 V100 NVIDIA GPUs. Training this model took 9.2 days of continuous usage of the GPUs. Given that each V100 GPU has a maximum power dissipation of 250 W **[6]**, the total energy required to train this model is twice the average energy than an American household consumes within a year **[7]**. While this is clearly not a scalable solution in terms of carbon emissions required to train a single model, it also poses another threat with regards to the democratisation of deep learning research. The amount of computational resources and energy costs associated with training state-of-art models renders them accessible only to large corporations that have access to these resources. The danger is that in the near future it might become very difficult for academic institutions and startups to research this technology. The trends since 2012 show that the computational power required for training state-of-the-art deep learning models have been increasing exponentially with a 3.4-month doubling time **[8].**

Besides the training process, running state-of-the-art models on mobile devices is nearly impossible due to the memory requirements and the number of operations required to be executed. Currently if a mobile device requires a deep learning algorithm it has to send the user-data to large data centers that are capable of running deep learning models. The data goes through the deep learning model and the prediction is sent back from the data center to the user device **[9]**. This, however, introduces communication overheads thus increasing the latency of the application rendering it unsuitable for real-time applications for example autonomous robotics such as Unmanned aerial vehicles (UaVs) that require quick response times. Moreover, there are certain situations where the user-data cannot leave the device due to legal constraints (e.g. sensitive medical data). In addition, NVIDIA the lead GPU manufacturer used for deep learning "*estimates that 80-90% of the cost of neural networks lies in inference processing*" (Inference is the process of executing neural networks to solve a task after training has been finished) **[10]**. These reasons gave researchers a good incentive for investigating alternative technologies to enable the execution of deep neural networks in a low-power, low-latency manner. Some of these alternative methods are inspired by biology, after all the human brain is capable of solving complex cognitive tasks while having a maximum power dissipation of 20 watts **[11]**.

2. Neuromorphic computing platforms: A biologically inspired method for energy efficient execution of neural networks

Neuromorphic engineering, a term coined by Carver Mead in the early 90s, is an interdisciplinary field that draws inspiration from biology, physics, computer science and electrical engineering with the purpose of designing hardware models of neuronal and sensory circuits. The neural networks models used in neuromorphic engineering are called Spiking Neural Networks (SNNs) and the mathematical equations describing them are based on the empirical model of the Nobel prize winners Hodgkin and Huxley (1952) **[12]** and the experiments they performed on the giant axon of the squid. The main difference of SNNs compared to ANNs is that SNNs introduce the concept of time. In ANNs all neurons in the same layer generate a continuous (real-valued) output signal synchronously at each propagation cycle, which can be vaguely thought of as the firing rate of a neuron within a period of time. Spiking neurons, on the other hand, operate in an asynchronous manner. Each spiking neuron has a membrane potential which changes with time and input signals. Whenever the membrane potential reaches a threshold value a spiking neuron generates a stereotypical all-or-nothing (binary) signal, referred to as spike or action potential (AP). This AP travels along the axon of a neuron to the synapses of other neurons and alters their membrane potential. The advantages of operating in an asynchronous manner is that depending on how neuromorphic circuits have been implemented (analogue,

DOI: 10.26386/obrela.v3i3.166

digital, mixed signal circuits) neurons that do not participate in the computations dissipate very low power leading to significant energy savings. For example, *TrueNorth*, a digital neuromorphic platform developed by IBM and funded by Defense Advanced Research Projects Agency (DARPA), was used to simulate a million spiking neurons in real-time (one millisecond membrane updates) whilst dissipating 63 mW **[13]**. For the same experiment a software simulator running on a conventional computer executed 100 to 200x slower than real-time while dissipating 100,000 to 300,000 times more energy per synaptic event **[13]**. Real-time execution of neural networks can be a desirable property for cognitive neuroscientists and roboticists that would like to test and validate their hypothesis using embodied agents while interacting with the environment **[14]**.

