FIRE: THE FIRST-YEAR INNOVATION & RESEARCH EXPERIENCE

QUANTUM MACHINE LEARNING

QUANTUM NATURAL LANGUAGE PROCESSING OPTIMIZATION

Evren Yücekuş~Kissane, Neil Kakhandiki, Ivan Wang, Anish Dhanrajani

https://tinyurl.com/fireqnlp

Introduction

The paper "QNLP in Practice: Running Compositional Models of Meaning on a Quantum Computer" by Coecke et al. presents the implementation of NLP models on quantum hardware. It explores how sentences, represented as tensor networks in the DisCoCat model, are mapped to quantum circuits. Simulations were conducted to compare different NLP models. This involved assessing models that vary in their syntactic sensitivity and in the circuit parameterization to understand how quantum processing might impact NLP tasks differently compared to classical approaches.

The paper proposes a fully syntax-sensitive based model according to a pre-group grammar, which is essentially a set of rules describing a language. This diagrammatic sentence representation is then converted into a quantum circuit based on an ansatz.

IQP ansatz description:
(1) Euler parameterization (3 params): $Rx(\theta_3)Rz(\theta_2)Rx(\theta_1)|0\rangle$
(2) Single Rx gate (1 param): $Rx(\theta)|0\rangle$

Depending on the hardware, one of the two options is chosen to represent all words in a sentence. For adjectives, verbs, etc. that are represented by multiple qubits, layers of instantaneous quantum polynomial (IQP) gates are used (consisting of a Hadamard gate applied to each qubit, and a series of controlled Rz gates). There is a special case where the word "that" is represented by a GHZ state.

The paper subsequently trains these sentences using a Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm to minimize the cost which is computationally less expensive than back propagation.

Ansatz Improvements

The motivation behind specific ansatzes are dictated by the DisCoCat model developed by Coecke et al. We propose two more ansatzes that have proven to be as successful as the IQP layers. The Sim14.1 and Sim15.1 ansatzes are as follows.

- **Sim14.1:** For an n qubit circuit, each layer of Sim14.1 consists of two "sublayers". Each sublayer consists of n RZ gates, followed by n CNOT gates, arranged in a ring topology. In the second sublayer, the ring of CNOT gates iterates in reverse.
- **Sim15.1:** Same layout as Sim14.1, but replaces all CNOT gates with CRZ gates. An additional layer of Hadamard gates is added inspired by the IQP ansatz.

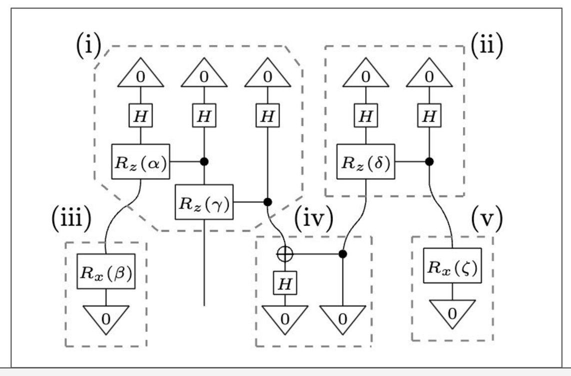


Fig 1. The quantum circuit for the sentence "person prepares tasty dinner" based on the IQP ansatz. Note that group (iv) has no parameter.

Training Algorithmic Improvements

To improve the SPSA algorithm, we introduce learning rates that adjust over the iterations that allows the parameters to substantially converge faster. Additionally, we implement a version of the ADAM optimizer that has adjustable learning rates while maintaining a low compute time. We finally implement a genetic algorithm which could be efficient when significantly large quantum computers are realized. The loss function is a standard cross entropy function:

$$C(\Theta) := -\frac{1}{|\mathcal{T}|}\sum_{P\in\mathcal{T}} L(P)^T \cdot \log(l_\Theta(P))$$

Experiments

We tested our algorithms and ansatz choice on a binary classification task on the MC dataset provided by Coecke et al. The dataset includes 130 sentences generated by a pregroup grammar that are either classified as food or IT. The results section shows the plots of the various algorithms used with different ansatz choices. The number of qubits used to represent the noun and sentence types based on the DisCoCat model are both one due to current limitations in quantum hardware. We achieve an approximate 20% increase in training accuracy and an impressive 30% increase in testing accuracy when comparing all algorithms across the board compared to the standard SPSA.

