



Deep Learning on SpiNNaker

MASTER THESIS

Jonas Fassbender

jonas@fassbender.dev

In the course of studies
HIGH PERFORMANCE COMPUTING WITH DATA SCIENCE

For the degree of
MASTER OF SCIENCE

The University of Edinburgh

First supervisor: Caoimhín Laoide-Kemp
EPCC, University of Edinburgh

Second supervisor: Dr Kevin Stratford
EPCC, University of Edinburgh

Third supervisor: Dr Alan Stokes
APT, University of Manchester

Edinburgh, August 2020

Declaration

I declare that this dissertation was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

Jonas Fassbender
August 2020

Abstract

Contents

1	Introduction	1
2	Background	2
2.1	An Introduction to Deep Learning	2
2.2	Benchmarking Deep Learning Systems for Computer Vision	2
2.3	SpiNNaker as a Neuromorphic Computer Architecture	2
3	Related Work	3
4	Deep Learning on SpiNNaker	3
5	Benchmark	3
6	Discussion	3
7	Conclusion	3
8	Next Steps	3
	References	6

1. Introduction

Deep learning is revolutionizing the world. It has become part of our daily lives as consumers, powering major software products—from recommendation systems and translation tools to web search (LeCun et al., 2015). Major breakthroughs in fields like computer vision (Krizhevsky et al., 2012) or natural language processing (Hinton et al., 2012) were achieved through the use of deep learning. It has emerged as a driving force behind discoveries in numerous domains like particle physics (Ciodaro et al., 2012), drug discovery (Ma et al., 2015), genomics (Leung et al., 2014) and gaming (Silver et al., 2016).

Deep learning has become so ubiquitous that we are changing the way we build modern hardware to account for its computational demands. From the way edge devices like mobile phones or embedded systems are built (Deng, 2019) and modern CPUs (Perez, 2017) to specialized hardware designed only for deep learning models, such as Google’s tensor processing unit (TPU) (Jouppi et al., 2017) or NVIDIA’s EGX Edge AI platform (Boitano, 2020). Whole state-of-the-art supercomputers are built solely for deep learning. An example would be a supercomputer built by Microsoft for OpenAI, which is part of the Azure cloud (Langston, 2020).

Hardware manufacturer are faced with a major challenge in meeting the computational demands arising from inference, and more importantly, training deep learning models. OpenAI researchers have estimated that the computational costs of training increases exponentially; approximately every 3.4 months the cost doubles (Amodei et al., 2019). Amodei et al. (2019) claims the deep reinforcement learning agent AlphaGo Zero—the successor of the famous AlphaGo program, which was able to beat Go world champion Lee Sedol (Silver et al., 2017)—to be the system with the highest computational demands of approximately 1850 petaflop/s-days. AlphaGo Zero was trained for 40 days on a machine with 4 TPUs (Silver et al., 2017). With the end of Moore’s Law (Loeffler, 2018), chip makers have to get creative in scaling up computing, the same way machine learning researchers are scaling up their models (Simonite, 2016). Therefore production and research into new hardware designs for deep learning are well on the way.

Another field which has high computational demands for very specific tasks and algorithms is computational neuroscience. Computational neuroscience has long been linked to deep learning, which has its origin in research done by neuroscientists (McCulloch and Pitts, 1943). While in the recent past deep learning research has been more focused on mathematical topics like statistics and probability theory, optimization or linear algebra, researchers are again looking to neuroscience to further improve the capabilities of deep learning models (Marblestone et al., 2016).

But the algorithms developed by computational neuroscientists are not the only aspect drawing attention from the deep learning community. Computational neuroscience has long a long standing history of developing custom hardware for the efficient modeling of the human brain, so called neuromorphic computing. Neuromorphic computing—a computer architecture inspired by the biological nervous system—has been around since the 1980s (Mead, 1989). Today, neuromorphic computers are being developed to meet the demands for efficient computing needed to run large-scale spiking neural networks used for modeling brain functions (Furber, 2016). While being developed mainly for the task of modeling the human brain, deep learning has been linked to neuromorphic computing, especially in the context of commercial usability (Gomes, 2017). Both the low energy demands of neuromorphic computers—such as IBM’s True North (Cassidy et al., 2013) or The University of Manchester’s Spiking Neural Network Architecture (SpiNNaker) (Furber et al., 2006)—and their scalability and massive-parallelism are intriguing for two very important use cases of deep learning:

(i) edge computing, for example robotics and mobile devices, (ii) supercomputers and the cloud-era (Gomes, 2017).

This thesis investigates the performance of SpiNNaker machines for deep learning by training the state-of-the-art computer vision model ResNet-50 (He et al., 2015) under the closed division rules of the MLPerf benchmark (Mattson et al., 2019). In order to benchmark ResNet-50 on SpiNNaker a prototypical implementation was developed as part of this thesis.

