Artiste-qb.net

Quantum Machine Learning, TensorFlow and Edward

Feynman quote

Talk by R.R.Tucci

Quantum Toronto

PottersVille is allusion to famous American movie “It’s a Wonderful Life”

Artiste-qb.net

Quantum Open Source

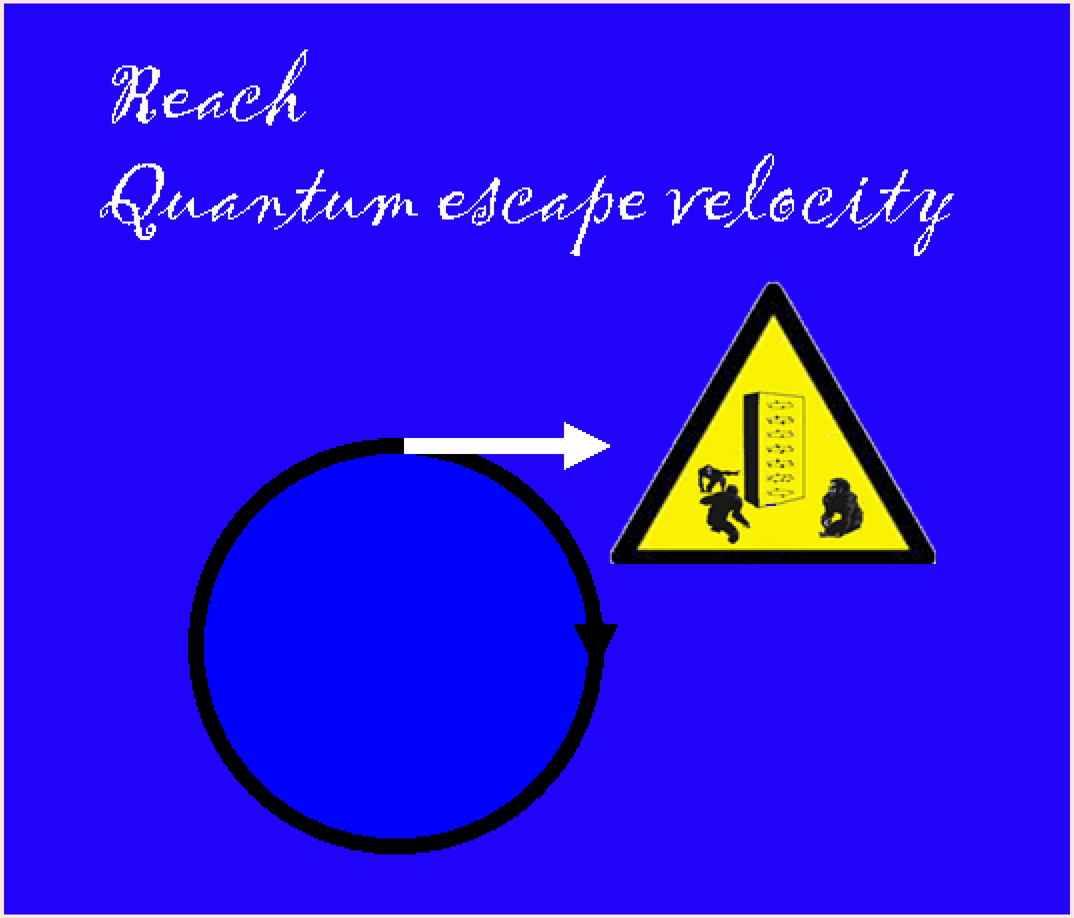
From
Classic
1984
Apple adv.

Artiste-qb.net

Quantum Ready

Artiste-qb.net

Reach Quantum Escape Velocity (Quantum Supremacy)



