Title: Simulating quantum circuits with neural machine translation

Speakers: Juan Carrasquilla

Collection: Machine Learning for Quantum Design

Date: July 09, 2019 - 11:30 AM

URL: http://pirsa.org/19070025

SIMULATING QUANTUM CIRCUITS WITH WITH NEURAL MACHINE TRANSLATION

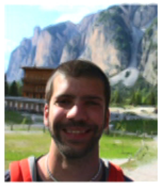
Juan Carrasquilla
Vector Institute
Canada CIFAR AI chair

Machine Learning for Quantum Design
Waterloo, ON. July 9th 2019

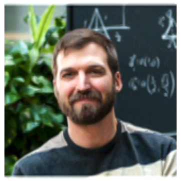
VECTOR INSTITUTE | INSTITUT VECTEUR CIFAR NSERC CRSNG UNIVERSITY OF WATERLOO

ML/QUANTUM FRIENDS

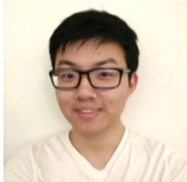
Leandro Aolita (Universidade Federal do Rio de Janeiro, ICTP-SAIFR)



Giacomo Torlai (Flatiron Institute)



Roger Melko (U. Waterloo and Perimeter)



Andrew Tan (U of Toronto—> MIT PhD)

Ashley Milsted (Perimeter)

Martin Ganahl (Perimeter)

DIMENSIONALITY OF QUANTUM SYSTEMS VS NEURAL MACHINE TRANSLATION

$|\Psi\rangle$ *vector with* 2^N

- Today's best supercomputers can solve the wave equation **exactly** for systems with a maximum of ~45 spins.

$$2^N \sim 3.5 \times 10^{13}$$

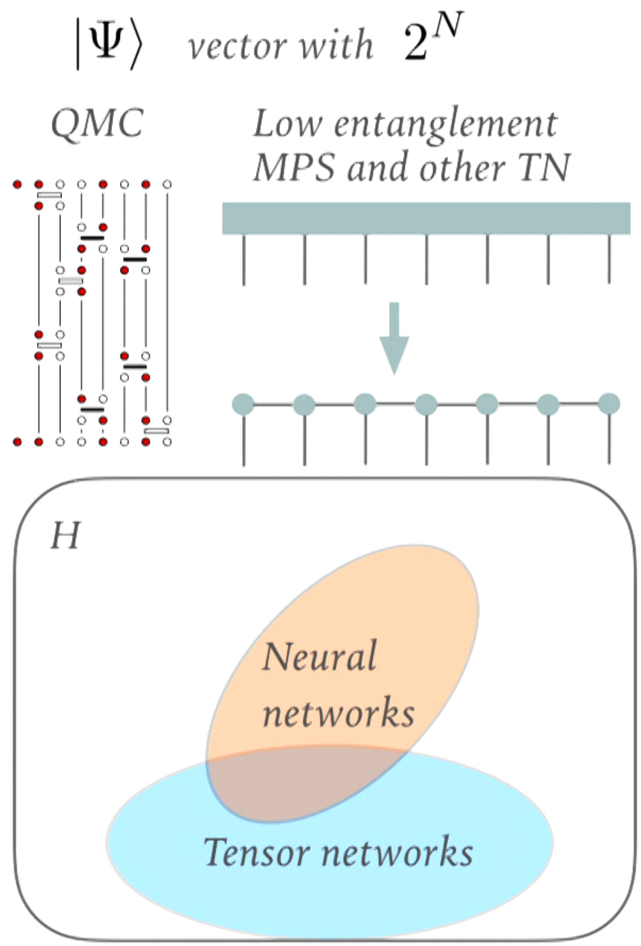
- Language translation models live in very high dimensional spaces too (example from Attention is all you need)

$$\text{Vocab. Size}^{\text{Max length of sentence}} \sim 8000^{100} \sim 2.03 \times 10^{390}$$

Storage of these distributions requires a computer with big memory: size is bigger than a number of universes bigger than there are atoms in the known universe.

QUANTUM STATES, NATURAL IMAGES, NATURAL LANGUAGES ARE "PHYSICAL"

- The amount of information for quantum states, language modelling, computer vision, is smaller than the maximum capacity
- Quantum Monte Carlo and other numerical methods based on Tensor Networks exploit this fact.
- Both quantum and ML problems have a lot of (**shared?**) structure and symmetry that we can exploit



EXAMPLE OF A LANGUAGE MODEL (TRANSFORMER)

SYSTEM PROMPT (HUMAN-WRITTEN)	*In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.*
MODEL COMPLETION (MACHINE-WRITTEN, 10 TRIES)	The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.

Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns.

While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English. Pérez stated, "We can see, for example, that they have a common 'language,' something like a dialect or dialectic."

Dr. Pérez believes that the unicorns may have originated in Argentina, where the animals were believed to be descendants of a lost race of people who lived

Input

Output

generating samples from a variety of inputs —>close to human quality and long-range coherence over a **page or more of text**

↓

Strong correlation (power law)

$$P(\text{output text}|\text{input text})$$

LANGUAGE TRANSLATION

- Language translation $P(\text{English}|\text{Spanish})$

Input

Utilizando la base de datos Epistemonikos, la cual es mantenida mediante búsquedas realizadas en 30 bases de datos, identificamos seis revisiones sistemáticas que en conjunto incluyen 36 estudios aleatorizados pertinentes a la pregunta.

Output

Using the Epistemonikos database, which is maintained through searches in 30 databases, we identified six systematic reviews that altogether include 36 randomized studies relevant to the question.

Neural Machine Translation with the Transformer and Multi-Source Romance Languages for the Biomedical WMT 2018 task. Brian Tubay and Marta R. Costa-jussa` (2018)

COMPUTER VISION: GENERATING NEW IMAGES

$P_{\text{model}}(\mathbf{a}) \longrightarrow$ **Generative adversarial networks**

