Sentiment Analysis Using Quantum Neural Networks

Owen Bardeen, Thomas Lu, Nathan Song,
Tvisha Londhe, Richard Wang, Andris Huang

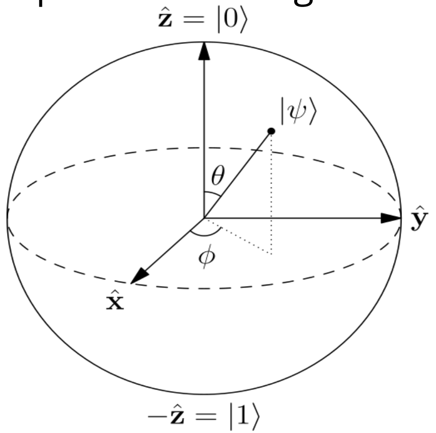
Background

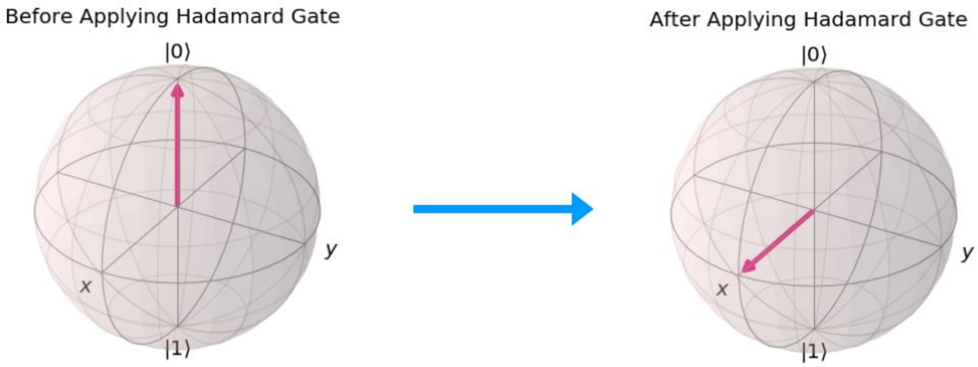
Intro: Qubits

Qubit: Analogous to a classical bit, but instead of binary states have superposition properties where quantum state can be a linear combination of 0 and 1.

- Represented using 2D state vectors → Bloch Sphere

$$|\psi\rangle = \cos\frac{\theta}{2} \cdot |\mathbf{0}\rangle + e^{j\varphi} \sin\frac{\theta}{2} \cdot |\mathbf{1}\rangle$$

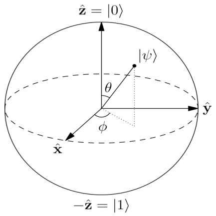




Quantum Gate: Converts a qubit from one quantum state to another

- Single-qubit gates (Hadamard gate: to create a state of superposition in qubit)
- Multi-qubit gates (CNOT gate: to create quantum entanglement)

Intro: Quantum Algorithms



$$|\psi\rangle = \cos\frac{\theta}{2}\cdot|\mathbf{0}\rangle + e^{j\varphi}\sin\frac{\theta}{2}\cdot|\mathbf{1}\rangle$$

$$\hat{H}\psi = E\psi$$

$$H = \frac{p^2}{2m} + V(r) \qquad H = \sum_i w_i Z_i + \sum_{i<j} w_{ij} Z_i Z_j. \qquad \hat{H}_{elec}(r,R) = -\frac{\hbar^2}{2m}\nabla^2 - \frac{e^2}{4\pi\epsilon_0 r_A} - \frac{e^2}{4\pi\epsilon_0 r_B} + \frac{e^2}{4\pi\epsilon_0 R}$$

Classical **_Ising (MaxCut)_** *Electronic (H2)*

Goal: Minimize cost function by...

