Sparse-Matrix based Kernels for Expediting Quantum Neural Networks on Hardware Accelerators

First Author[1][0000−1111−2222−3333], Second Author[2,3][1111−2222−3333−4444], and Third Author[3][2222−−3333−4444−5555]

[1] Princeton University, Princeton NJ 08544, USA
[2] Springer Heidelberg, Tiergartenstr. 17, 69121 Heidelberg, Germany
lncs@springer.com
http://www.springer.com/gp/computer-science/lncs
[3] ABC Institute, Rupert-Karls-University Heidelberg, Heidelberg, Germany
{abc,lncs}@uni-heidelberg.de

Abstract. The abstract should briefly summarize the contents of the paper in 15–250 words.

Keywords: First keyword · Second keyword · Another keyword.

1 Introduction

Researchers have applied Deep Neural Networks (DNN) [1] to a range of scientific and societal applications, which includes deploying text and speech recognition systems and exploring the vast unknown frontiers of science. Loosely modeled after the complex structure of a biological learning system, DNNs derive an input-output mapping, through an interconnected set of nodes. Even though substaintial strides have been taken in reducing the computational expense of training a DNN, researchers have often incorporated concepts from different fields to ascend barriers. For example, expediting matrix computations on hardware accelerators is insipred from the field of algorithm-hardware co-design; replacing backpropagation algorithm with genetic algorithm for DNN optimization is an idea drawn from natrual meta-heuristics.

With the advent of Quantum Computing (QC), integrating the main components of QC with DNNs is an extremely exciting avenue. The electic combination of quantum mechanics with neural computation gave rise to Quantum Neural Network models (QNN) [2] [3]. The motivation behind developing QNNs is to circumvent the limitations faced by classic DNNs in handling complex unstructured datasets by resorting to features of quantum entanglement, superposition and unitary transformations. The work done in developing QNNs has revolved around generalizing the perceptron structure and the associated activation function, as the classical representations resist the mathematical formulation of quantum mechanics [4]. The principle theoretical challenge faced in QNN development is the linear nature of quantum processes which is a hurdle as majority of the

networks have a crucial dependence on non-linear activation functions. Despite such complications, QNNs present a enticing approach to develop large scale networks with reduced computational expense.

The latency observed when a QNN is subjected to large volumes of datasets contributes to sustaintial overhead. The main technique employed to reduce inference latency for DNNs is to prune the network, which transforms the dense network to a sparse model by trimming off network connections which reduce the model accuracy. However, there exists a trade-off between the amount of sparsity introduced in a network and model validity. Researchers have dwelled on the level of fine tunning must be applied, which element-wise pruning[5] being the most granular and sparsity pattern being the least granular[6].

To accelerate the sparse QNN, the core computations are ported onto accelerator platforms like GPUs and FPGAs. The hardware architecture of these accelerators is radically different from a CPU architecture, which emphasis placed on data crunching rather than accomodating multiple tasks. GPUs have multiple multiprocessors, each of which has multiple Streaming Processors (SM) organized into blocks, having larger globally shared memory rather than locally shared memory. Organizing these blocks along with shared memory offers ultra fine-tuned task and data parallelism, resulting in high throughput and bottleneck elimination.

1.1 A Subsection Sample

Please note that the first paragraph of a section or subsection is not indented. The first paragraph that follows a table, figure, equation etc. does not need an indent, either.

Subsequent paragraphs, however, are indented.

Sample Heading (Third Level) Only two levels of headings should be numbered. Lower level headings remain unnumbered; they are formatted as run-in headings.

Sample Heading (Fourth Level) The contribution should contain no more than four levels of headings. Table 1 gives a summary of all heading levels.

Table 1. Table captions should be placed above the tables.

Heading level	Example	Font size and style
Title (centered)	**Lecture Notes**	14 point, bold
1st-level heading	**1 Introduction**	12 point, bold
2nd-level heading	**2.1 Printing Area**	10 point, bold
3rd-level heading	**Run-in Heading in Bold.** Text follows	10 point, bold
4th-level heading	*Lowest Level Heading.* Text follows	10 point, italic

Displayed equations are centered and set on a separate line.

$$x + y = z \tag{1}$$

Please try to avoid rasterized images for line-art diagrams and schemas. Whenever possible, use vector graphics instead (see Fig. 1).

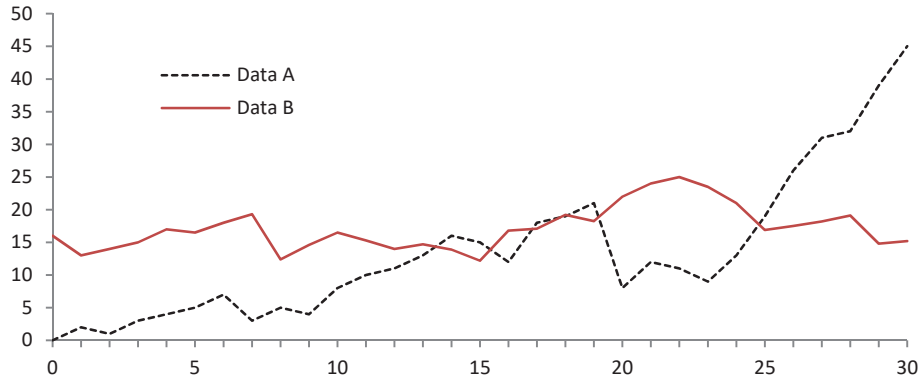

Fig. 1. A figure caption is always placed below the illustration. Please note that short captions are centered, while long ones are justified by the macro package automatically.

Theorem 1. *This is a sample theorem. The run-in heading is set in bold, while the following text appears in italics. Definitions, lemmas, propositions, and corollaries are styled the same way.*

Proof. Proofs, examples, and remarks have the initial word in italics, while the following text appears in normal font.

For citations of references, we prefer the use of square brackets and consecutive numbers. Citations using labels or the author/year convention are also acceptable. The following bibliography provides a sample reference list with entries for journal articles [**?**], an LNCS chapter [**?**], a book [**?**], proceedings without editors [**?**], and a homepage [7]. Multiple citations are grouped [**?**,**?**,**?**], [**?**,**?**,**?**,7].

References

1. Hinton, G. E., Osindero, S. and Teh, Y. A fast learning algorithm for deep belief nets. Neural Computation 18, pp 1527-1554.
2. Kak, S. (1995). "On quantum neural computing". Advances in Imaging and Electron Physics. 94: 259–313. doi:10.1016/S1076-5670(08)70147-2. ISBN 9780120147366.
3. Chrisley, R. (1995). "Quantum Learning". In Pylkkänen, P.; Pylkkö, P. (eds.). New directions in cognitive science: Proceedings of the international symposium, Saariselka, 4–9 August 1995, Lapland, Finland. Helsinki: Finnish Association of Artificial Intelligence. pp. 77–89. ISBN 951-22-2645-6.

4. Behrman, E. C., Steck, J. E., Kumar, P., & Walsh, K. A. Quantum Algorithm design using dynamic learning. Quantum Information and Computation. 2008, 8(1&2),12–29.
5. S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in Advances in neural information processing systems, 2015, pp. 1135–1143.
6. S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. A. Horowitz, and W. J. Dally, "Eie: Efficient inference engine on compressed deep neural network," in Proceedings of the 43rd International Symposium on Computer Architecture, ser. ISCA 16. IEEE Press, 2016, p. 243254. [Online]. Available: https://doi.org/10.1109/ISCA.2016.30
7. LNCS Homepage, http://www.springer.com/lncs. Last accessed 4 Oct 2017