

Hybrid Quantum-Classical Neural Networks for Text Classification

Dhruv Baronia

Abstract — Quantum Computing presents an interesting paradigm where it can possibly offer certain improvements and additions to a classical network while training. This method is particularly prevalent in the current Noisy Intermediate-Scale Quantum era, where we can test these theories using libraries such as PennyLane in conjunction with robust ML frameworks such as TensorFlow. This paper presents a proof-of-concept for the same, using a hybrid quantum-classical model to solve a text classification problem on the IMDB Movie Sentiment Dataset. These hybrid models utilize precalculated embeddings and dense layers alongside a variational quantum circuit layer. We created 4 such models, utilizing various kinds of embeddings, namely NNLM-128, NNLM-50, Swivel and USE, using TFHub and PennyLane. We also trained classical versions of these models, without the variational quantum layer to evaluate the performances. All models were trained on the same data, keeping the batch size and epochs constant.

I. INTRODUCTION

Transfer learning is a typical example of an artificial intelligence technique that has been originally inspired by biological intelligence. It originates from the simple observation that the knowledge acquired in a specific context can be transferred to a different area [1]. In the context of Natural Language Processing, this is most prevalent in the form of precalculated word embeddings, wherein textual representations obtained networks trained on copious amounts of unlabeled data for a generalized task are made available, so models for specific tasks can be built using them with a little fine-tuning. This is also referred to as Sequential Transfer Learning. [2]. In this paper, we explore the possibility of combining this technique with quantum computing. Quantum Computing utilizes qubits rather than classical binary state bits. These qubits can represent both 0 and 1 states due to superposition and attain a particular value when they are observed. The most common way they are represented is in the form of Bloch spheres, where each operation on a qubit is represented as a rotation of the quantum state vector along the three axes of the sphere. This representation leads to highly specific algorithms and circuits being formed, which show a significant increase in speed over calculations using classical bits due to the superposition of qubits. This is known as super-polynomial speedup. The application of this concept in traditional computing is something that has been speculated heavily, particularly in computationally heavy tasks such as machine learning. While notable advancements have been made in combining the two [3-7], the speedups observed have been few and far between and always been on quantum hardware. Right now, quantum computing is in the NISQ (Noisy Intermediate Scale Quantum) era. This means devices are available, but not powerful enough to outcompete classical

methods (especially on tasks that they have been highly tuned for, e.g., image processing). One of the biggest positives of transfer learning is its democratization, and the opportunity for all ML practitioners, from hobbyists to researchers to utilize models pretrained on a huge amount of data and suit it to their needs. Quantum Computing and Quantum Machine Learning especially are quite a ways off from achieving the popularity and accessibility levels of models such as ResNet and BERT, but software libraries such as PennyLane and TensorFlow Quantum are making huge strides in this direction, making simulations of qubits and quantum algorithms possible on classical computers [8-9]. Increasingly difficult simulations, such as those in QML are difficult to execute quickly, but they are good predictors of actual qubit behavior on quantum hardware.

II. METHODS

I tested the hybrid models with 4 different embeddings taken from TFHub. The first two are NNLM-128 and NNLM-50, which are 128-dimensional and 50-dimensional vectors, respectively. The NNLM model produces token-based embeddings [10]. Swivel is a much smaller model, with a 20-dimensional output vector, with the individual embeddings being combined into sentence embeddings [11]. Finally, we have the USE model, with 512-dimension vectors as the output, as it takes greater than word length text for calculations [12]. The classical models were simple sequential models, with the embeddings layer followed by a fully connected dense layer of 16 units, followed by another dense layer of 4 units, and an output layer of 1 unit. We trained the models for 10 epochs with a batch size of 512.

S. No.	Classical Model	Train Acc.	Train Loss	Test Acc.	Test Loss
1	Swivel	87.5	0.297	82.6	0.374
2	NNLM-50	99.8	0.022	83.8	0.432
3	NNLM-128	99.7	0.006	85.3	0.463
4	USE	84.0	0.335	83.7	0.380

For the hybrid models, we added a variational quantum layer after the dense layer with 4 units. This essentially acts as a fully-connected dense layer, but since it cannot do explicit matrix multiplication due to the nature of the qubits, the variational layer is actually a dressed quantum layer composed of 3 parts. The first is an AngleEmbedding layer, which encodes N features into the rotation angles of n qubits, where $N \leq n$. The next layer is a StroglyEntangledLayers layer, which contains a quantum circuit of parametrised single and two-qubit gates together with a single-qubit measurement

is used to classify the inputs. The output of this is then converted back into a classical output for the final output layer of our sequential model by passing through a dense layer in the dressed quantum circuit with softmax activation. The entire circuit is wrapped in a QNode, which builds the circuit and provides the qubits for simulation. We use 3 layers for all the hybrid models, and train them for 10 epochs.