Finding ground-state energy!
(Or a good approximation)

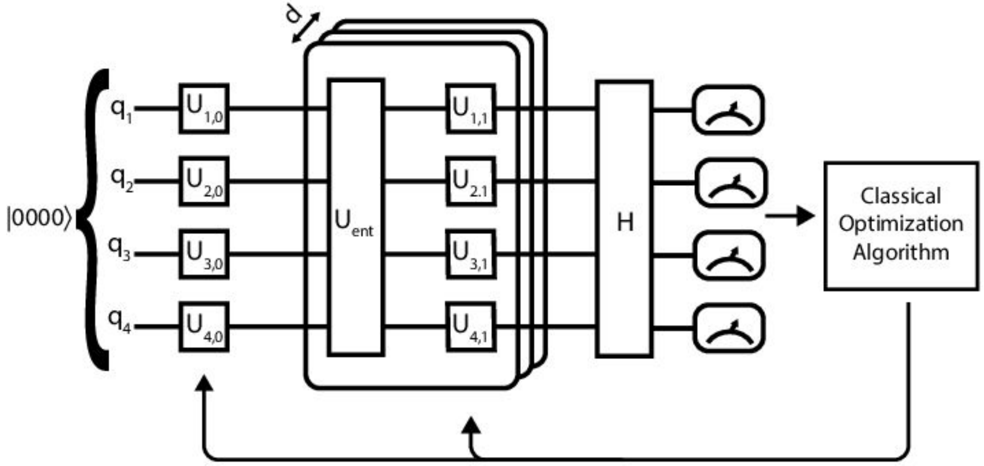
$$E(\theta_1,\cdots,\theta_n) = \langle\hat{H}\rangle = \sum_i \alpha_i \langle\psi(\theta_1,\cdots,\theta_N)|\hat{P}_i|\psi(\theta_1,\cdots,\theta_N)\rangle$$

Intro: Quantum Algorithms

$$E(\theta_1, \cdots, \theta_n) = \langle \hat{H} \rangle = \sum_i \alpha_i \langle \psi(\theta_1, \cdots, \theta_N) | \hat{P}_i | \psi(\theta_1, \cdots, \theta_N) \rangle$$