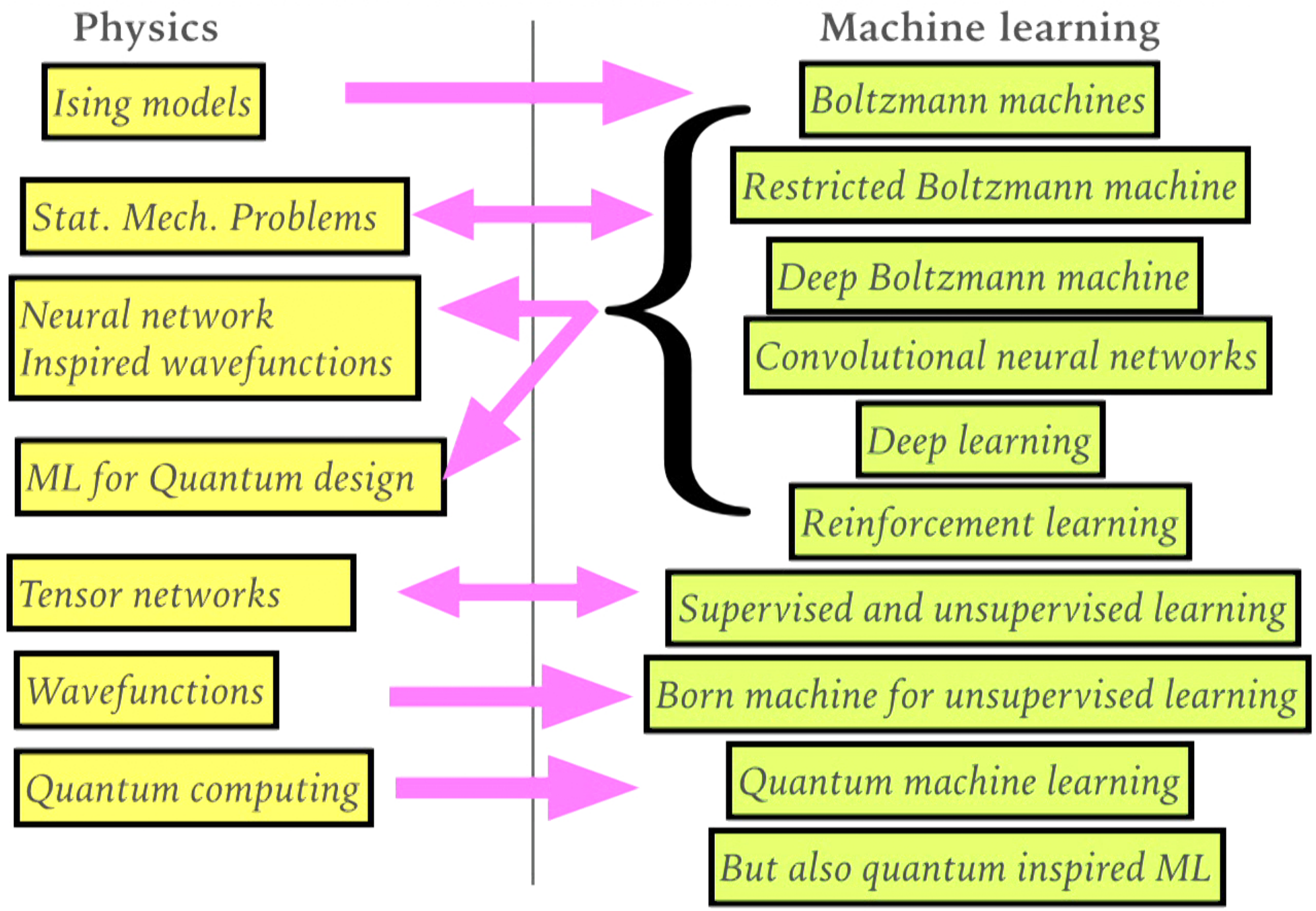
- Understand probability distributions defined over high-dimensional data like images. Sampling new faces:

https://arxiv.org/pdf/1812.04948.pdf

https://thispersondoesnotexist.com/

CAN WE USE THE POWER OF THESE MODELS TO STUDY STRONGLY INTERACTING QUANTUM SYSTEMS?

THERE IS AN INTRICATE HISTORY OF EXCHANGE BETWEEN FIELDS



THERE IS AN INTRICATE HISTORY OF EXCHANGE BETWEEN FIELDS

Physics

- *Neural network Inspired wavefunctions*
- *ML for Quantum design*

Machine learning

- *Boltzmann machines*
- *Restricted Boltzmann machine*
- *Deep Boltzmann machine*
- *Convolutional neural networks*
- *Deep learning*
- *Reinforcement learning*

We have been relatively successful —> transform ML models and algorithms to make them look like quantum mechanics.

CAN WE MAKE QUANTUM THEORY LOOK MORE LIKE MACHINE LEARNING/STATS INSTEAD?

FEYNMAN 1981:

Simulating Physics with Computers

Richard P. Feynman

5. CAN QUANTUM SYSTEMS BE PROBABILISTICALLY SIMULATED BY A CLASSICAL COMPUTER?

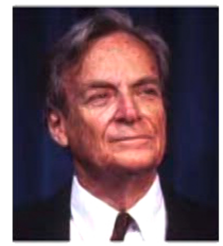
Now the next question that I would like to bring up is, of course, the interesting one, i.e., Can a quantum system be probabilistically simulated by a classical (probabilistic, I'd assume) universal computer? In other words, a computer which will give the same probabilities as the quantum system does. If you take the computer to be the classical kind I've described so far, (not the quantum kind described in the last section) and there're no changes in any laws, and there's no hocus-pocus, the answer is certainly, No! This is called the hidden-variable problem: it is impossible to represent the results of quantum mechanics with a classical universal device. To learn a little bit about it, I say let us try to put the quantum equations in a form as close as

Using a Wigner representation, Feynman concludes:

the great difficulty. The only difference between a probabilistic classical world and the equations of the quantum world is that somehow or other it appears as if the probabilities would have to go negative, and that we do not know, as far as I know, how to simulate. Okay, that's the fundamental problem. I don't know the answer to it, but I wanted to explain that if I try my best to make the equations look as near as possible to what would be imitable by a classical probabilistic computer, I get into trouble.

Motivated the whole field

Of quantum computing



This is all still true today and is **fundamentally** linked to the notion of quantum speed-up in quantum computing.

However, I'll provide a heuristic to **simulate quantum systems probabilistically** using RNNs

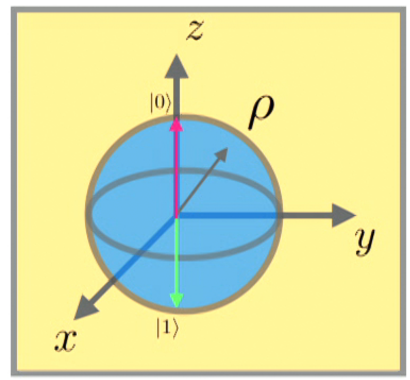
IN THIS TALK

- I will introduce a formulation of quantum physics that looks a bit like machine learning
- We use generative models, in particular RNNs to parametrize quantum states.
- I will show an example to motivate why this may be a good idea in the context of quantum state reconstruction
- I will show you a heuristic to simulate a quantum circuit with RNNs

HOW IS A QUANTUM STATE TRADITIONALLY DESCRIBED?

- A **density matrix** describes the statistical state of a system in quantum mechanics. Everything we can possibly know about a quantum system is encoded in the density matrix.
- A quantum state is a positive semidefinite, Hermitian operator of trace 1 acting on the state space.
- The family of quantum states forms a convex set. For one qubit: Bloch sphere.

$$\rho$$



HOW TO REPRESENT A QUANTUM STATE WITH ONLY PROBABILITY?

MEASUREMENTS: POSITIVE OPERATOR-VALUED MEASURE (POVM)

- POVM elements $\mathbf{M} = \{M^{(a)} \mid a \in \{1, ..., m\}\}$
- Positive semidefinite operators
- $\sum_i M^{(a)} = \mathbb{1}$

EXAMPLE: MEASUREMENT IN THE COMPUTATIONAL BASIS

$$M^{(0)} = |0\rangle\langle 0| \quad M^{(1)} = |1\rangle\langle 1| \quad \mathbb{I} = |0\rangle\langle 0| + |1\rangle\langle 1|$$

STATE $|\Psi\rangle = a|0\rangle + b|1\rangle$

$$\left.\begin{aligned} p(0) &= \langle\Psi|M^{(0)}|\Psi\rangle = |a|^2 \\ p(1) &= \langle\Psi|M^{(1)}|\Psi\rangle = |b|^2 \end{aligned}\right\}$$

BORN RULE: PROVIDES A LINK BETWEEN QUANTUM THEORY AND EXPERIMENT

MEASUREMENTS: POSITIVE OPERATOR-VALUED MEASURE (POVM)

POVM ELEMENTS $\mathbf{M} = \{M^{(a)} \mid a \in \{1, ..., m\}\}$ $\sum_i M^{(a)} = \mathbb{1}$

$P(\mathbf{a}) = \mathrm{Tr}\, \rho M^{\mathbf{a}}$ Born rule. Defines a distribution over the generalized measurements => link between quantum theory and experimental outcome

INFORMATIONALLY COMPLETE POVM

- The measurement statistics $P(\mathbf{a})$ contains all of the information about the state.
- If m is $D^2=2^{2N}$ and M span the entire Hilbert space

$$O = \sum_{\mathbf{a}} O(a) M^{(a)}$$

- Relation between ρ and distribution $P(\mathbf{a})$ can be inverted.