S. No	Hybrid Model	Train Acc.	Train Loss	Test Acc.	Test Loss
1	Swivel	86.7	0.352	80.9	0.417
2	NNLM-50	99.6	0.270	84.4	0.430
3	NNLM-128	99.8	0.004	84.2	0.375
4	USE	88.0	0.280	85.8	0.320

We can observe that the results are comparable to the classical models, and the models with larger embedding vectors (NNLM-128 and USE) show improvements in the form of lowered loss values as well as increase in accuracies for USE.

III. RESULTS

After training all the models, the Quantum-Hybrid model with USE embeddings was the only one that showed a noticeable improvement in the final accuracy over its classical counterpart with 85.8%, a 2.1% increase. Even though the other models did not show an increase in the accuracy, they lowered the loss in many cases. The quantum models took much longer (about 90 minutes per epoch for a batch size of 512) to train using the PennyLane simulator. However, this is to be expected, as efficient quantum simulations are not widely available yet. As the number of qubits increase, the operations on the same become increasingly difficult to carry out. However, seeing that the end results of the hybrid models are so comparable to those of the classical models, it is evident that if hardware limitations are overcome, they can show tangible increases in accuracy, and maybe even open the door to fully quantum neural networks. For now, transfer learning for a downstream task is the most effective way of achieving quantum speedup in a hybrid model.

IV. DISCUSSION

Limitations of current classical computers forbid us from trying out circuits with more number of qubits or training the hybrid models for more epochs, especially when considering those with large embedding vectors. The NISQ era promises quantum speedup for around 50-200 qubits or so, but the fact is that quantum computing is still at such an early stage that integrating it with other areas of computer science such as ML is not efficient on classical hardware. Quantum simulation libraries are paving the way for the democratization of quantum computing, but quantum hardware is still required to

run highly complex algorithms and take advantage of the super-polynomial speedup promised by quantum computing.

ACKNOWLEDGMENT

I would like to thank Josh Izaac from Xanadu Quantum Technologies for his help throughout the project and suggesting methods to improve the results. I would also like to thank my Computer Science teacher, Dr. Ruchika Thukral for aiding me in the research process.

REFERENCES

- [1] Andrea Mari, Thomas R. Bromley, Josh Izaac, Maria Schuld, and Nathan Killoran. *Transfer learning in hybrid classical-quantum neural networks*. arXiv:1912.08278 (2019).
- [2] Sebastian Ruder, Matthew E. Peters, Swabha Swayamdipta, and Thomas Wolf. Transfer learning in natural language processing. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials, pages 15–18, (2019).
- [3] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023023, (2016).
- [4] Alejandro Perdomo-Ortiz, Marcello Benedetti, John Realpe-Gómez, and Rupak Biswas. Opportunities and challenges for quantum-assisted machine learning in near-term quantum computers. *Quantum Science and Technology*, 3(3):030502, (2018).
- [5] Sukin Sim, Peter D Johnson, and Alan Aspuru-Guzik. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. arXiv preprint arXiv:1905.10876, (2019).
- [6] Nathan Killoran, Thomas R Bromley, Juan Miguel Arrazola, Maria Schuld, Nicolás Quesada, and Seth Lloyd. Continuous-variable quantum neural networks. arXiv preprint arXiv:1806.06871, 2018.
- [7] John Preskill. Quantum computing in the NISQ era and beyond. *Quantum*, 2:79, (2018).
- [8] Bergholm, Ville, et al. "PennyLane: Automatic differentiation of hybrid quantum-classical computations." arXiv preprint arXiv:1811.04968 (2018).
- [9] Broughton, Michael, et al. "Tensorflow quantum: A software framework for quantum machine learning." arXiv preprint arXiv:2003.02989 (2020).
- [10] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, Christian Jauvin. A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 3:1137-1155, (2003).
- [11] Shazeer, Noam, et al. "Swivel: Improving embeddings by noticing what's missing." arXiv preprint arXiv:1602.02215 (2016).
- [12] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Céspedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, Ray Kurzweil. Universal Sentence Encoder. arXiv:1803.11175, (2018).