

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 and part-of- speech (POS) tagging tasks. 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, while a self-made POS tagging dataset was employed for the POS tagging task. 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 computational overhead of quantum computations. In POS tagging, both models demonstrated comparable accuracies, with LSTM exhibiting faster training times. These findings provide insights into the potential benefits and trade-offs associated with QLSTM in comparison to traditional LSTM models for sentiment analysis and POS tagging tasks.

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.

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. The QLSTM is designed to overcome some of the limitations of classical LSTMs and enable efficient training and inference on quantum computing platforms.

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

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 devices 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 quantum computations, such as "default.qubit", "qiskit.basicaer", or "qiskit.ibm". The wires represent the qubits used for each component.
- C. Quantum Circuit Definitions: Quantum circuits 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 process the inputs and weights and produce output values for each component.
- D. Quantum Node Definitions: QNodes are defined for each quantum circuit using the defined 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:

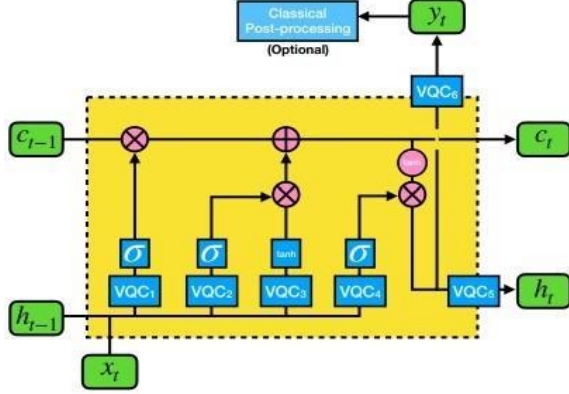


Figure 1: QLSTM model architecture

- Apply linear transformation to match the qubit dimensions. Pass the transformed input through the QNodes corresponding to the forget gate, input gate, update gate, and output gate.
 - Apply activation functions to the QNode outputs to obtain the values for the forget gate (f_t), input gate (i_t), update gate (g_t), and output gate (o_t).
 - Update the cell state (c_t) and hidden state (h_t) using the computed gate values and the current cell and hidden states.
- H. Output: The hidden states are concatenated, transposed, and returned as the output sequence. The final hidden state and cell state are also returned as a tuple.

Comparison with LSTM

- Representation of Information:** In classical 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.
- Computing Paradigm:** Classical models

follow a sequential and deterministic computing paradigm, where computations are performed sequentially and the outcome is deterministic. 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.

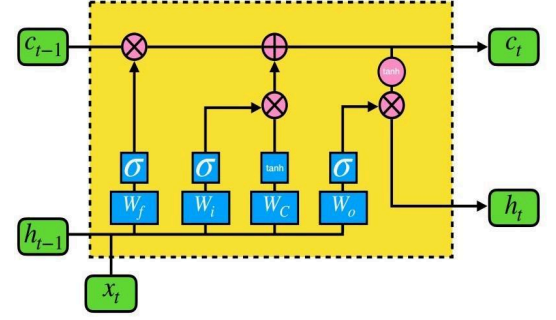


Figure 2: LSTM model architecture

- 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.

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

$$i_t = \sigma(V_{QC2}(v_t))$$

$$C_t = \tanh(V_{QC3}(v_t))$$

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

$$o_t = \sigma(V_{QC4}(v_t))$$

$$h_t = V_{QC5}(o_t \cdot \tanh(c_t))$$

$$y_t = V_{QC6}(o_t \cdot \tanh(c_t))$$

- Quantum Effects:** Quantum models can leverage unique quantum effects, such as superposition and entanglement, to perform computations that are not efficiently achievable by classical

models. These effects allow for parallel processing of information, exploring multiple possibilities simultaneously, and potentially solving certain problems more efficiently than classical counterparts.

- e. **Computational Power:** Quantum models have the potential to provide exponential speedup for certain computational tasks compared 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 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.

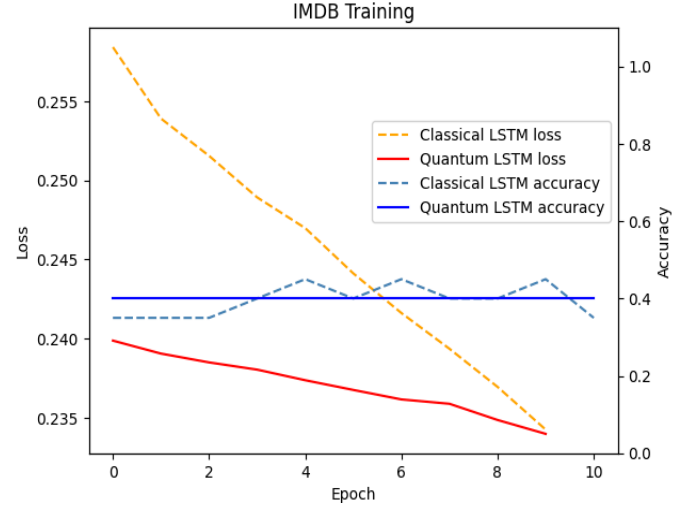


Figure 3: Loss and Accuracy curves for LSTM and QLSTM

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. <https://arxiv.org/pdf/2009.01783.pdf>
2. <https://github.com/rdisipio/glstm/tree/main#readme>
3. <https://towardsdatascience.com/sentiment-analysis-using-lstm-step-by-step/>
4. https://pennylane.ai/qml/demos/tutorial_kernel_based_training.html