MEASUREMENTS: POSITIVE OPERATOR VALUED MEASURES (POVM)

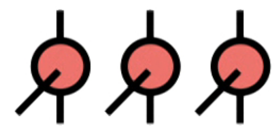
CONSTRUCTING POVMS: TAKE A SINGLE QUBIT POVM AND MAKE A TENSOR PRODUCT

Pauli measurement for one qubit

$$\begin{aligned} \boldsymbol{M}_{\text{Pauli}} := \{ & M^{(0)} := p(3) \times |0\rangle\langle 0| \,, M^{(1)} := p(3) \times |1\rangle\langle 1| \,, \\ & M^{(+)} := p(1) \times |+\rangle\langle +| \,, M^{(-)} := p(1) \times |-\rangle\langle -| \,, \\ & M^{(r)} := p(2) \times |r\rangle\langle r| \,, M^{(l)} := p(2) \times |l\rangle\langle l| \,\}, \end{aligned}$$

$$\boldsymbol{M}_{\text{Pauli}} \bigotimes \boldsymbol{M}_{\text{Pauli}} \bigotimes \boldsymbol{M}_{\text{Pauli}} \bigotimes \boldsymbol{M}_{\text{Pauli}} \bigotimes \boldsymbol{M}_{\text{Pauli}} \bigotimes \boldsymbol{M}_{\text{Pauli}} \bigotimes$$

$$\boldsymbol{M} = \left\{ M^{(a_1)} \otimes M^{(a_2)} \otimes \ldots M^{(a_N)} \right\}_{a_1, \ldots a_N}$$

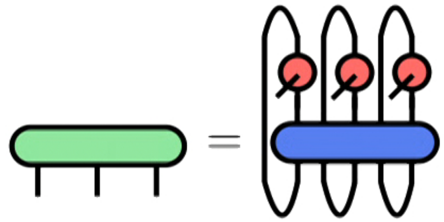


Experimental realization: pick a random direction with probability 1/3, then measure in that direction on each qubit independently

Easy to implement in gate-based QC (Qiskit, Cirq, Rigetti, etc.)

GRAPHICAL NOTATION AND INVERSE

Born rule $P(\mathbf{a}) = \mathrm{Tr}\,\rho M^{\mathbf{a}}$



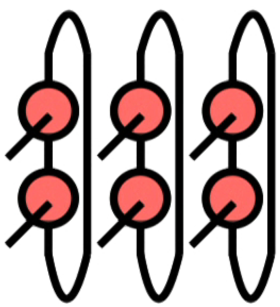
If the POVM is informationally complete then

$$\rho = \sum_a O_\rho(a) M^{(a)}$$

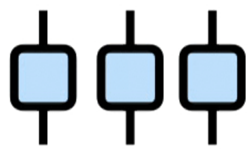
Insert this relation into Born's rule $P(a) = \sum_{a'} O_\rho(a') Tr[M^{(a)} M^{(a')}] = \sum_{a'} O_\rho(a') T_{a'a}$

$$\rho = \sum_{a,a'} T^{-1}_{a,a'} P(a') M^{(a)}$$

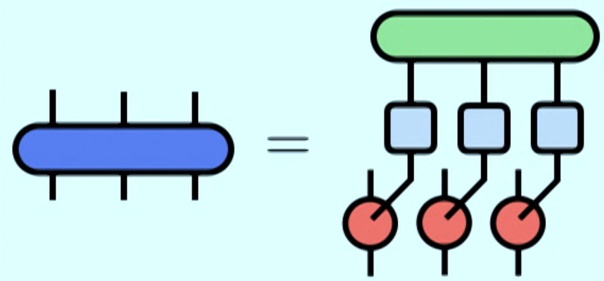
$T_{\alpha,\beta} = \mathrm{Tr}\ M^\alpha M^\beta$



$\mathbf{T}^{-1}$



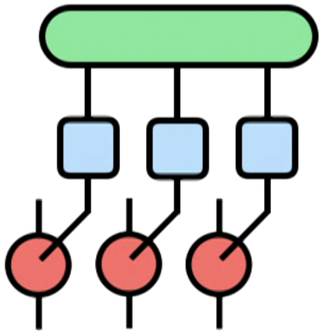
$$\rho = \sum_{a,a'} T^{-1}_{a,a'} P(a') M^{(a)}$$



SUMMARY: REPRESENTATION OF THE QUANTUM STATE

MODEL FOR THE QUANTUM STATE

$$\rho = (T^{-1}P)^T M$$



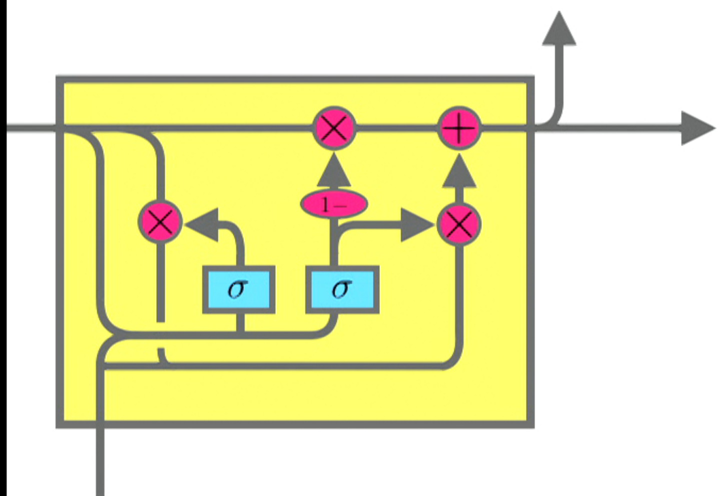
- Factorization of the state in terms of a probability distribution and a set of tiny tensors
- All the **entanglement** and potential complexity of the state comes from the structure of the P(a)
- Very efficient to handle numerically for some tasks
- Sign structure of the state is in the tiny tensors

KEY INSIGHT: PARAMETRIZE STATISTICS OF MEASUREMENT AND INVERT

$$P(\mathbf{a}) = \mathrm{Tr}\,\rho M^{\mathbf{a}}$$

=> Unsupervised learning of P(a)

$P_{\mathrm{model}}(\mathbf{a})$ →



Autoregressive models

1. Allow for exact sampling
2. Tractable density $P_{\mathrm{model}}(\mathbf{a})$
3. Traditionally used in neural machine translation

KEY INSIGHT: PARAMETRIZE STATISTICS AND INVERT

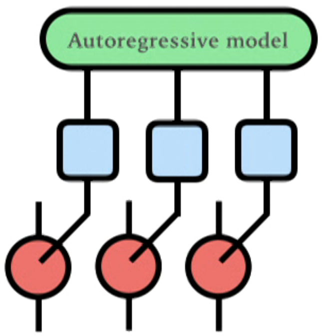
$$P(\mathbf{a}) = \mathrm{Tr}\, \rho M^{\mathbf{a}}$$

=> Unsupervised learning of P(a)

