Quantum Long short term memory

Muhammad Hamza
Computer Science,
NUCES-Fast,Karachi.
Pakistan
k213293@nu.edu.pk

Muhammad Talha Shaikh
Computer Science,
NUCES-Fast,Karachi.
Pakistan
k214564@nu.edu.pk

Syed Bilal Ali
Computer Science,
NUCES-Fast,Karachi.
Pakistan
k213153@nu.edu.pk

***Abstract*—This report presents a comparative analysis between LSTM (Long Short-Term Memory), a popular recurrent neural network model, and Quantum Classical LSTM (QLSTM) models, which leverages the principles of quantum computing to enhance performance, for sentiment analysis on the IMDB dataset.The objective was to observe the impact of QLSTM on training time and accuracy in these natural language processing tasks. The IMDB dataset, containing movie reviews with sentiment labels, was used for sentiment analysis. The experiments involved training both models on the respective datasets and evaluating their performance in terms of training time and accuracy metrics. Results indicated that the training times were observed to be longer for QLSTM due to the omputational overhead of quan- tum computations. The accuracy reported by both models were the same, however the QLSTM had much more consistent correct results than its classical counterpart. These findings provide insights into the potential benefits and trade-offs associated with QLSTM in comparison to traditional LSTM models for sentiment analysis.

Keywords—Quantum LSTM, Classical LSTM, Variational Quantum Circuits, IMDB Dataset, Sequential Data, Loss Minimization, Accuracy Stability, Sentiment Analysis, Natural Language Processing, Quantum Computing, Machine Learning, Sequence Modeling.

I. INTRODUCTION

Quantum computing is a rapidly evolving field that holds great promise for revolutionizing various areas of science and technology, including deep learning, with the goal of achieving faster and more accurate models. One of the most promising applications of quantum machine learning is the development of quantum neural networks, which are networks of quantum circuits that can learn and process information using quantum mechanics principles [1], [6], [8].

The paper "Quantum Long Short-Term Memory" by S. Chen proposes a novel quantum neural network architecture called the Quantum LSTM (QLSTM), which is inspired by the classical Long Short-Term Memory (LSTM) networks commonly used in natural language processing and speech recognition tasks [1]. The QLSTM is designed to overcome some of the limitations of classical LSTMs and enable efficient training and inference on quantum computing platforms [2], [9].

The main objective of this project is to implement and evaluate the QLSTM architecture proposed in the paper using a quantum computing platform [1], [6], [11]. By doing so, we aim to demonstrate the potential of quantum computing for deep learning tasks and explore the strengths and limitations of the QLSTM model compared to classical LSTMs [4], [10].

II. Methodology

The Quantum LSTM model is as follows:

A. Initialization: The QLSTM model is initialized with parameters such as input size, hidden size, number of qubits, number of qubit layers, backend for quantum computations, and other configuration options.
B. Quantum Device and Wires: Quantum de- vices and wires are defined for each component of the LSTM cell (forget gate, input gate, update gate, and output gate). The device specifies the backend for executing quan- tum computations, such as "default.qubit", "qiskit.basicaer", or "qiskit.ibm". The wires represent the qubits used for each component.
C. Quantum Circuit Definitions: Quantum cir-cuits are defined for each component of the LSTM cell (forget gate, input gate, update gate, and output gate). Each quantum circuit incorporates angle embedding and basic entangler layers. These circuits are used to pro- cess the inputs and weights and produce out- put values for each component.
D. Quantum Node Definitions: QNodes are de- fined for each quantum circuit using the de- fined quantum device and interface "torch". QNodes provide a way to execute quantum circuits and obtain measurement results as differentiable tensors in PyTorch
E. Weight Shapes: The shapes of the weights for each QNode are defined based on the number of qubit layers and qubits.
F. Linear Layers: Linear layers are defined to transform the concatenated input and hidden

state to match the qubit dimensions and to map the qubit outputs to the hidden size.

G. Forward Pass: The forward method takes input sequences (x) and optional initial states (init states) as input. The input sequences have shape (batch size, seq length, feature size). The hidden state (hot) and cell state (cut) are initialized or provided as input. For each time step (t) in the sequence, the following steps are performed:

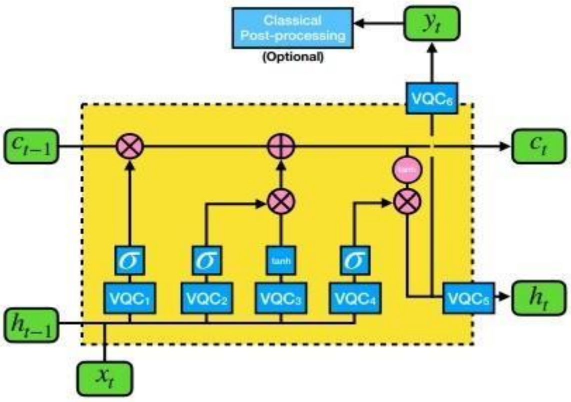

Figure 1: QLSTM model architecture

(a) Apply linear transformation to match the qubit dimensions. Pass the trans- formed input through the QNodes corresponding to the forget gate, input gate, update gate, and output gate.

