Development of a QNN based binary classifier and comparison of its performance on different Classical Processors

Rita Abani[1], Amlan Chakrabarti[2]

[1]Department of Electrical Engineering and Computer Science, Indian Institute of Science Education and Research, Bhopal , India

[2]A. K. Choudhury School of Information Technology, University of Calcutta, Kolkata, India

Extended abstract

In this work, we investigate the feasibility of a binary Quantum Neural Network-based classifier developed using the integration of Cirq and TensorFlow Quantum and tested on a variety of hardware accelerators to see if parallel computing can provide an advantage. Numerous amounts of data have been generated, particularly in the last few years, which has witnessed a growth in the Big-data sector, and research is underway to identify strong and favourable techniques to analyse and understand data in faster, more efficient, and less chaotic ways. Moore's Law's recent developments haven't been encouraging, owing to the curve's progressive saturation, which has prompted us to look into various pathways of unorthodox computing methods that could help alleviate the current growing computing needs of industry and academia. Applied physics, biomedical engineering, chemical simulations, cryptography, and finance are just a few of the fields that require advanced data management and machine learning techniques. Several milestones have been neared, especially concerning Quantum Hardware with advances in the manufacture of qubits, scalability, error, and noise reduction increasingly convincing academia of the purported power of Quantum Computing in addressing the current shortage of computational power and the challenges of computational complexity.TensorFlow Quantum, a quantum machine learning library that enables quick prototyping of hybrid quantum classical ML models, is used in our research. It integrates quantum computation algorithms and logic designed in Cirq, a Python software library for manipulating, writing, and optimising quantum circuits as well as running them on quantum computers and quantum simulators with quantum computation primitives that are compatible with existing TensorFlowAPIs as well as high-performance quantum circuit simulators.

References

Cong, Iris & Choi, Soonwon & Lukin, Mikhail. (2018). Quantum Convolutional Neural Networks. https://arxiv.org/pdf/1802.06002.pdf

Farhi, Edward & Neven, Hartmut. (2018). Classification with Quantum Neural Networks on Near Term Processors.

Aliyev, Vagif. "A definitive explanation to the Hinge Loss for Support Vector Machines." *Towards data science*, Medium, https://towardsdatascience.com/a-definitive-explanation-to-hinge-loss-for-support-vector-machines-ab6d8d3178f1.

Beer, K., Bondarenko, D., Farrelly, T. *et al.* Training deep quantum neural networks. *Nat Commun* 11, 808 (2020). https://doi.org/10.1038/s41467-020-14454-2

Chowdhury, Kunal. "Understanding loss functions : Hinge loss." *Analytics Vidhya*, Medium, https://medium.com/analytics-vidhya/understanding-loss-functions-hinge-loss-a0ff112b40a1.

Laurel, Carlos & Dong, Shihai & Cruz-Irisson, M.. (2015). Equivalence of a Bit Pixel Image to a Quantum Pixel Image. Communications in Theoretical Physics. 64. 501-506. 10.1088/0253-6102/64/5/501. https://www.researchgate.net/publication/284750787_Equivalence_of_a_Bit_Pixel_Image_to_a_Quantum_Pixel_Image

Other websites:

- https://towardsdatascience.com/a-definitive-explanation-to-hinge-loss-for-support-vector-machines-ab6d8d3178f1
- https://serverguy.com/comparison/cpu-vs-gpu-vs-tpu/
- https://www.geeksforgeeks.org/intuition-of-adam-optimizer/