Scaling Neural Networks Through Sparsity

Louis Kirsch
http://louiskirsch.com

July 2018, *updated October 2018*

Contemporary neural network architectures are several orders of magnitude smaller than a human brain. But we need extraordinary large neural networks if knowledge about the entire interaction history of a reinforcement agent should be distilled into the network. In the simplest case, this may be a policy to gain maximum reward in a certain environment. Much larger networks will be required if predictive understanding about the environment should be included, such as a detailed model of the environment. This predictive understanding helps to solve new, currently unknown, tasks beyond learning to maximize the reward on the current task. Thus, even if we could find the right learning algorithm that exhibits similar behavior to biological intelligence, it is unlikely that current small neural networks would scale to human intelligence.

Furthermore, the current deep learning revolution can in large parts be attributed to larger neural networks and increased data set sizes [11]. Hence, it is conceivable that larger neural network architectures that learn from large amounts of data from many different tasks already exhibit intelligent behavior without much further development of new learning algorithms and inductive biases. We conclude that our goal should be to explore the limit of extremely large or even infinitely sized neural networks. With current GPU hardware, we quickly reach computational and memory limits [15, 4, 19]. In this short report, we will explore how sparsity and conditional computation can help us to get around these issues.

1 The Importance of Sparsity

This section describes how sparsity can help us to scale artificial neural networks. Sparsity refers both to the connectivity between two layers, i.e. zero values in the weight matrix, as well as zeros in the activations of each layer. We differentiate between unconditional and conditional sparsity: Unconditional sparsity is independent of the input, while conditional sparsity describes sparsity that occurs for a specific input.

The neuroscientific argument. While the human brain has around 150 trillion synapses [16] our largest neural networks only achieve a few billion parameters at most [1]. Furthermore, a single synapse is known to be more complex than a single parameter in an artificial neural network. The problem with training really large neural networks lies both in memory and compute. The number of parameters increases both memory and compute resources requirements linearly. In the case of neurons, the growth is even quadratic. This is very unlike the brain, where connectivity is very sparse. In the brain, due to the sparse connectivity, computation would scale only linearly in the number of neurons instead of quadratically. Furthermore, energy is mostly required proportional to the number of times a neuron fires. If a neuron does not fire,

this is similar to having a zero activation in an artificial neural network. Contemporary GPU architectures cannot leverage this kind of sparsity.

The evidence in existing neural networks. Existing neural networks are already automatically quite sparse. This includes weights that have very small (negligible) magnitudes and in particular zero activations due to the ReLU non-linearity. Furthermore, we often pre-specify a sparsity inductive bias in the form of CNNs that can be computationally leveraged on GPUs but describes a fixed sparsity pattern that can not be learned.

Sparsity as regularization. It can be shown that employing L1-regularization or other Bayesian priors on model parameters can incur additional sparsity [13, 7, 15, 12, 2]. This sparsity is already an interesting research area because it tends to increase generalization. Thus, we can increase the value of sparsity even further by introducing compute and memory benefits. Currently, computational benefits are very limited. We have to rely on structured sparsity such as [15, 21] where entire neurons or convolutional filters are removed to gain computational speedups. Furthermore, these speedups only apply after pruning which cannot be leveraged during training.

1.1 Near-term Solutions

In the near-term we will probably not be able to achieve a computational complexity class such as it can be found in the brain where energy consumption is proportional to neuron firing. Nevertheless, there are several approaches we might want to pursue.

Modularization. In the brain, we can observe that different areas have different functionality [18]. We could enforce a similar structure by portioning our computational model into modules - entities that compute different functions depending on the input. In this manner, we can actually omit entire non-relevant connections in the neural network. Thus, this method allows leveraging both unconditional, as well as conditional sparsity. For instance, we proposed a learning algorithm that induces this modularization without any regularization [8]. On the downside, this requires the task to be actually partitionable into such modules and an overhead in terms of a controller that has to select modules and the training of the architecture. This means this method is not necessarily applicable to all problems.

Row Sparsity. If the activations are highly sparse, such as in the case of ReLU activations, we can simply prune entire rows from the weight matrix for each switched off unit. It is easy to see that this does neither change the value of the function nor the gradients w.r.t. to these pruned weights. We can further encourage this behavior by employing a sparsity regularizer (e.g. L1) that further improves generalization. If only a constant number of units is active for any given datapoint in each layer, then computing the feed-forward activations and gradients for this datapoint is only linear in the number of neurons N per layer compared to the $O(N^2)$ compute necessary to perform dense-matrix-matrix multiplications. We are currently investigating this approach further and will publish the results.

Block-Sparsity. A very exciting new development is the idea of block-sparsity GPU kernels [4, 14]. It allows partitioning a weight matrix into several small square-chunks. The matrix multiplication can then be efficiently executed in parallel as before, with the additional benefit that chunks that have entirely zeros only do not have to be evaluated. Research has shown that this may lead to speedups of several magnitudes. Current techniques to generate such sparsity patterns rely heavily on heuristics [4, 14]. Future work on principled Bayesian methods that can lead to efficient sparsity patterns is a very interesting research direction. On the downside, this

technique does currently only apply to the weight matrix and not the highly sparse activations. Furthermore, it only considers unconditional sparsity.

1.2 Long-term Solutions

In the long term, we should aim at getting as close as possible to the computational model in the brain. This means we require new hardware and software solutions.