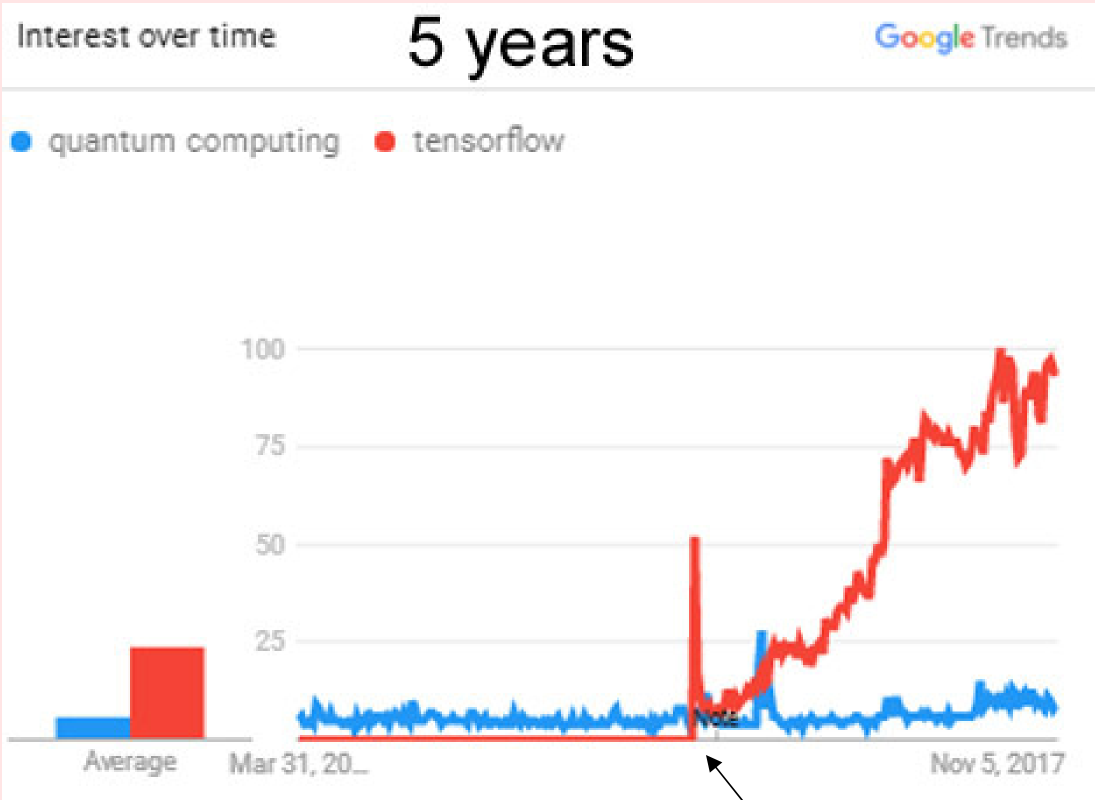
Artiste-qb.net

What Does a Quantum Computer Look Like

Everything in this image except blue rings comes from tee-shirt sold on web

Artiste-qb.net

Result of GoogleTrends search for keywords "TensorFlow" and "Quantum Computing"



TensorFlow Released as Open-Source

Relationship between TensorFlow and Edward

Before Edward came along, TensorFlow could only do networks with deterministic nodes.

With the addition of Edward, TensorFlow now can do nets with both deterministic and non-deterministic (probabilistic) nodes.

The main author of Edward is Dustin Tran. He started writing Edward at Columbia Univ, as part of his Ph.D. thesis, under Prof. David Blei.

Edward is now officially a part of TensorFlow, and Dustin Tran now works for Google/TensorFlow.

artiste-qb.net

Pioneers mentioned in this talk

Judea Pearl, Bayesian Networks

Andrew Gelman, Hierachical Models

Hierarchical Models \subset Bayesian Networks

Dustin Tran, Main Author of Edward Lib

David Blei, thesis advisor of Dustin,
one of 3 authors of seminal paper for
BBVI (Black Box Variational Inference)

Artiste-qb.net

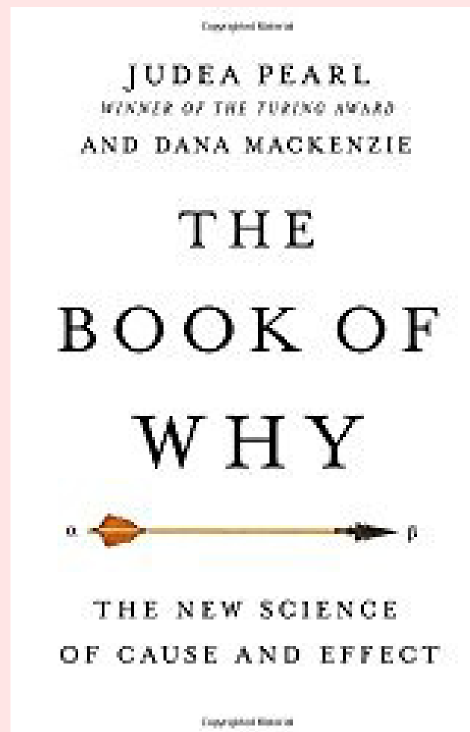
Judea Pearl (UCLA)

Bayesian Networks, Causality.

Winner of Turing Prize in Computer Science

Wikipedia: https://en.wikipedia.org/wiki/Judea_Pearl

Amazon Page: https://www.amazon.com/Judea-Pearl/e/B001HCTYSO/



Just Published, May 2018

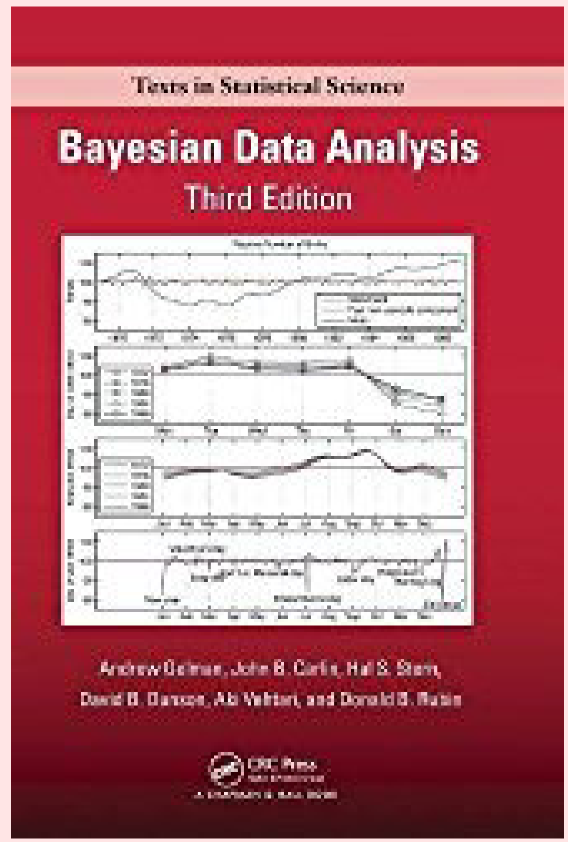
Artiste-qb.net

Andrew Gelman (Columbia Univ)