$P_{\text{model}}(\mathbf{a})$ →

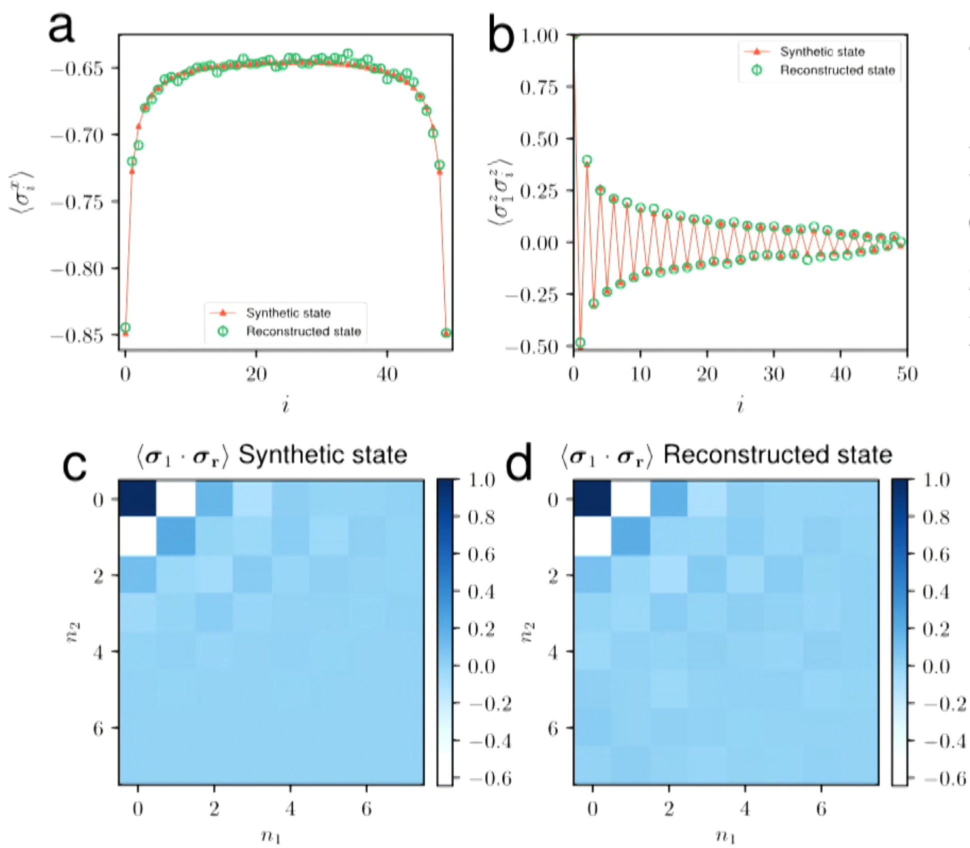
Autoregressive models

1. Allow for exact sampling
2. Tractable density $P_{\text{model}}(\mathbf{a})$
3. Traditionally used in neural machine translation



EXAMPLE: LEARN A QUANTUM STATE FROM MEASUREMENTS

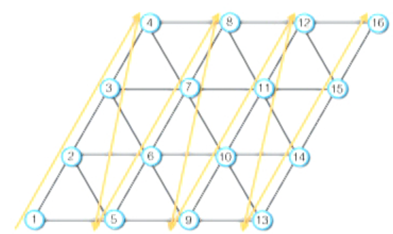
LEARNING GROUND STATES OF LOCAL HAMILTONIANS FROM DATA



$$\mathcal{H} = J\sum_{ij}\sigma_i^z\sigma_j^z + h\sum_i \sigma_i^x$$

N=50 spins. P(a) is a deep (3 layer GRU) recurrent neural network language model.

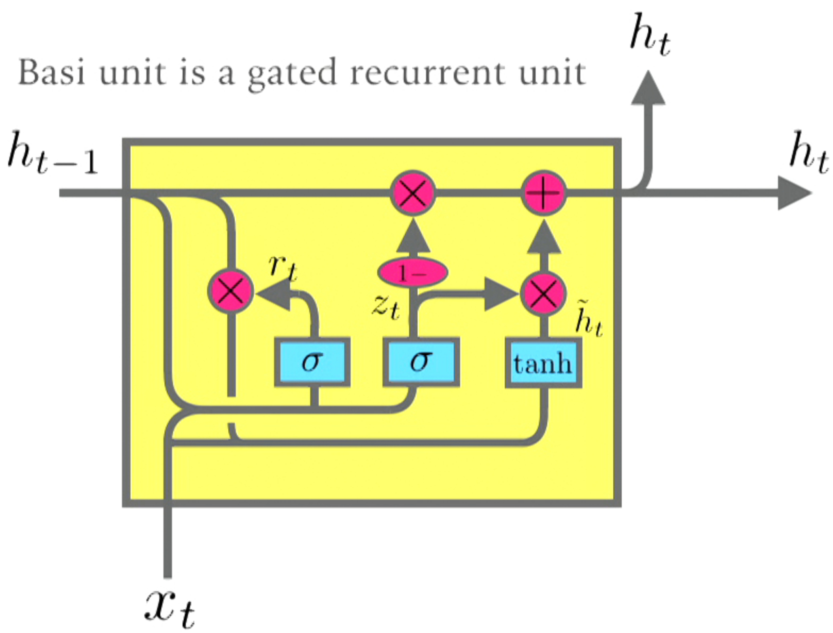
$$H = J\sum_{i,j}\boldsymbol{\sigma}_i.\boldsymbol{\sigma}_j$$



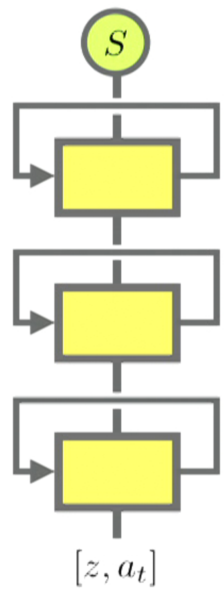
Carrasquilla, Torlai, Melko, Aolita. Nature Machine Intelligence 1, 200 (2019)

RECURRENT NEURAL NETWORK MODEL

Basi unit is a gated recurrent unit



Full model stacks three of these units and adds a softmax dense layer at each "time" step



BUT QUANTUM THEORY GOES BEYOND REPRESENTATION. DYNAMICS (E.G. SCHRODINGER EQUATION)? MEASUREMENTS?

UNITARY DYNAMICS AND QUANTUM CHANNELS

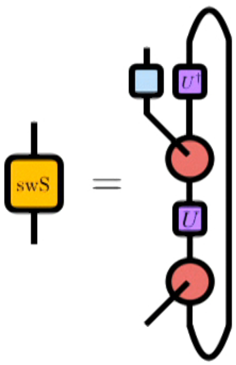
$$\varrho_U = U \varrho U^\dagger$$ **BORN RULE** $$P_U(\boldsymbol{a}'') = \mathrm{Tr}\left[U \varrho U^\dagger M^{(\boldsymbol{a}'')}\right] = \sum_{\boldsymbol{a}'} O_{\boldsymbol{a}''\boldsymbol{a}'} P(\boldsymbol{a}')$$

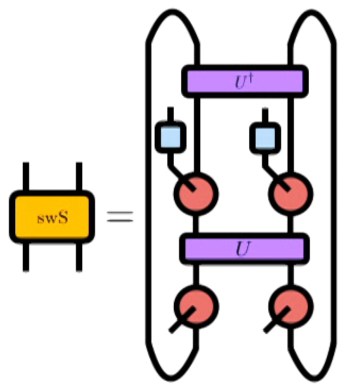
$$O_{\boldsymbol{a}''\boldsymbol{a}'} = \sum_{\boldsymbol{a}} \mathrm{Tr}\left[U M^{(\boldsymbol{a})} U^\dagger M^{(\boldsymbol{a}'')}\right] T^{-1}_{\boldsymbol{a},\boldsymbol{a}'}$$

Probabilistic gates: **Somewhat** (or quasi) stochastic matrices

$$P_U(\boldsymbol{a}'') = \sum_{\boldsymbol{a}'} O_{\boldsymbol{a}''\boldsymbol{a}'} P(\boldsymbol{a}')$$

Evolution of probability is **somewhat** classical :)





If the starting unitaries are k-local, the swS matrices are also k-local

Somewhat stochastic matrices

Branko Ćurgus, Robert I. Jewett

(Submitted on 3 Sep 2007)

The standard theorem for regular stochastic matrices is generalized to matrices with no sign restriction on the entries. The condition that column sums be equal to 1 is kept, but the regularity condition is replaced by a condition on the ℓ_1-distances between columns.