$$\hat{H}\psi = E\psi$$

(VQE) Variational Quantum Eigensolver

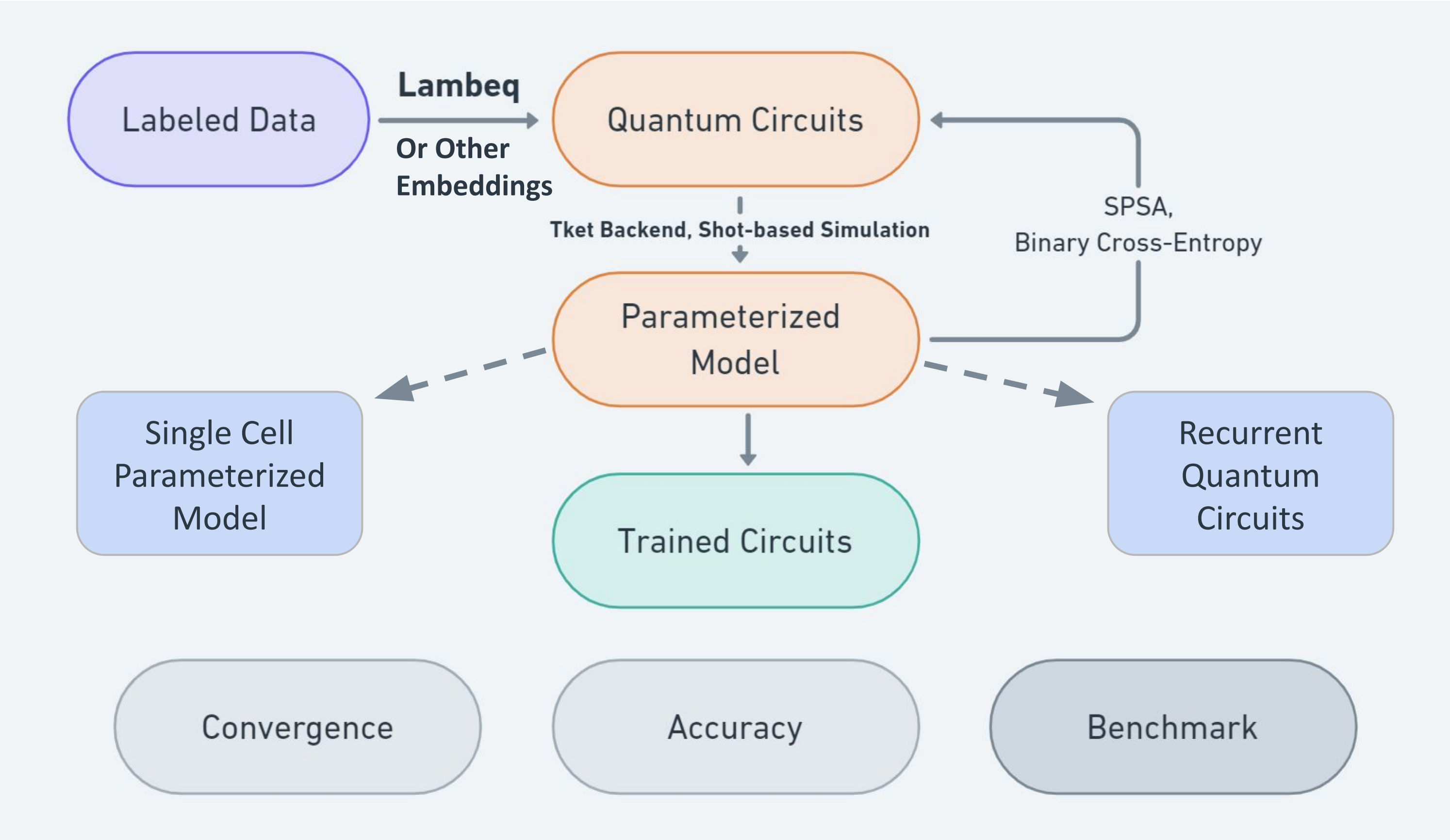


- Enter initial wavefunction
- Run initial circuit
- **Measure** average energy
- Reparameterize circuit to minimize energy
- Repeat until convergence

(QAOA) Quantum Approximate Optimization Algorithm

(SPSA) Simultaneous Perturbation Stochastic Approximation

Model

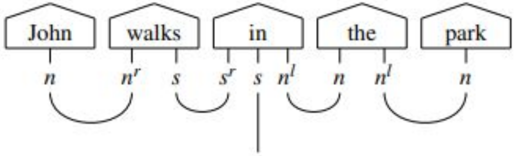

Labeled Data
Lambeq
Or Other Embeddings
Quantum Circuits
Tket Backend, Shot-based Simulation
SPSA,
Binary Cross-Entropy
Parameterized Model
Single Cell Parameterized Model
Trained Circuits
Recurrent Quantum Circuits
Convergence
Accuracy
Benchmark

Lambeq: Overview

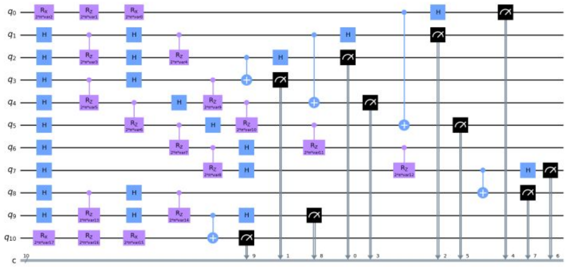
- Python library used for quantum NLP.
- High level: Converts input sentences into quantum circuits. Tunable structures to parameterize nouns, verbs, etc.