Future Work

- Expand the grammar and DisCoCat model to represent more complex sentences
- Implement adaptable ansatzes
- Test implementation on Aer simulator to mimic noise or on real quantum hardware
- Use pre trained quantum word embeddings for faster computation
- Use more efficient and computationally less expensive learning algorithms that converge faster other than SPSA and ADAM

References

View of QNLP in Practice: Running Compositional Models of Meaning on a Quantum Computer. (2024). Jair.org. https://jair.org/index.php/jair/article/view/14329/26923

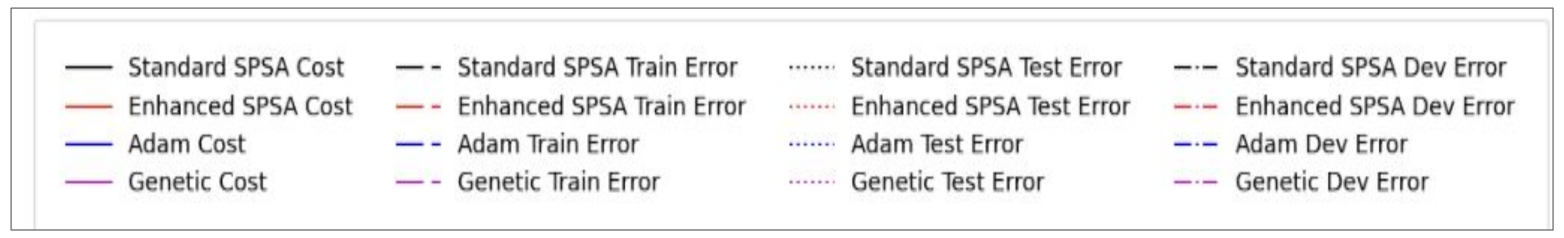
Khatri, N., & Hall, L. (n.d.). Experimental Comparison of Ansätze for Quantum Natural Language Processing. https://www.cs.ox.ac.uk/people/aleks.kissinger/theses/khatri-thesis.pdf

SPSA (v0.39) | IBM Quantum Documentation. (2017). IBM Quantum Documentation. https://docs.quantum.ibm.com/api/qiskit/0.39/qiskit.algorithms.optimizers.SPSA

Spall, J. C. (2000). Adaptive stochastic approximation by the simultaneous perturbation method. Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No.98CH36171), 4, 3872–3879. https://doi.org/10.1109/cdc.1998.761833

PennyLane. (n.d.). Qml.AdamOptimizer. qml.AdamOptimizer - PennyLane 0.39.0 documentation. https://docs.pennylane.ai/en/stable/code/api/pennylane.AdamOptimizer.html