UNITARY DYNAMICS AND QUANTUM CHANNELS

$$\varrho_U = U \varrho U^\dagger$$

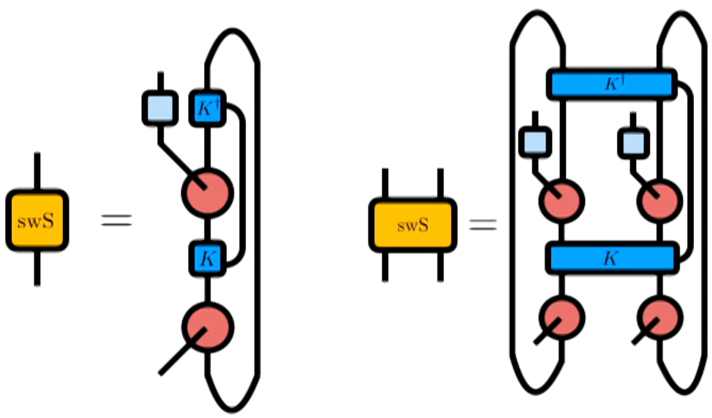
$$P_U\left(\boldsymbol{a}''\right) = \mathrm{Tr}\left[U \varrho U^\dagger M^{(\boldsymbol{a}'')}\right] = \sum_{\boldsymbol{a}'} O_{\boldsymbol{a}''\boldsymbol{a}'} P(\boldsymbol{a}')$$

$$O_{\boldsymbol{a}''\boldsymbol{a}'} = \sum_{\boldsymbol{a}} \mathrm{Tr}\left[U M^{(\boldsymbol{a})} U^\dagger M^{(\boldsymbol{a}'')}\right] T^{-1}_{\boldsymbol{a},\boldsymbol{a}'}$$

Probabilistic gates: Somewhat (or quasi) stochastic matrices

$$P_U\left(\boldsymbol{a}''\right) = \sum_{\boldsymbol{a}'} O_{\boldsymbol{a}''\boldsymbol{a}'} P(\boldsymbol{a}')$$

Evolution of probability is somewhat classical :)



If the starting unitaries are k-local, the swS matrices are also k-local

Somewhat stochastic matrices

Branko Ćurgus, Robert I. Jewett

(Submitted on 3 Sep 2007)

The standard theorem for regular stochastic matrices is generalized to matrices with no sign restriction on the entries. The condition that column sums be equal to 1 is kept, but the regularity condition is replaced by a condition on the ℓ_1-distances between columns.

QUANTUM DYNAMICS

Schrödinger equation

$$i\frac{\partial P(\boldsymbol{a}'',t)}{\partial t} = \sum_{\boldsymbol{a},\boldsymbol{a}'} \mathrm{Tr}\left(\left[\mathcal{H}, M^{(\boldsymbol{a})}\right] M^{(\boldsymbol{a}'')}\right) T^{-1}_{\boldsymbol{a},\boldsymbol{a}'} P(\boldsymbol{a}',t)$$

BORN RULE

$$i\frac{\partial\rho}{\partial t} = [\mathcal{H},\rho]$$

"Solution"

$$\boldsymbol{P}(t) = e^{-iAt}\boldsymbol{P}(0) \qquad A_{\boldsymbol{a}''\boldsymbol{a}'} = \sum_{\boldsymbol{a}} T^{-1}_{\boldsymbol{a},\boldsymbol{a}'}\left[\mathrm{Tr}\left(\left[\mathcal{H},M^{(\boldsymbol{a})}\right]M^{(\boldsymbol{a}'')}\right)\right]$$

QUANTUM DYNAMICS OF OPEN QUANTUM SYSTEMS

Linblad equation

$$i\frac{\partial P(\boldsymbol{a},t)}{\partial t} = \sum_{\boldsymbol{a}} A_{\boldsymbol{a}'',\boldsymbol{a}'} P(\boldsymbol{a},t)$$

$$\boldsymbol{P}(t) = e^{-iAt}\boldsymbol{P}(0)$$

$$\begin{aligned} A_{\boldsymbol{a}''\boldsymbol{a}'} = \sum_{\boldsymbol{a}} T^{-1}_{\boldsymbol{a},\boldsymbol{a}'} \Big(\mathrm{Tr}\left(\left[\mathcal{H}, M^{(\boldsymbol{a})}\right] M^{(\boldsymbol{a}'')}\right) \\ + \sum_k \Big[-\frac{i}{2}\mathrm{Tr}\left(\{L_k^\dagger L_k, M^{(\boldsymbol{a})}\} M^{(\boldsymbol{a}'')}\right) \\ + i\,\mathrm{Tr}\left(L_k M^{(\boldsymbol{a})} L_k^\dagger M^{(\boldsymbol{a}'')}\right)\Big]\Big) \end{aligned}$$

MEASUREMENTS

- Suppose we want to measure the quantum state. The measurement is described by some other POVM $\Pi^{(\boldsymbol{b})}$

$$P(\boldsymbol{b}) = \sum_{\boldsymbol{a},\boldsymbol{a}'} P(\boldsymbol{a}')\, T^{-1}_{\boldsymbol{a},\boldsymbol{a}'}\, \mathrm{Tr}\left[M^{(\boldsymbol{a})}\Pi^{(\boldsymbol{b})}\right] = \sum_{\boldsymbol{a}'} q(\boldsymbol{b}|\boldsymbol{a}')P(\boldsymbol{a}')$$

$$q(\boldsymbol{b}|\boldsymbol{a}') = \sum_{\boldsymbol{a}} T^{-1}_{\boldsymbol{a},\boldsymbol{a}'}\, \mathrm{Tr}\left[M^{(\boldsymbol{a})}\Pi^{(\boldsymbol{b})}\right]$$

- can be characterized as a **somewhat** conditional probability since its entries can either be positive or negative but its trace over b is the identity.
- evocative resemblance with the law of total probability—> **quantum law of total probability** in quantum Bayesianism.

CIRCUITS AND TENSOR NETWORKS IN OUR LANGUAGE

Quantum circuits and quantum computing

$$P_U(\mathbf{a}'') = \sum_{\mathbf{a}'} P(\mathbf{a}') O_{\mathbf{a}',\mathbf{a}''}$$

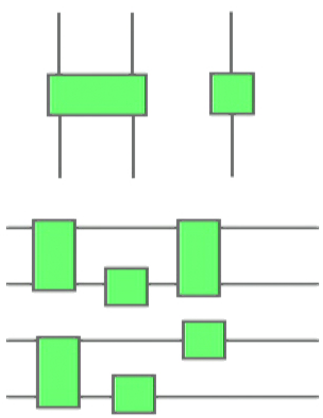
$$O_{\mathbf{a}'\mathbf{a}''} = \sum_{\mathbf{a}} \mathrm{Tr}(U M^{(\mathbf{a})} U^{\dagger} M^{(\mathbf{a}'')}) T^{-1}_{\mathbf{a},\mathbf{a}'}$$

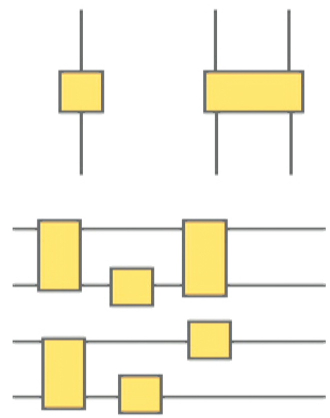
$$\rho_U = U \rho U^{\dagger}$$

Unitary matrices U

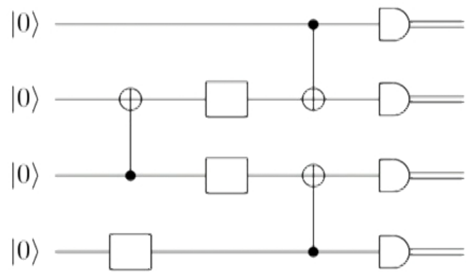
Completely positive (CP) trace preserving map

Tensor networks and quantum circuits





QUANTUM CIRCUIT



- Start the quantum device in a simple product state
- Apply a sequence of simple unitary matrices acting on the initial state
- To obtain the result, usually measure in the computational basis.