Input sentence → String diagram → Quantum circuit

"John walks in the park." →

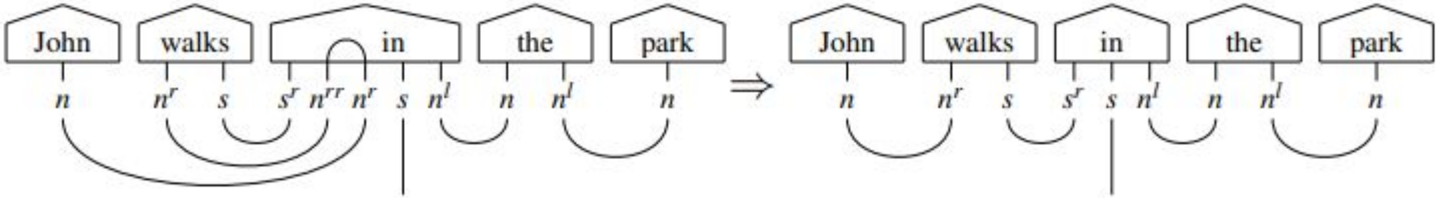


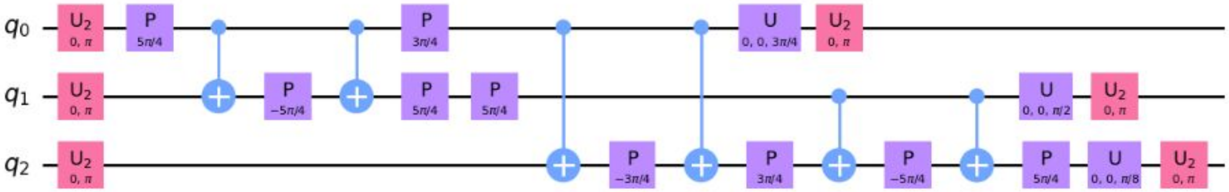
→



```
from lambeq.ccg2discocat import DepCCGParser
from lambeq.rewrite import Rewriter

# Parse the sentence
diagram = depccg_parser.sentence2diagram(’John walks in the park’)

# Apply rewrite rule for prepositional phrases
rewriter = Rewriter([’prepositional_phrase’])
rewritten_diagram = rewriter(diagram)

rewritten_diagram.draw()
```

Rewriting: By simplifying the string diagram representing a sentence, we can reduce the number of qubits that it takes to codify a sentence





SPSA Optimizer:

Traditional Gradient Descent:

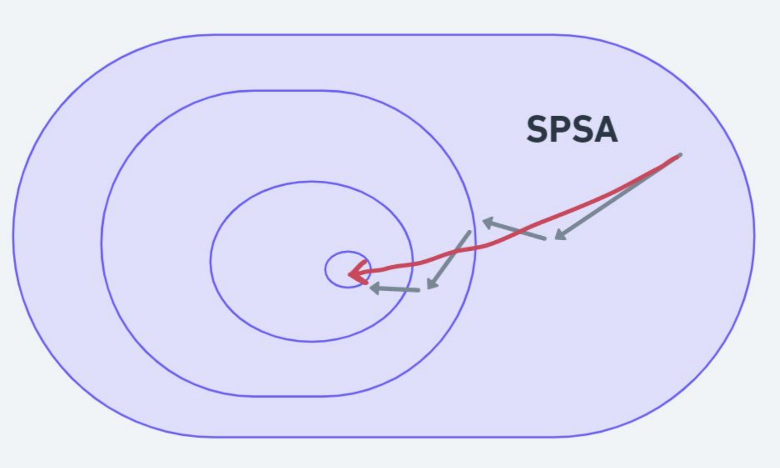
$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \eta \nabla f(\boldsymbol{\theta}^{(k)})$$ O(d), d => # Dimensions

Replace O(d) with O(1): Random sampling. (Not great, but *unbiased*)

$$\nabla f(\boldsymbol{\theta}^{(k)}) \approx \frac{f(\boldsymbol{\theta}^{(k)} + \epsilon \boldsymbol{\Delta}^{(k)}) - f(\boldsymbol{\theta}^{(k)} - \epsilon \boldsymbol{\Delta}^{(k)})}{2\epsilon} \boldsymbol{\Delta}^{(k)},$$

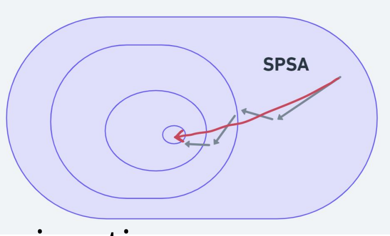
QN-SPSA:

$$\hat{g}^{(k)} = -\frac{1}{2} \frac{\delta F}{2\epsilon^2} \frac{\boldsymbol{\Delta}_1^{(k)} \boldsymbol{\Delta}_2^{(k)T} + \boldsymbol{\Delta}_2^{(k)} \boldsymbol{\Delta}_1^{(k)T}}{2}$$



* SPSA of the Quantum Fisher Information, Gacon et. al 2021

SPSA Optimizer:



Traditional Gradient Descent

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \eta \nabla f(\boldsymbol{\theta}^{(k)})$$

→

2nd Order Approximation

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \eta H^{-1}(\boldsymbol{\theta}^{(k)}) \nabla f(\boldsymbol{\theta}^{(k)}).$$

H -> Hessian of f. Not unbiased!!!