Results



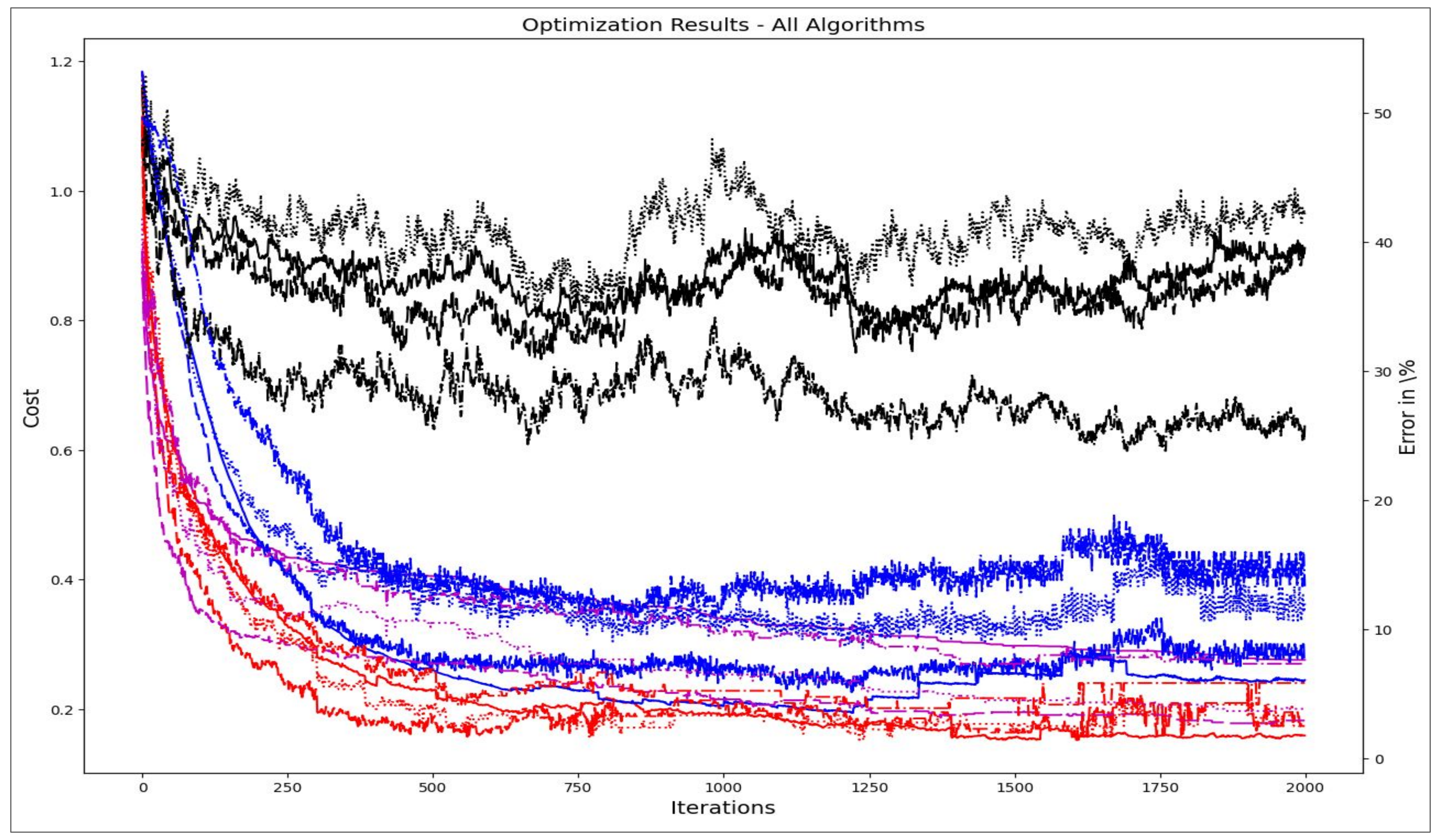


Fig 2. Displays the development, train, and test error for the standard SPSA, improved SPSA, ADAM optimizer, and Genetic algorithms with runs = 20 and iterations = 2000. The IQP ansatz is used.