The ability of neuromorphic platforms to simulate neural networks while requiring very little energy has drawn the attention of large semiconductor companies. Intel Labs recently announced their own neuromorphic processor named *Loihi* **[15]**, and IBM's/DARPA-funded *TrueNorth* neuromorphic processor was developed with the purpose of "bringing the sort of intelligence that people usually associate with the cloud down to the handset" **[16]**. A similar interested is reflected in the scientific community with large European research projects like the Human Brain Project **[17], [18],** which has a dedicated neuromorphic computing track funding projects like *BrainScaleS* **[19]** designed to accelerate the simulations of computational neuroscientists by running simulations faster than real-time, and *SpiNNaker* **[20]** which is designed with the purpose of investigating new computational frameworks inspired by the human brain.

2.1. The current state of training deep spiking neural networks

SNNs have been characterized as the 3rd generation of ANNs **[21]** and while it has been theoretically proven that they are more computationally powerful than ANNs **[22]** they still lack the popularity of ANNs mainly because SNNs cannot directly utilise popular ANNs training algorithms such as *Backpropagation* **[23]**. This is because backpropagation-based algorithms require differentiable equations, whereas the equations that describe spiking neurons are discontinuous due to the membrane thresholding function. To overcome this obstacle, many research groups have followed an intermediate approach. Instead of attempting to train directly SNNs they train deep learning models using the conventional backpropagation-based algorithms and then devise methods for converting the trained models to SNNs **[24], [25], [26], [27].** While this approach does not solve the main problem of deep learning which is the energy and time required to train a deep learning model it does allow for a low-latency low-power execution on neuromorphic hardware **[28]**. The main drawback of this method is that the conversion process introduces a drop in the classification performance of the trained model **[24], [25], [26], [27].** More recent efforts have attempted to develop variations of backpropagation capable of working directly with spiking neurons **[29], [30]**. Results are promising as the trained SNNs achieve performance comparable to state-of-the-art for image recognition tasks. However, neither of these methods has attempted to perform the training directly on neuromorphic hardware. Instead these methods use software simulators to train and thus resulting in long training times.

Finally, another approach is to investigate biologically plausible learning methods that work directly with spikes like Spike Timing Dependent Plasticity (STDP) **[31], [32**]. STDP has been derived by biological observations which have demonstrated that synaptic plasticity depends on relative pre- and post-synaptic spike timing. Scientists have investigated the possibility of utilising STDP as an unsupervised feature extractor for machine learning tasks. Diehl et al. in 2015 **[33]** investigated the use of a SNN with plastic synapses using STDP for updating the model parameters on a handwritten digit recognition task and achieved a classification accuracy of 95%. While this accuracy is far from the state-of-the-art for this particular dataset (>99% **[34]**), the advantages of this method is that it is a fully unsupervised

DOI: 10.26386/obrela.v3i3.166

method, no ground truth labels and teacher signals are required and also the STDP algorithm is able to execute directly on neuromorphic platforms **[35].**

3. Conclusions

Tech industry giants recognise the need to bring deep learning models closer to the application and not on remote servers. Intel alone in the past few years has acquired several companies (Movidius, MobileEye, Altera, and Nervana) **[36],** related to dedicated hardware for real-time image processing and execution of neural networks. Unfortunately, the current trend of training state-of-the-art performing deep learning models requires computational and power resources that might only be accessible by large corporations in the near future. Researchers ought to focus on developing computationally efficient hardware and algorithms for deep learning. To this end, Neuromorphic computing platforms offer an attractive alternative for executing efficiently neural network models **[13]**. However, up till now very limited research has been performed on training directly on neuromorphic hardware, which would yield the largest power savings. Additional advantages of training directly on the device would also open new possibilities for neural networks that are able to learn to adapt to changes in the environment and changes on the hardware (e.g. circuit failures or hardware degradation).