1st-Order SPSA

$$\nabla f(\boldsymbol{\theta}^{(k)}) \approx \frac{f(\boldsymbol{\theta}^{(k)} + \epsilon \boldsymbol{\Delta}^{(k)}) - f(\boldsymbol{\theta}^{(k)} - \epsilon \boldsymbol{\Delta}^{(k)})}{2\epsilon} \boldsymbol{\Delta}^{(k)}$$

→

2nd-Order SPSA

$$\hat{H}^{(k)} = \frac{\delta f}{2\epsilon^2} \frac{\boldsymbol{\Delta}_1^{(k)} \boldsymbol{\Delta}_2^{(k)T} + \boldsymbol{\Delta}_2^{(k)} \boldsymbol{\Delta}_1^{(k)T}}{2}$$

QN-SPSA (H -> g)

$$g_{ij}(\boldsymbol{\theta}) = \mathrm{Re}\left\{ \left\langle \frac{\partial \psi}{\partial \theta_i} \middle| \frac{\partial \psi}{\partial \theta_j} \right\rangle - \left\langle \frac{\partial \psi}{\partial \theta_i} \middle| \psi \right\rangle \left\langle \psi \middle| \frac{\partial \psi}{\partial \theta_j} \right\rangle \right\}$$

$$g_{ij}(\boldsymbol{\theta}) = -\frac{1}{2} \frac{\partial}{\partial \theta_i} \frac{\partial}{\partial \theta_j} |\langle \psi(\boldsymbol{\theta}') | \psi(\boldsymbol{\theta}) \rangle|^2 \Big|_{\boldsymbol{\theta}' = \boldsymbol{\theta}}$$

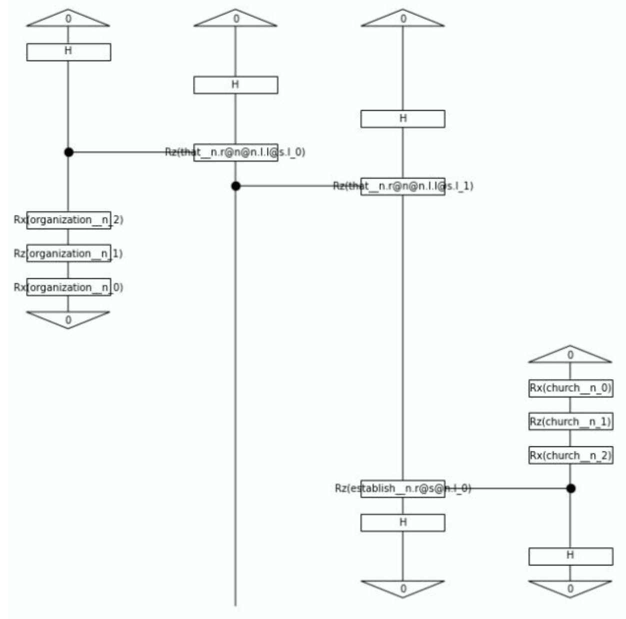
→

$$\hat{g}^{(k)} = -\frac{1}{2} \frac{\delta F}{2\epsilon^2} \frac{\boldsymbol{\Delta}_1^{(k)} \boldsymbol{\Delta}_2^{(k)T} + \boldsymbol{\Delta}_2^{(k)} \boldsymbol{\Delta}_1^{(k)T}}{2}$$

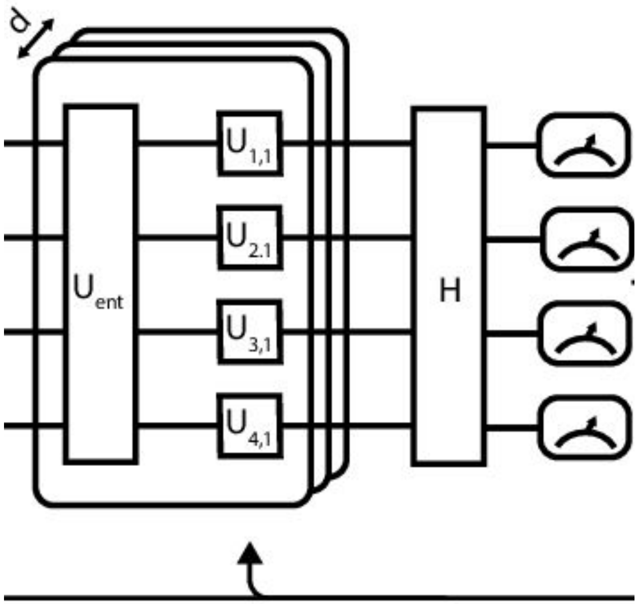
* SPSA of the Quantum Fisher Information, Gacon et. al 2021