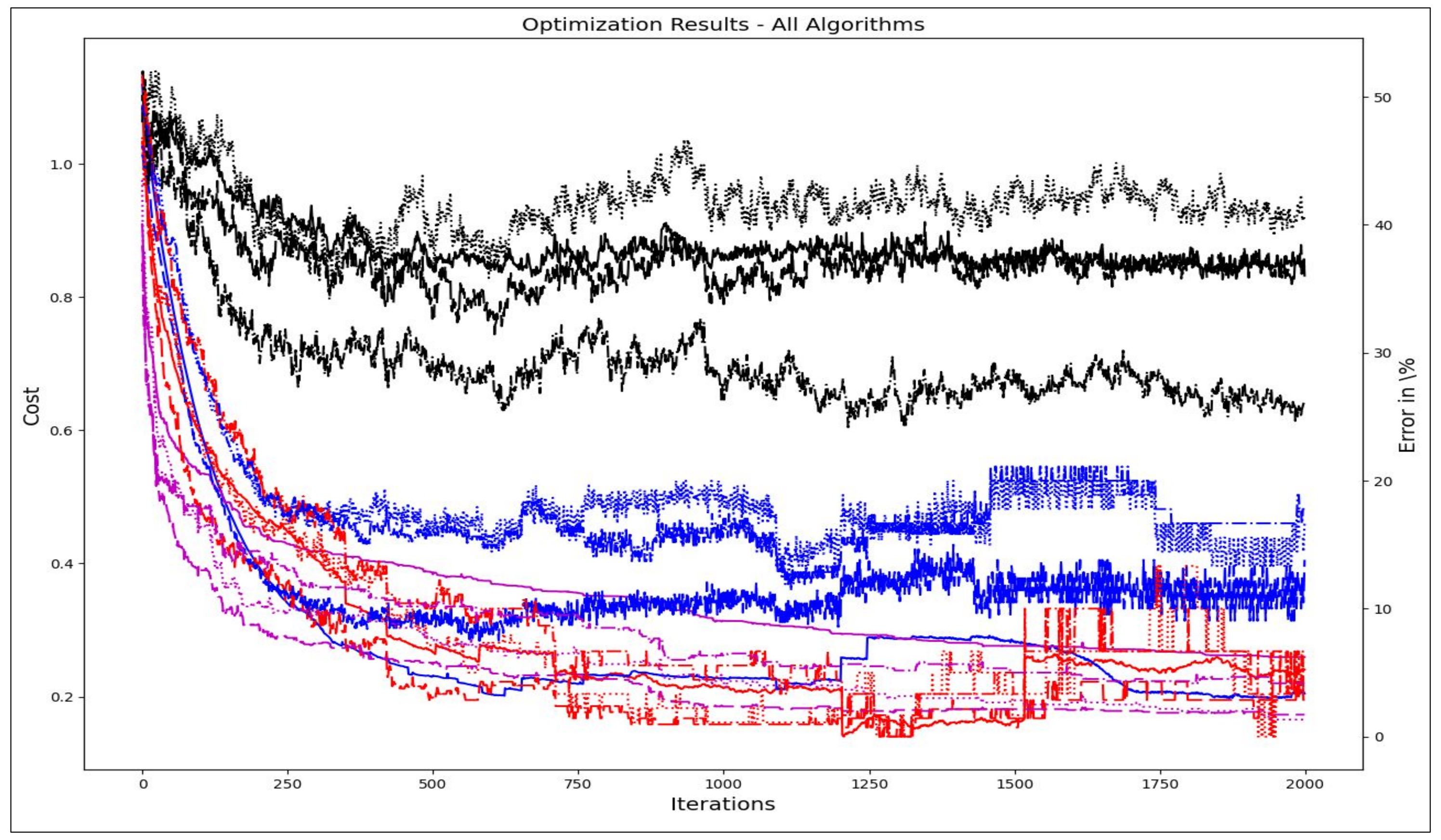


Fig 3. Displays the development, train, and test error for the standard SPSA, improved SPSA, ADAM optimizer, and Genetic algorithms with runs = 20 and iterations = 2000. The Sim14.1 ansatzes is used.