References

1. Goodfellow I, Bengio Y, Courville A. 2016. Deep Learning. The MIT Press.
2. Alom, M.Z.; Taha, T.M.; Yakopcic, C.; Westberg, S.; Sidike, P.; Nasrin, M.S.; Hasan, M.; Van Essen, B.C.; Awwal, A.A.S.; Asari, V.K. A State-of-the-Art Survey on Deep Learning Theory and Architectures. *Electronics* 2019, 8: 292. doi:10.3390/electronics8030292
3. Kawaguchi K. A Multithreaded Software Model for Backpropagation Neural Network Applications. University of Texas at El Paso, 2000. https://digitalcommons.utep.edu/dissertations/AAIEP05411
4. Litjens G, Ciompi F, Wolterink JM, de Vos BD, Leiner T, Teuwen J, Išgum I, State-of-the-Art Deep Learning in Cardiovascular Image Analysis, JACC: *Cardiovascular Imaging* 2019, 12 (8): Part 1, 1549-1565. DOI: 10.1016/j.jcmg.2019.06.009
5. Shoeybi M, Patwary M, Puri R, LeGresley P, Casper J, Catanzaro B: Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism. CoRR abs/1909.08053 (2019)
6. NVIDIA (2017, June 21) TESLA V100 GPU ACCELERATOR, Datasheet. Retrieved from https://images.nvidia.com/content/technologies/volta/pdf/437317-Volta-V100-DS-NV-US-WEB.pdf
7. EIA (2019, October 2) How much electricity does an American home use? Retrieved from https://www.eia.gov/tools/faqs/faq.php?id=97&t=3
8. Amodei D, Hernandez D (2018, May 16). AI and Compute. Retrieved from https://openai.com/blog/ai-and-compute/
9. Gomes L. (2017, May 29) Neuromorphic Chips Are Destined for Deep Learning—or Obscurity. IEEE Spectrum. Retreived from https://spectrum.ieee.org/semiconductors/design/neuromorphic-chips-are-destined-for-deep-learningor-obscurity
10. Freund K. (2019, May 9) Google Cloud Doubles Down On NVIDIA GPUs For Inference. Forbes. Retrieved from: https://www.forbes.com/sites/moorinsights/2019/05/09/google-cloud-doubles-down-on-nvidia-gpus-for-inference/#40200e376792
11. Drubach D. The Brain Explained. New Jersey: Prentice-Hall, 2000.
12. Hodgkin A. L. and Huxley A. F. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology* 1952, 117 (4):500–544, ISSN 1469-7793. doi: 10.1113/jphysiol.1952.sp004764. URL http://dx.doi.org/10.1113/jphysiol.1952.sp004764.
13. Merolla PA, Arthur JV, Alvarez-Icaza R, Cassidy AS, Sawada J, Akopyan P, Jackson BL, Imam N, Guo C, Nakamura Y, Brezzo B, Vo I, Esser SK, Appuswamy R, Taba B, Amir A, Flickner MD, Risk WP, Manohar R, Modha DS. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* 2014, 345(6197):668–673, doi: 10.1126/science.1254642.
14. Galluppi F, Brohan K, Davidson S, Serrano-Gotarredona T, Perez Carrasco JA, Linares-Barranco B, Furber S. A real-time, event-driven neuromorphic system for goal-directed attentional selection. In Neural Information Processing, pages 226–233. Springer, 2012. doi.org/10.1007/978-3-642-34481-7_28