- here a paragraph about the results

Section 2 presents the background of this thesis. An introduction to deep learning is given in Section 2.1, as well as an overview of the benchmark in Section 2.2. Section 2.3 describes the SpiNNaker architecture and compares it to current deep learning hardware. Related work can be found in Section 3. Section 4 presents the architecture of the prototype developed for benchmarking and Section 5 presents the benchmarks and its results. In Section 6 the results of the benchmark are discussed, as well as the development process. Section 7 contains the conclusion, while Section 8 outlines the next steps for further increasing the performance of SpiNNaker by enhancing the prototype.

2. Background

2.1 An Introduction to Deep Learning

1. history of DL
2. clarify that DNNs are statistical methods (glorified non-linear classifiers) not biological like SNNs
3. concepts of the MLP:
 - layers
 - activations
 - forward- and backward-pass
 - SGD
 - ...
4. CNNs

2.2 Benchmarking Deep Learning Systems for Computer Vision

1. short section about imagenet and ilsvrc and their importance for computer vision
2. ResNet50 and residual stuff
3. paragraph about MLPerf

2.3 SpiNNaker as a Neuromorphic Computer Architecture

1. describe spinnaker and the spinnaker architecture
2. compare to other DL accelerators (GPUs and TPUs)

3. Related Work

1. SNNToolbox for translating DNNs to SNNs (only inference)
2. TrueNorth has a paper about its DL implementation
3. (optional) The 2011 paper about mapping MLP's and recurrent networks onto SpiNNaker

4. Deep Learning on SpiNNaker

- concepts (layers, neurons, ...)
- partitions and how that allows me to use min keys to discern between what I have received
- communication structure (partitions and global partition manager)
- ping-pong
- graph structure (especially focused on edge and host-SpiNN communication)
- interpreting neurons as domain decomposition over linear algebra compute graph
- backward pass: gradients computed two times so comm fabric is not overly used by unique partitions
- How I crushed nd -kernels into a single blob of weights (same for 2D convolutions even though less interesting)

5. Benchmark

6. Discussion

- space used inefficiently (cores and memory) → better domain decomposition

7. Conclusion

8. Next Steps

- multiple copies of the same network on the same machine → use all resources available
- better domain decomposition (SpiNNaker application graph or custom solution (application graph not helpful for neurons which become too big))
- smart algorithms vs. integrating with state-of-the-art libraries (investing time in stuff like SLIDE and the one paper by the Austrian guys about sparse connections explicitly mentioning SpiNNaker and neuromorphic chips or rather work on a trans-/compiler that efficiently translates linear algebra operations (like TF, PyTorch,...) onto SpiNNaker)
- integrate into compiler projects like Apache-TVM, XLA, Glow, nGraph, etc.
- implementing ONNX spec to make it easy for developers to use SpiNNaker (develop in PyTorch → run on SpiNNaker)

References

- Dario Amodei, Danny Hernandez, Girish Sastry, Jack Clark, Greg Brockman, and Ilya Sutskever. AI and Compute. <https://openai.com/blog/ai-and-compute/>, 2019.
- Justin Boitano. How NVIDIA EGX Is Forming Central Nervous System of Global Industries, 05 2020. URL <https://blogs.nvidia.com/blog/2020/05/15/egx-security-resiliency/>.
- Andrew Cassidy, Paul Merolla, John Arthur, S.K. Esser, Bryan Jackson, Rodrigo Alvarez-Icaza, Pallab Datta, Jun Sawada, Theodore Wong, Vitaly Feldman, Arnon Amir, Daniel Ben Dayan Rubin, Filipp Akopyan, Emmett McQuinn, W.P. Risk, and Dharmendra Modha. Cognitive computing building block: A versatile and efficient digital neuron model for neurosynaptic cores. 08 2013. doi: 10.1109/IJCNN.2013.6707077.
- T Ciodaro, D Deva, Joao Seixas, and Denis Oliveira Damazio. Online particle detection with neural networks based on topological calorimetry information. *Journal of Physics: Conference Series*, 368, 06 2012. doi: 10.1088/1742-6596/368/1/012030.
- Yunbin Deng. Deep learning on mobile devices - A review. *CoRR*, abs/1904.09274, 2019. URL <http://arxiv.org/abs/1904.09274>.
- Steve Furber. Large-scale neuromorphic computing systems. *Journal of Neural Engineering*, 13, 08 2016. doi: 10.1088/1741-2560/13/5/051001.
- Steve Furber, Steve Temple, and Andrew Brown. High-performance computing for systems of spiking neurons. 2, 01 2006.
- Lee Gomes. Neuromorphic Chips Are Destined for Deep Learning—or Obscurity, 05 2017. URL <https://spectrum.ieee.org/semiconductors/design/neuromorphic-chips-are-destined-for-deep-learning-or-obscurity>.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL <http://arxiv.org/abs/1512.03385>.
- G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.
- Norman P. Jouppi, Cliff Young, Nishant Patil, David A. Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, Rick Boyle, Pierre-luc Cantin, Clifford Chao, Chris Clark, Jeremy Coriell, Mike Daley, Matt Dau, Jeffrey Dean, Ben Gelb, Tara Vazir Ghaemmaghami, Rajendra Gottipati, William Gulland, Robert Hagmann, Richard C. Ho, Doug Hogberg, John Hu, Robert Hundt, Dan Hurt, Julian Ibarz, Aaron Jaffey, Alek Jaworski, Alexander Kaplan, Harshit Khaitan, Andy Koch, Naveen Kumar, Steve Lacy, James Laudon, James Law, Diemthu Le, Chris Leary, Zhuyuan Liu, Kyle Lucke, Alan Lundin, Gordon MacKean, Adriana Maggiore, Maire Mahony, Kieran Miller, Rahul Nagarajan, Ravi Narayanaswami, Ray Ni, Kathy Nix, Thomas Norrie, Mark Omernick, Narayana Penukonda, Andy Phelps, Jonathan Ross, Amir Salek, Emad Samadiani, Chris Severn, Gregory Sizikov, Matthew Snelham, Jed Souter, Dan Steinberg, Andy Swing, Mercedes Tan, Gregory Thorson, Bo Tian, Horia Toma,