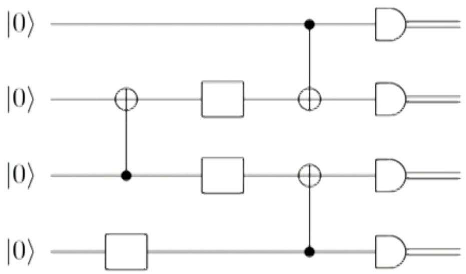
That what a general quantum computation is. The quantum algorithms (set of unitaries) are designed so that measuring the evolved quantum state results in the solution of a computational problem

$$\rho_U = U_N ... U_1 \rho_0 U_1^\dagger ... U_N^\dagger$$

$$\mathbf{P}_U = \mathbf{O}_N ... \mathbf{O}_2 \mathbf{O}_1 \mathbf{P}_0$$

Quantum computing for the very curious https://quantum.country/qcvc

QUANTUM CIRCUIT



- Start the quantum device in a simple product state
- Apply a sequence of simple unitary matrices acting on the initial state
- To obtain the result, usually measure in the computational basis.

Looks similar to Green's function Monte Carlo, but it has sign problem

Because O are somewhat stochastic

$$\rho_U = U_N ... U_1 \rho_0 U_1^\dagger ... U_N^\dagger$$

$$\mathbf{P}_U = \mathbf{O}_N ... \mathbf{O}_2 \mathbf{O}_1 \mathbf{P}_0$$

Quantum computing for the very curious https://quantum.country/qcvc

SIMULATING QUANTUM COMPUTERS WITH RNN

SIMULATING QUANTUM CIRCUITS WITH RNN

$$P_U(\mathbf{a}'') = \sum_{\mathbf{a}'} P(\mathbf{a}') O_{\mathbf{a}',\mathbf{a}''}$$

Take an initial distribution

Multiply it by a somewhat stochastic matrix

Results in an evolved distribution Pu(a)

Problem: P lives in a huge dimensional space —> $4^{\#\text{ of qubits}}$

Language translation models live in even higher dimensional spaces—> use RNNs to learn Pu

SIMULATING QUANTUM CIRCUITS WITH RNN

$$P_U(\mathbf{a}'') = \sum_{\mathbf{a}'} P(\mathbf{a}') O_{\mathbf{a}',\mathbf{a}''}$$

Introduce a model $P_\theta(\mathbf{a})$

Compute "distance" between model and evolved $P_U(\mathbf{a})$ through sampling

Minimize distance

$$D_{\mathrm{KL}}\left(P_U || P_\theta\right) = -\sum_{\mathbf{a}} P_U\left(\mathbf{a}\right) \ln \frac{P_\theta\left(\mathbf{a}\right)}{P_U\left(\mathbf{a}\right)}$$

$$D_{\mathrm{KL}}\left(P_U || P_\theta\right) = H(P_U, P_\theta) - H(P_U)$$

$$H(P_U, P_\theta) = -\sum_{\mathbf{a}} P_U\left(\mathbf{a}\right) \ln P_\theta\left(\mathbf{a}\right) = -\sum_{\mathbf{a},\mathbf{a}'} P\left(\mathbf{a}'\right) O_{\mathbf{a}\mathbf{a}'} \ln P_\theta\left(\mathbf{a}\right)$$

SIMULATING QUANTUM CIRCUITS WITH RNN

$$H(P_U, P_\theta) = -\sum_{\mathbf{a}} P_U(\mathbf{a}) \ln P_\theta(\mathbf{a}) = -\sum_{\mathbf{a},\mathbf{a}'} P(\mathbf{a}')\, O_{\mathbf{a}\mathbf{a}'} \ln P_\theta(\mathbf{a})$$

$$H(P_U, P_\theta) \approx -\frac{1}{N_s} \sum_{\mathbf{a}' \sim P} \sum_{\mathbf{a}} O_{\mathbf{a},\mathbf{a}'} \ln P_\theta(\mathbf{a})$$

Minimize cross entropy to search for an approximation to Pu from samples drawn from P

SIMULATING QUANTUM CIRCUITS WITH RNN

Sequence of N unsupervised learning problems

ALGORITHM FOR APPLYING N QUANTUM GATES

Given N gates, and an initial simple state $P_0 \longrightarrow P(a_1)P(a_2)\cdots P(a_N)$

$$P_i = P_0$$

For k in N:

Get Ns samples from P_i

Use samples to train P_θ *minimizing* $H(P_U, P_\theta) \approx -\frac{1}{N_s}\sum_{\mathbf{a}' \sim P_i}\sum_{\mathbf{a}} O_{\mathbf{a},\mathbf{a}'} \ln P_\theta(\mathbf{a})$

$$P_i = P_\theta$$

At the end of the algorithm, the model P_θ

$$\rho_\theta \approx U_N ... U_1 \rho_0 U_1^\dagger ... U_N^\dagger$$

BORN RULE

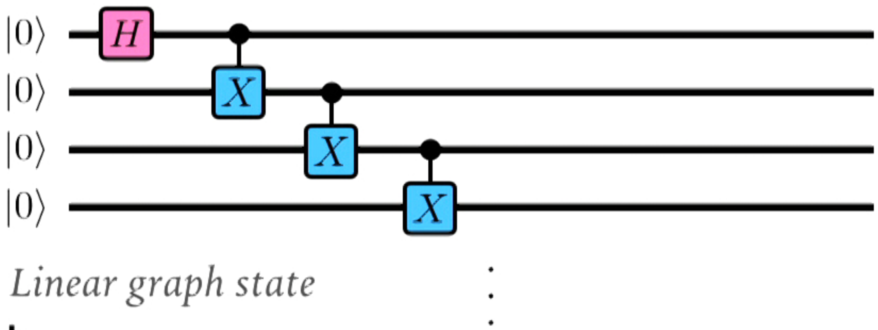
$$\mathbf{P}_\theta \approx \mathbf{O}_N ... \mathbf{O}_2 \mathbf{O}_1 \mathbf{P}_0$$