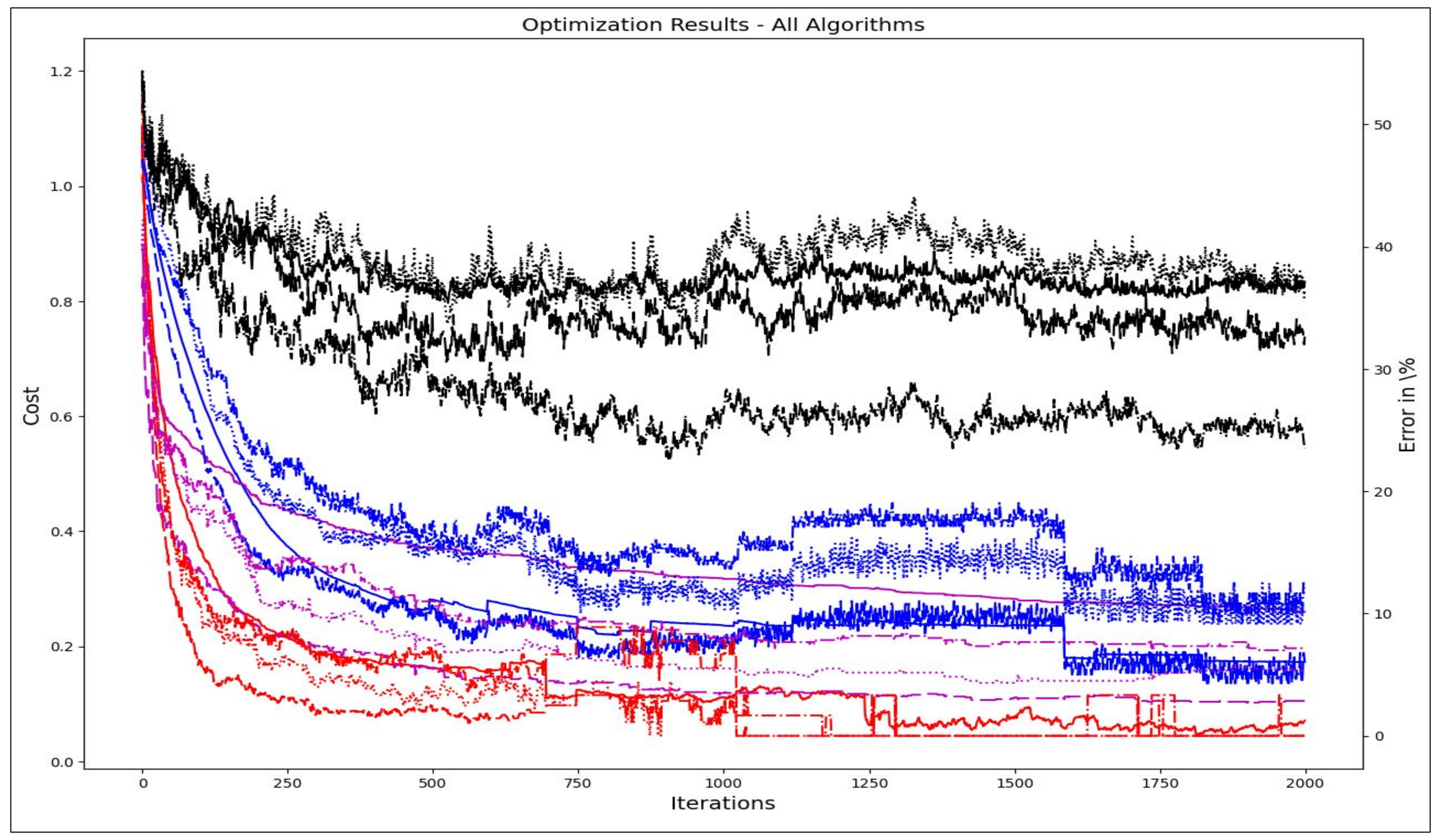


Fig 4. Displays the development, train, and test error for the standard SPSA, improved SPSA, ADAM optimizer, and Genetic algorithms with runs = 20 and iterations = 2000. The Sim15.1 ansatzes is used.