- Erick Tuttle, Vijay Vasudevan, Richard Walter, Walter Wang, Eric Wilcox, and Doe Hyun Yoon. In-datacenter performance analysis of a tensor processing unit. *CoRR*, abs/1704.04760, 2017. URL <http://arxiv.org/abs/1704.04760>.
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.
- Jennifer Langston. Microsoft announces new supercomputer, lays out vision for future AI work, 05 2020. URL <https://blogs.microsoft.com/ai/openai-azure-supercomputer/>.
- Yann LeCun, Y. Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–44, 05 2015. doi: 10.1038/nature14539.
- Michael Leung, Hui Xiong, Leo Lee, and Brendan Frey. Deep learning of the tissue-regulated splicing code. *Bioinformatics (Oxford, England)*, 30:i121–i129, 06 2014. doi: 10.1093/bioinformatics/btu277.
- John Loeffler. No More Transistors: The End of Moore’s Law, 11 2018. URL <https://interestingengineering.com/no-more-transistors-the-end-of-moores-law>.
- Junshui Ma, Robert Sheridan, Andy Liaw, George Dahl, and Vladimir Svetnik. Deep neural nets as a method for quantitative structure–activity relationships. *Journal of chemical information and modeling*, 55, 01 2015. doi: 10.1021/ci500747n.
- Adam H. Marblestone, Greg Wayne, and Konrad P. Kording. Towards an integration of deep learning and neuroscience. *bioRxiv*, 2016. doi: 10.1101/058545. URL <https://www.biorxiv.org/content/early/2016/06/13/058545>.
- Peter Mattson, Christine Cheng, Cody Coleman, Greg Diamos, Paulius Micikevicius, David Patterson, Hanlin Tang, Gu-Yeon Wei, Peter Bailis, Victor Bittorf, David Brooks, Dehao Chen, Debojyoti Dutta, Udit Gupta, Kim Hazelwood, Andrew Hock, Xinyuan Huang, Bill Jia, Daniel Kang, David Kanter, Naveen Kumar, Jeffery Liao, Guokai Ma, Deepak Narayanan, Tayo Oguntebi, Gennady Pekhimenko, Lillian Pentecost, Vijay Janapa Reddi, Taylor Robie, Tom St. John, Carole-Jean Wu, Lingjie Xu, Cliff Young, and Matei Zaharia. Mlperf training benchmark, 2019.
- Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, pages 115–133, 12 1943. doi: 10.1007/BF02478259. URL <http://link.springer.com/10.1007/BF02478259>.
- Carver Mead. Analog vlsi and neural systems. 1989.
- Andres Felipe Rodriguez Perez. Intel Processors for Deep Learning Training, 11 2017. URL <https://software.intel.com/content/www/us/en/develop/articles/intel-processors-for-deep-learning-training.html>.
- David Silver, Aja Huang, Christopher Maddison, Arthur Guez, Laurent Sifre, George Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–489, 01 2016. doi: 10.1038/nature16961.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, 550:354–359, 10 2017. doi: 10.1038/nature24270.

Tom Simonite. Moore’s Law Is Dead. Now What?, 05 2016. URL <https://www.technologyreview.com/2016/05/13/245938/moores-law-is-dead-now-what/>.