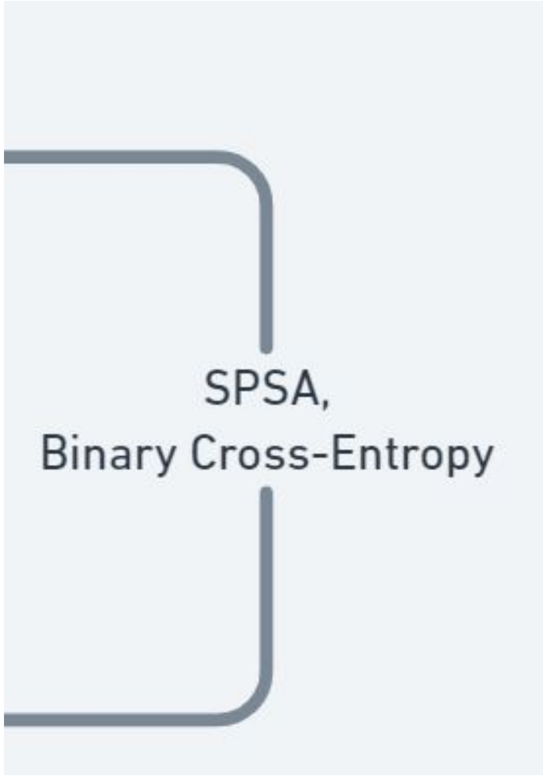
Quantum Model:

Input sentences as circuits



Parameterized Model





Recap: Quantum Simulation

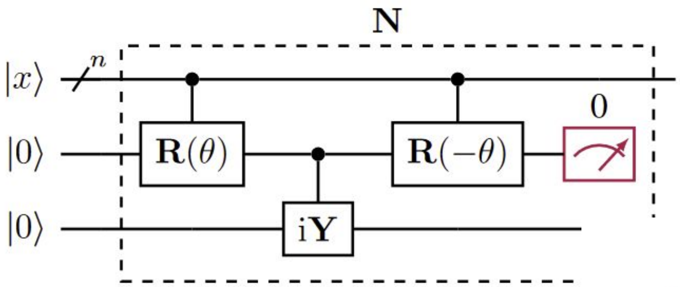
- Initialize model to pass sentence circuits into.
- Initialize loss function
- Use optimizer (SPSA) to calculate ‘Energy’ (loss).
- Repeat until convergence

Used binary-cross entropy loss function. $$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

Used “shot” data (8192) gathered from experiments to simulate quantum computation.

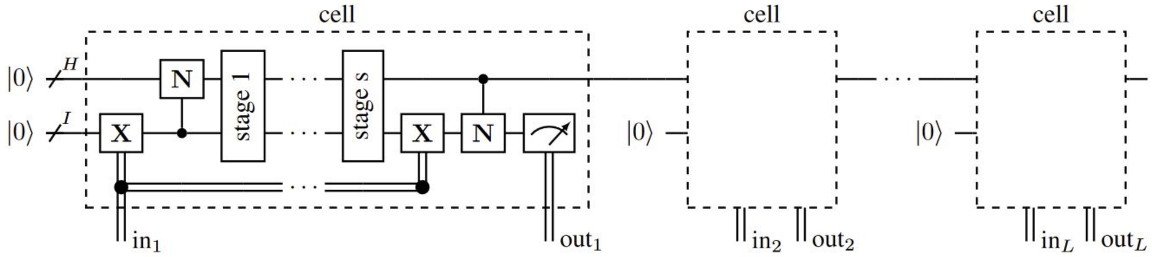
QRNN: Overview

- Iteratively apply QRNN cells to the input sequence, which perform parametrized rotations with nonlinear activation
- Use traditional optimizers (simulation) or SPSA



QRNN cell

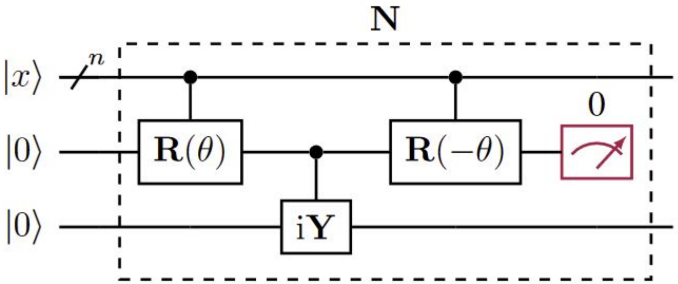
Multiple inputs applied to the same cell