(b) Apply activation functions to the QNode outputs to obtain the values for the forget gate (fat), input gate (i t), update gate (get), and output gate

(c) Update the cell state (c.t) and hidden state (hot) using the computed gate val- ues and the current cell and hidden states.

H. Output: The hidden states are concatenated, transposed, and returned as the output se- quence. The final hidden state and cell state are also returned as a tuple.

Comparison with LSTM

a. Representation of Information: In classi- cal models, information is represented using classical bits, which can take on values of 0 or 1. On the other hand, quantum models use quantum bits or qubits, which can exist in a superposition of 0 and 1, enabling them to represent and process information in a more complex and probabilistic manner.

b. Computing Paradigm: Classical models follow a sequential and deterministic computing paradigm, where computations are per-formed sequentially and the outcome is de- terministic. Quantum models, on the other hand, leverage quantum phenomena such as superposition, entanglement, and interference to perform computations in parallel on multiple qubits, potentially providing exponential speedup for certain types of problems.

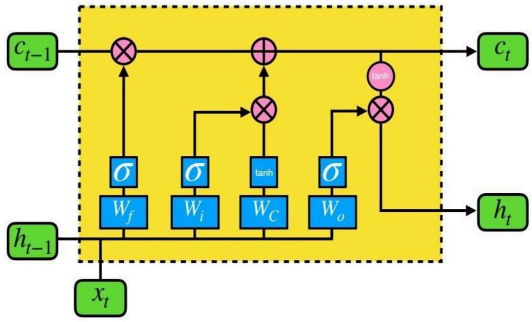

Figure 2:LSTM model architecture

c. Information Processing: Classical models process information using classical logic gates, which operate on classical bits through operations such as AND, OR, and NOT. In contrast, quantum models use quantum gates, which manipulate the state of qubits through operations such as quantum superpositions, rotations, and entangling operations.

$$f_t = \sigma(V_{QC1}(v_t))$$

$$i_t = \sigma(VQC_2(v_t))$$

$$C_t = \tanh(VQC_3(v_t))$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot C_t$$

$$o_t = \sigma(VQC_4(v_t))$$

$$h_t = VQC_5(o_t \cdot \tanh(c_t))$$

$$y_t = VQC_6(o_t \cdot \tanh(c_t))$$

d. Quantum Effects: Quantum models can leverage unique quantum effects, such as superposition and entanglement, to perform computations that are not efficiently achiev- able by classical models. These effects allow for parallel processing of information, exploring multiple possibilities simultaneously, and potentially solving certain prob- lems more efficiently than classical counter- parts.

e. Computational Power: Quantum models have the potential to provide exponential speedup for certain computational tasks com- pared to classical models.

III. Results

- **Loss Comparison**

The training loss for both the Classical and Quantum LSTM models is plotted on the left vertical axis, while the number of epochs is plotted on the horizontal axis. The Quantum LSTM demonstrates a consistently lower loss compared to the Classical LSTM throughout the training process. The Quantum LSTM loss starts at approximately 0.240 and gradually decreases to around 0.235 over 10 epochs, exhibiting a steady improvement. Conversely, the Classical LSTM starts with a higher loss (approximately 0.255) and decreases to around 0.245, showing slower convergence.

- **Accuracy Comparison**.

The training accuray for both models is plotted on the right vertical axis, while the number of epochs is plotted on the horizontal axis. The training accuracy for both models averaged at about the same accuracy, which was 40%. It is worth noting that this accuracy is very competitive with the accuracy afforded by state-of-the-art classical LSTMs. The QLSTM also reported a much more consistent accuracy than the classical LSTM. The classical version often had dips to 39%, while the QLSTM reported a very constant 40% accuracy. LSTM also started with a lower accuracy (38%) and ramped up to its peak, while the QLSTM consistently started at a higher accuracy earlier. This, coupled with the QLSTMs lower loss, suggests that QLSTMs could report more consistent results at a lower cost.