RESULTS: STATE PREPARATION FOR SIMPLE STATES IN QUANTUM INFORMATION

GHZ state

$$\text{CNOT} = cX = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

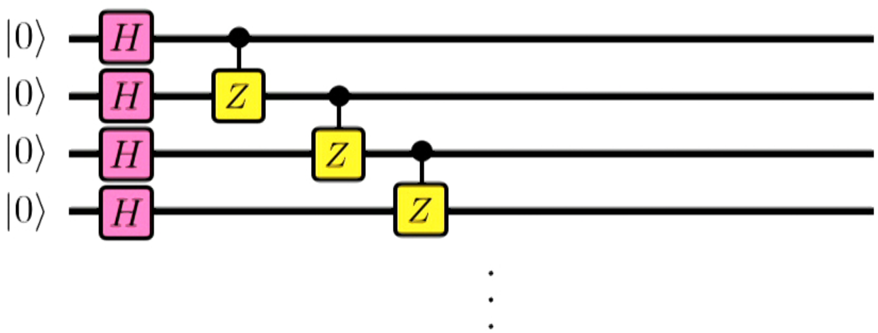
a



$$H_1 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

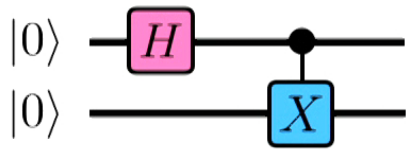
Linear graph state

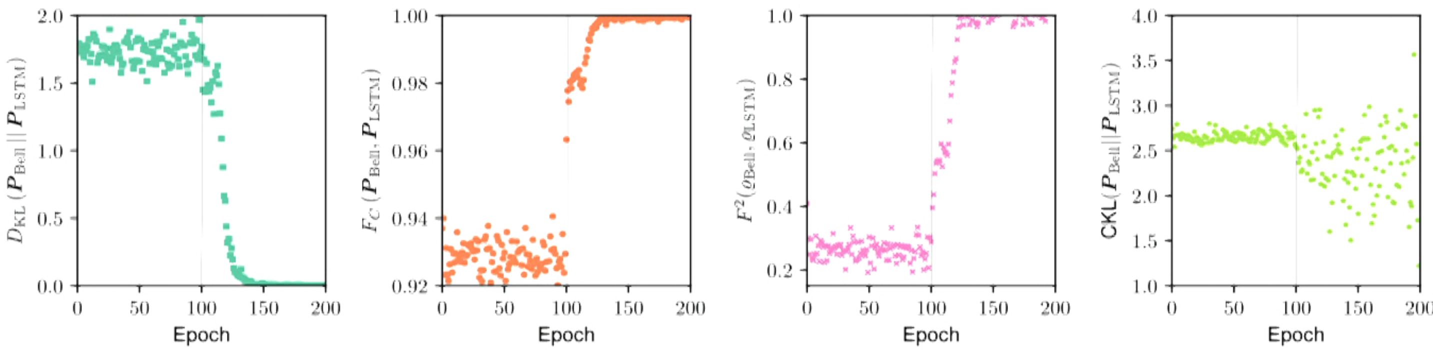
b



$$cZ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

TRAINING DYNAMICS OF THE BELL STATE PREPARATION





Ns = 10000

Batch size = 100

LSTM model with two stacked layers with hidden states d = 10 followed by a softmax

$$M_{4P}^{(0)} = \frac{1}{3}\left|0\right\rangle\left\langle 0\right|$$

$$M_{4P}^{(1)} = \frac{1}{3}\left|+\right\rangle\left\langle +\right|$$

$$M_{4P}^{(2)} = \frac{1}{3}\left|r\right\rangle\left\langle r\right|$$

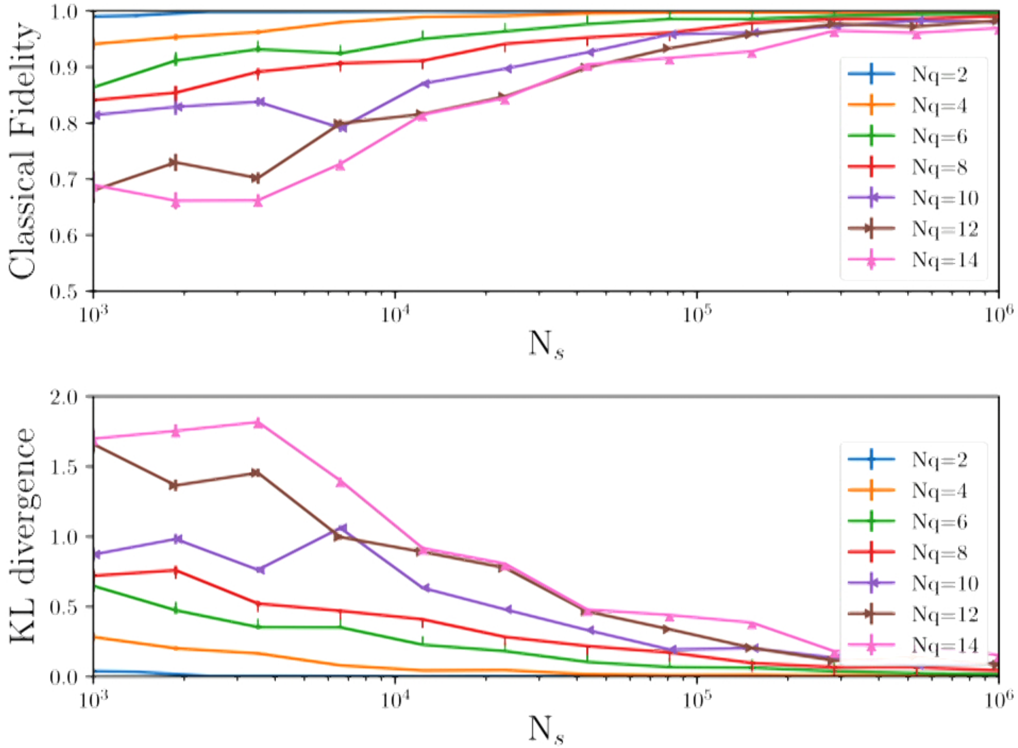
$$M_{4P}^{(3)} = \mathbf{1} - M^{(0)} - M^{(1)} - M^{(2)}$$

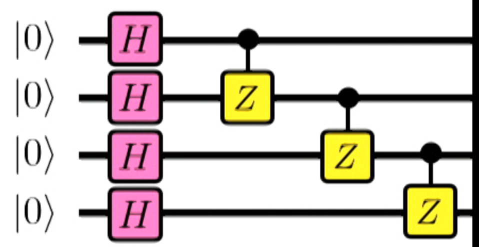
$$KL\left(P_{\mathrm{model}}|P\right) = -\sum_a P(a)\log\frac{P_{model}(a)}{P(a)}$$

$$F_{\mathrm{Classical}} = \sum_a \sqrt{P(a)P_{model}(a)}$$

$$F\left(\rho,\sigma\right) = \mathrm{Tr}\left[\sqrt{\sqrt{\rho}\,\sigma\sqrt{\rho}}\right]$$