DOI: 10.26386/obrela.v3i3.166

15. Davies M et al., Loihi: A Neuromorphic Manycore Processor with On-Chip Learning. *IEEE* Micro 2018, 38 (1): 82-99. doi: 10.1109/MM.2018.112130359
16. Monroe D. Neuromorphic computing gets ready for the (really) big time. Commun. *ACM* 2014, 57(6):13–15, doi: 10.1145/2601069.
17. Human Brain Project, 2013. URL https://www.humanbrainproject.eu.
18. Markram H, Meier K, Lippert T, Grillner S, Frackowiak R, Dehaene S, Knoll A, Sompolinsky H, Verstreken K, DeFelipe J, Grant S, Changeux J-P, Saria A. Introducing the human brain project. *Procedia Computer Science* 2011, 7(0):39 – 42. doi: http: //dx.doi.org/10.1016/j.procs.2011.12.015.
19. Calimera A, Macii E, Poncino M. The Human Brain Project and neuromorphic computing. *Funct Neurol*. 2013, 28(3):191-6. doi: 10.11138/FNeur/2013.28.3.191.
20. Furber SB, Galluppi F, Temple S, Plana LA, The SpiNNaker Project, *Proceedings of the IEEE* 2014, 102 (5): 652-665, DOI: 10.1109/JPROC.2014.2304638.
21. Maass W. Networks of spiking neurons: The third generation of neural network models. *Neural Networks* 1997, 10(9):1659–1671 doi: http: //dx.doi.org/10.1016/S0893-6080(97)00011-7.
22. Maass W and Markram H. On the computational power of circuits of spiking neurons. *Journal of Computer and System Sciences* 2004, 69(4):593–616. doi: http://dx.doi.org/10.1016/j.jcss.2004.04.001.
23. Werbos P. Beyond regression: new tools for prediction and analysis in the behavioral sciences. Ph.D. dissertations, Committee on Appl. Math., Harvard University, Cambridge, MA, Nov. 1974.
24. Perez-Carrasco J-A, Zhao B, Serrano C, Acha B, Serrano-Gotarredona T, Chen C, Linares-Barranco B. Mapping from frame-driven to frame-free event- driven vision systems by low-rate rate coding and coincidence processing–application to feedforward convnets. Pattern Analysis and Machine Intelligence, *IEEE Transactions* 2013, 35(11):2706–2719. doi: 10.1109/TPAMI. 2013.71.
25. O'Connor P, Neil D, Liu S-C., Delbruck T, Pfeiffer M. Real-time classification and sensor fusion with a spiking deep belief network. *Frontiers in Neuroscience* 2013, 7(178). doi: 10.3389/fnins.2013.00178.
26. Diehl PU, Neil D, Binas J, Cook M, Liu S, Pfeiffer M. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing," 2015 International Joint Conference on Neural Networks (IJCNN), Killarney, 2015, pp. 1-8. doi: 10.1109/IJCNN.2015.7280696
27. Rueckauer B, Lungu I-A, Hu Y, Pfeiffer M, Liu S-C. Conversion of Continuous-Valued Deep Networks to Efficient Event-Driven Networks for Image Classification. *Frontiers in Neuroscience* 2017 December 2017. Doi: 10.3389/fnins.2017.00682
28. Stromatias E, Neil D, Galluppi F, Pfeiffer M, Liu S-C, Furber S. Scalable energy-efficient, low-latency implementations of spiking deep belief networks on SpiNNaker. In Accepted in 2015 International Joint Conference on Neural Networks (IJCNN), Jul 2015.
29. Lee JH, Delbruck T, Pfeiffer M. Training deep spiking neural networks using backpropagation. *Front. Neurosci* 2016, 10:508. doi: 10.3389/fnins.2016.00508
30. Shrestha SB and Orchard G. SLAYER: Spike Layer Error Reassignment in Time. Advances in Neural Information Processing Systems 31, 2018.
31. Markram H, Lbke J, Frotscher M, Sakmann B. Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science* 1997, 275(5297):213–215. DOI: 10.1126/science.275.5297.213
32. Bi GQ and Poo MM. Synaptic modifications in cultured hippocampal neurons: de- pendence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of Neuroscience* 1998, 18(24):10464–10472. doi.org/10.1523/JNEUROSCI.18-24-10464.1998
33. Diehl PU and Cook M. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in computational neuroscience* 2015, 9. https://doi.org/10.3389/fncom.2015.00099
34. Byerly A Kalganova T, Dear I. A Branching and Merging Convolutional Network with Homogeneous Filter Capsules. abs/2001.09136, 2020. Url: https://arxiv.org/abs/2001.09136
35. Galluppi F, Lagorce X, Stromatias E, Pfeiffer M, Plana LA, Furber SB, Benosman RB. A framework for plasticity implementation on the spinnaker neural architecture. *Frontiers in Neuroscience* 2014, 8(429). doi: 10.3389/fnins.2014. 00429.
36. Freund K. (2018, June 1) Intel Shows Off Its AI Chips And Chops. Forbes. Retrieved from: https://www.forbes.com/sites/moorinsights/2018/06/01/intel-shows-off-its-ai-chips-and-chops/#49acfbb66439