Hardware and software solutions. There has been an interesting development on gains in memory usage and evaluation compute through compression algorithms and other hardware and software related changes [19, 20, 17]. This includes the development of FPGAs for deep learning and other new hardware architectures that can directly leverage sparsity [10]. It is entirely feasible that this would allow increasing the effective size of neural networks by several magnitudes. Unfortunately, due to hardware-related long development cycles, it is unlikely to see these approaches very soon.

Finally, the brute-force solution is to wait for Moore's law to catch-up. That said, Moore's law has stagnated over the last few years (we reach physical limitations in terms of scale) [9]. Furthermore, the brain achieves efficient computation with less than 20 Watts of power [5, 6]. There should be a better approach than requiring large clusters to imitate simple human-like intelligence.

1.3 The Batching Issue

Current machine learning methods rely heavily on mini-batching to estimate gradients [3]. If computation depends on each sample, there will be fewer gradients per parameter in a given mini-batch, thus increasing the noise in the overall gradient. Possible alternatives are either increasing the batch size, improved online learning methods or smart algorithms that can group data such that similar data is executed together. The latter may lead to more problems though, as the gradients would rely on non-iid data.

References

[1] Adam Coates, Brody Huval, Tao Wang, David Wu, Bryan Catanzaro, and Ng Andrew. Deep learning with COTS HPC systems. In *International Conference on Machine Learning*, pages 1337–1345, 2013.

[2] Soumya Ghosh, Jiayu Yao, and Finale Doshi-Velez. Structured Variational Learning of Bayesian Neural Networks with Horseshoe Priors. *arXiv preprint arXiv:1806.05975*, 6 2018.

[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[4] Scott Gray, Alec Radford, and Diederik P Kingma. GPU Kernels for Block-Sparse Weights. Technical report, OpenAI, 2017.

[5] Pavel D Hrdina. Basic Neurochemistry: Molecular, Cellular and Medical Aspects. *Journal of Psychiatry and Neuroscience*, 21(5):352, 1996.

[6] Ferris Jabr. Does thinking really hard burn more calories. *scientific american*, 18, 2012.

[7] Diederik P. Kingma, Tim Salimans, and Max Welling. Variational Dropout and the Local Reparameterization Trick. In *Advances in Neural Information Processing Systems*, pages 2575–2583, 6 2015.

[8] Louis Kirsch, Julius Kunze, and David Barber. Modular Networks: Learning to Decompose Neural Computation. *Advances in Neural Information Processing Systems*, 2018.

[9] Laszlo B Kish. End of Moore's law: thermal (noise) death of integration in micro and nano electronics. *Physics Letters A*, 305(3-4):144–149, 2002.

[10] Griffin Lacey, Graham W Taylor, and Shawki Areibi. Deep learning on fpgas: Past, present, and future. *arXiv preprint arXiv:1602.04283*, 2016.

[11] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[12] Juho Lee, Saehoon Kim, Jaehong Yoon, Hae Beom Lee, Eunho Yang, and Sung Ju Hwang. Adaptive Network Sparsification via Dependent Variational Beta-Bernoulli Dropout. *arXiv preprint arXiv:1805.10896*, 5 2018.

[13] Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational Dropout Sparsifies Deep Neural Networks. *arXiv preprint arXiv:1701.05369*, 1 2017.

[14] Sharan Narang, Eric Undersander, and Gregory Diamos. Block-Sparse Recurrent Neural Networks. *arXiv preprint arXiv:1711.02782*, 11 2017.

[15] Kirill Neklyudov, Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Structured Bayesian Pruning via Log-Normal Multiplicative Noise. In *Advances in Neural Information Processing Systems*, pages 6775–6784, 5 2017.

[16] Bente Pakkenberg, Dorte Pelvig, Lisbeth Marner, Mads J Bundgaard, Hans Jørgen G Gundersen, Jens R Nyengaard, and Lisbeth Regeur. Aging and the human neocortex. *Experimental gerontology*, 38(1-2):95–99, 2003.

[17] Angshuman Parashar, Minsoo Rhu, Anurag Mukkara, Antonio Puglielli, Rangharajan Venkatesan, Brucek Khailany, Joel Emer, Stephen W Keckler, and William J Dally. SCNN: An Accelerator for Compressed-sparse Convolutional Neural Networks. In *ACM SIGARCH Computer Architecture News*, volume 45, pages 27–40. ACM, 2017.

[18] Mahdi Ramezani, Kris Marble, H Trang, Ingrid S Johnsrude, and Purang Abolmaesumi. Joint sparse representation of brain activity patterns in multi-task fMRI data. *IEEE Transactions on Medical Imaging*, 34(1):2–12, 2015.

[19] Minsoo Rhu, Mike O'Connor, Niladrish Chatterjee, Jeff Pool, and Stephen W. Keckler. Compressing DMA Engine: Leveraging Activation Sparsity for Training Deep Neural Networks. In *High Performance Computer Architecture (HPCA), 2018 IEEE International Symposium*, pages 78–91. IEEE, 2018.

[20] Vivienne Sze, Yu Hsin Chen, Tien Ju Yang, and Joel S. Emer. Efficient Processing of Deep Neural Networks: A Tutorial and Survey, 2017.

[21] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning Structured Sparsity in Deep Neural Networks. 8 2016.