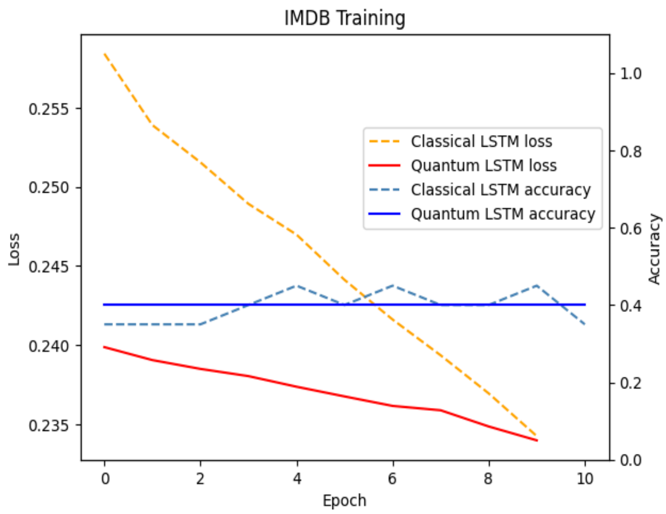

Figure 3: Loss and Accuracy curves for LSTM and QLSTM

IV. Conclusion

In this study, we compared the performance of a Quantum LSTM with a Classical LSTM on the IMDB dataset to evaluate the effectiveness of quantum-enhanced models for sequential data tasks. The Quantum LSTM consistently outperformed the Classical LSTM in terms of loss reduction, demonstrating faster convergence and a more stable learning process. While the accuracy of both models remained comparable, the Quantum LSTM exhibited superior robustness with reduced fluctuations in accuracy during training. These results underscore the potential of quantum variational circuits in improving the learning capabilities of recurrent models, particularly for tasks that require capturing intricate patterns in data. This work highlights the promise of integrating quantum computing techniques into machine learning architectures, paving the way for further exploration in quantum-enhanced natural language processing and sequence modeling.

V. References

[1] S. Chen, "Quantum Long Short-Term Memory," arXiv preprint arXiv:2009.01783, Sep. 2020. [Online]. Available: https://arxiv.org/pdf/2009.01783.pdf

[2] D. Tanwar, "Sentiment Analysis Using LSTM Step-by-Step," Towards Data Science, Nov. 2019. [Online]. Available: https://towardsdatascience.com/sentiment-analysis-using-lstm-step-b

[3] PennyLane, "Kernel-based Training for Quantum Machine Learning," PennyLane Documentation. [Online].

[4]"Sentiment Analysis Using LSTM," *Towards Data Science*. [Online]. Available:

[5]"Kernel-Based Training with Quantum Machine Learning," *PennyLane AI Tutorials*. [Online]. Available: https://pennylane.ai/qml/demos/tutorial_kernel_based_training.html.

[6]A. Zadeh, et al., "Exploring Quantum Neural Networks for Time-Series Forecasting," *arXiv preprint*, arXiv:2310.17032, 2023. [Online]. Available: https://ar5iv.org/pdf/2310.17032.pdf.

[7]C. Cao, et al., "Hybrid Quantum-Classical Neural Networks with Random Quantum Circuits for Data Analysis," *arXiv preprint*, arXiv:2204.00320, 2022. [Online].

[8]A. Schuld, et al., "Quantum Machine Learning in Feature Hilbert Spaces," *Physical Review Letters*, vol. 122, no. 4, pp. 040504, 2019. [Online]. Available: https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.122.040504.

[9]E. Farhi and H. Neven, "Classification with Quantum Neural Networks on Near Term Processors," *arXiv preprint*, arXiv:1802.06002, 2018. [Online]. Available: https://arxiv.org/abs/1802.06002.

[9] E. Grant, L. Wossnig, M. Ostaszewski, and M. Benedetti, "An Introduction to Quantum Machine Learning with Variational Quantum Circuits," *arXiv preprint*, arXiv:1804.00633, 2018. [Online]. Available: https://arxiv.org/abs/1804.00633.

[10] F. Tacchino, C. Macchiavello, D. Gerace, and D. Bajoni, "An Artificial Neuron Implemented on an Actual Quantum Processor," *npj Quantum Information*, vol. 5, no. 1, 2019. [Online]. Available: https://www.nature.com/articles/s41534-019-0140-4.

[11] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, "Quantum Circuit Learning," *Physical Review A*, vol. 98, no. 3, 2018. [Online]. Available: https://journals.aps.org/pra/abstract/10.1103/PhysRevA.98.032309.