Amplitude Amplification and Architecture

- QRNN cells are run in the Repeat-until-success (RUS) mode
  - Allows desired operations to be implemented in fewer Clifford gates
- Success indicated by measuring 0 on ancilla, otherwise undo and retry
- Increase probability of success with oblivious amplitude amplification (OAA)

$$A|0^m\rangle|\psi\rangle = \sqrt{\lambda_0}\,|0^m\rangle U|\psi\rangle + \sum_{i=1}^{2^m-1}\sqrt{\lambda_i}\,|i\rangle R_i|\psi\rangle$$



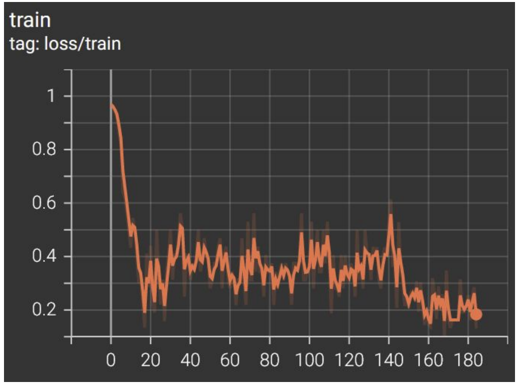
$$S_\pi = (I^m - 2|0^m\rangle\langle 0^m|)\otimes I$$

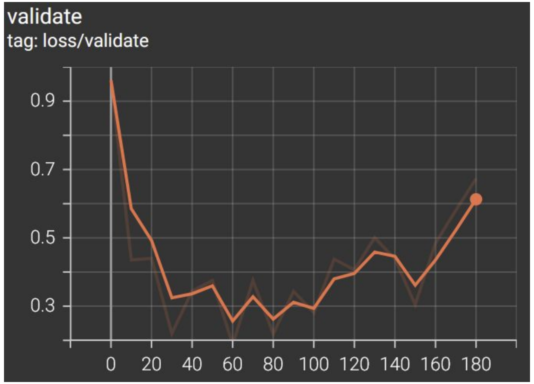
$$(-AS_\pi A^\dagger S_\pi)^j A|0^m\rangle|\psi\rangle = \sin[(2j+1)\theta]\,|0^m\rangle U|\psi\rangle + \cos[(2j+1)\theta]\,|\Phi^\perp\rangle$$

Results

Loss Function Convergence

- QRNN sees rapid convergence in ~80 batches with subsequent overfitting
  - Consistent with empirical performance analysis by Bausch et. al





Training Accuracy Results

		Models	
		Single Parameterized Cell	Recurrent Circuit
Embeddings	Classical Embedding	63.9%	75.8%
	Lambeq Embedding	69.60%	Limited by Hardware Capability

Discussion

	Single Parameterized Cell	Recurrent Circuit
Classical Embedding	Base Model	● Limited by Embedding Complexity ○ Input width was 5 bits, 439 trainable parameters ○ Only naive character-level embeddings could be used (vs Word2Vec, etc)
Lambeq Embedding	● Limited by Model Complexity ○ Input size is constrained ○ Slow when processing long inputs and using large models	● Future Work ○ If good embeddings can be obtained, can have similar results with only a small number of parameters

Acknowledgement

Intro: NLP & Sentiment Analysis

- Unfortunately, current classical NLP models find this task somewhat difficult.
    - Context-dependent
    - Other difficulties include non-straightforward speech like sarcasm.
    - This added complexity to a body of text leads to inaccurate classification
- Due to the difficulties that classical models face, we aim to test whether a quantum model will be able to perform sentiment analysis better.

Quantum Circuit

- **Quantum Circuits:**
  - Model for quantum computation, where the computation is carried out by an ordered sequence of quantum gates that work together to create the desired quantum states of certain qubits
  - Can parameterized quantum circuits that contain variational quantum gates.
    - quantum algorithms that depend on free parameters
  - Run quantum circuits on quantum computers to perform calculations to solve problems

- **Relating to sentiment analysis:**
  - At high level, words in each input sentence are transformed into quantum states by using parameterized/variational quantum gates