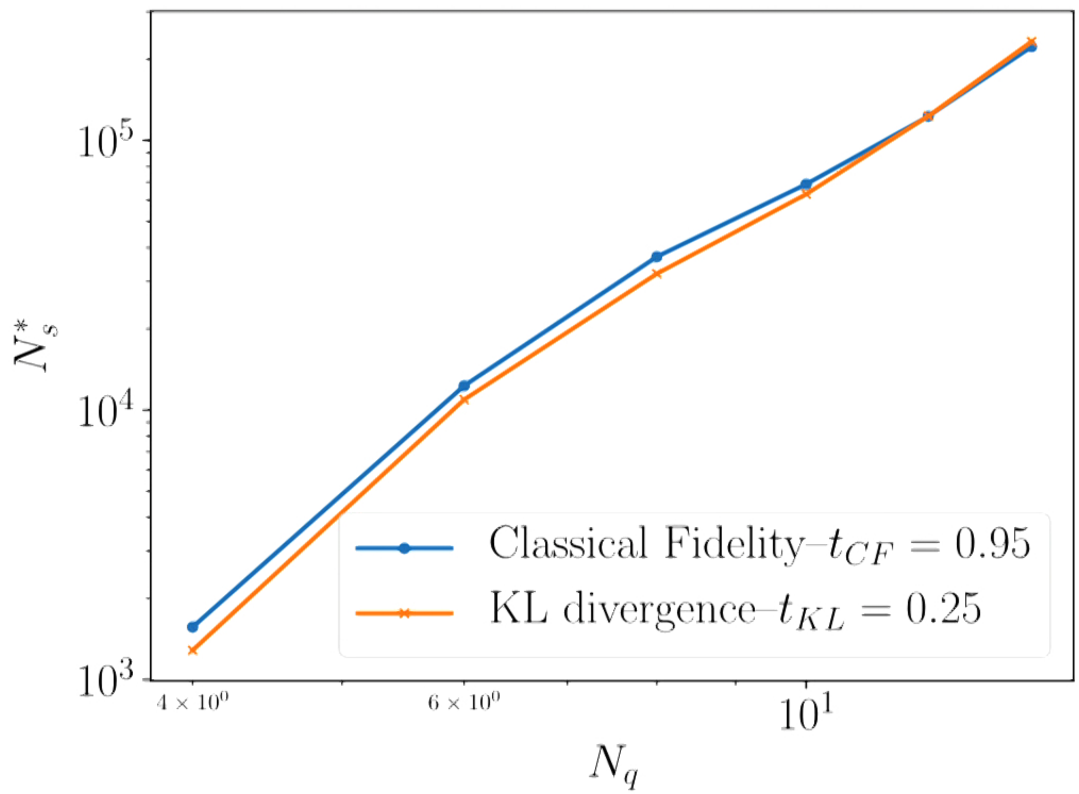
SAMPLE COMPLEXITY ANALYSIS OF THE LEARNING PROBLEM: GRAPH STATE





Ns is number of samples from P per gate applied

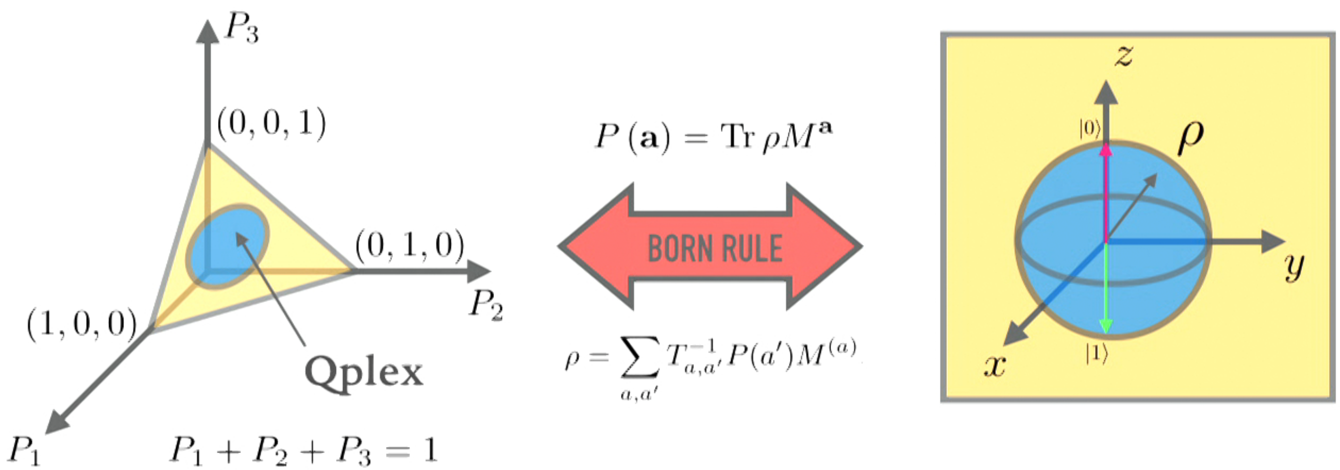
SAMPLE COMPLEXITY ANALYSIS OF THE LEARNING PROBLEM



Linear fit gives a slope of ~ 4.0, high complexity but still poly.

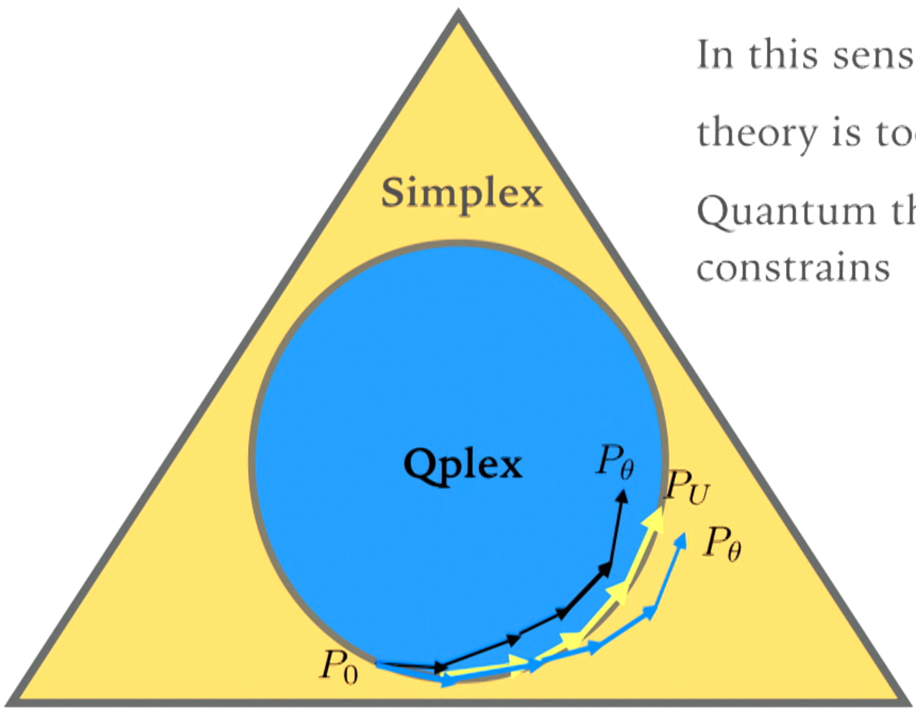
THE STANDARD SIMPLEX AND QUANTUM STATES

In probability, the points of the standard *n*-simplex in (*n* + 1)-space are the space of possible parameters (probabilities) of the categorical distribution on *n* + 1 possible outcomes.



$\{x \in \mathbb{R}^{k+1} : x_0 + x_1 + \ldots + x_k = 1, x_i \geq 0, i = 0, 1, \ldots k\}$ "All convex bodies behave a bit like Euclidean balls." Keith Ball

POSITIVITY AND VISUALIZING THE TRAINING IN THE Q-PLEX



In this sense probability theory is too general and Quantum theory needs constrains

Introducing the Qplex: A Novel Arena for Quantum Theory. Marcus Appleby, Christopher A. Fuchs, Blake C. Stacey, Huangjun Zhu. arXiv:1612.03234 [quant-ph]

CAN WE SIMULATE QUANTUM SYSTEMS WITH PROBABILITY?

5. CAN QUANTUM SYSTEMS BE PROBABILISTICALLY SIMULATED BY A CLASSICAL COMPUTER?

Now the next question that I would like to bring up is, of course, the interesting one, i.e., Can a quantum system be probabilistically simulated by a classical (probabilistic, I'd assume) universal computer? In other words, a computer which will give the same probabilities as the quantum system does. If you take the computer to be the classical kind I've described so far, (not the quantum kind described in the last section) and there're no changes in any laws, and there's no hocus-pocus, the answer is certainly, No! This is called the hidden-variable problem: it is impossible to represent the results of quantum mechanics with a classical universal device. To learn a little bit about it, I say let us try to put the quantum equations in a form as close as

Answer is still NO (duh), since evolving these distributions remains a challenge (eg circuits evolution has a sign problem). However we have introduced a **heuristic** to do it using RNNs

CONCLUSIONS

- Introduced a reformulation of quantum mechanics that is "closer" to statistics. A similar formulation is used in quantum Bayesian theory.
- Started exploring how to use this reformulation to use language translation models to simulate quantum circuits
- How to make this more efficient? Transformers, other generative models or give up :)
- Would like to find generative models directly living in the qplex
- Dream: run quantum algorithms on the transformer