Hierarchichal Models

Wikipedia: en.wikipedia.org/wiki/Andrew_Gelman

Amazon Page: https://www.amazon.com/Andrew-Gelman/e/B001IGUSKM/



Artiste-qb.net

Edward Lib

Home: http://edwardlib.org/

Forum : https://discourse.edwardlib.org/

GitHub: https://github.com/blei-lab/edward

Dustin Tran

Relationship between Edward and Quantum Machine Learning (1/2)

Classical Bayesian Networks → Quantum Bayesian Networks

Probability matrices attached to each node → Matrices of complex "amplitudes" attached to each node

Probability → |Amplitude|^2

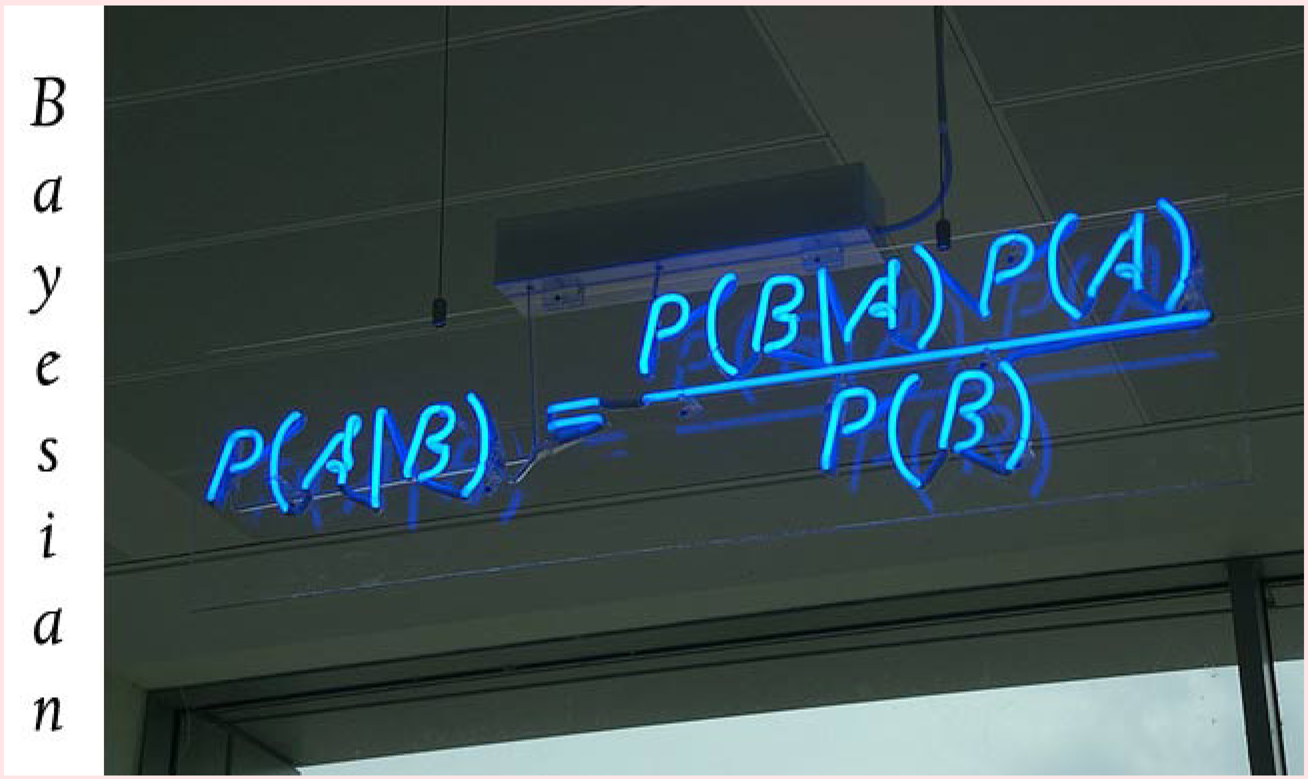
"Quantum Bayesian Nets", by R.R. Tucci,
https://arxiv.org/abs/quant-ph/9706039

"Factorization of Quantum Density Matrices According to Bayesian and Markov Networks", by R.R. Tucci
https://arxiv.org/abs/quant-ph/0701201

Quantum

"God does not play dice with the Universe", A. Einstein
Referring to Quantum Mechanics. God from Sistine Chapel

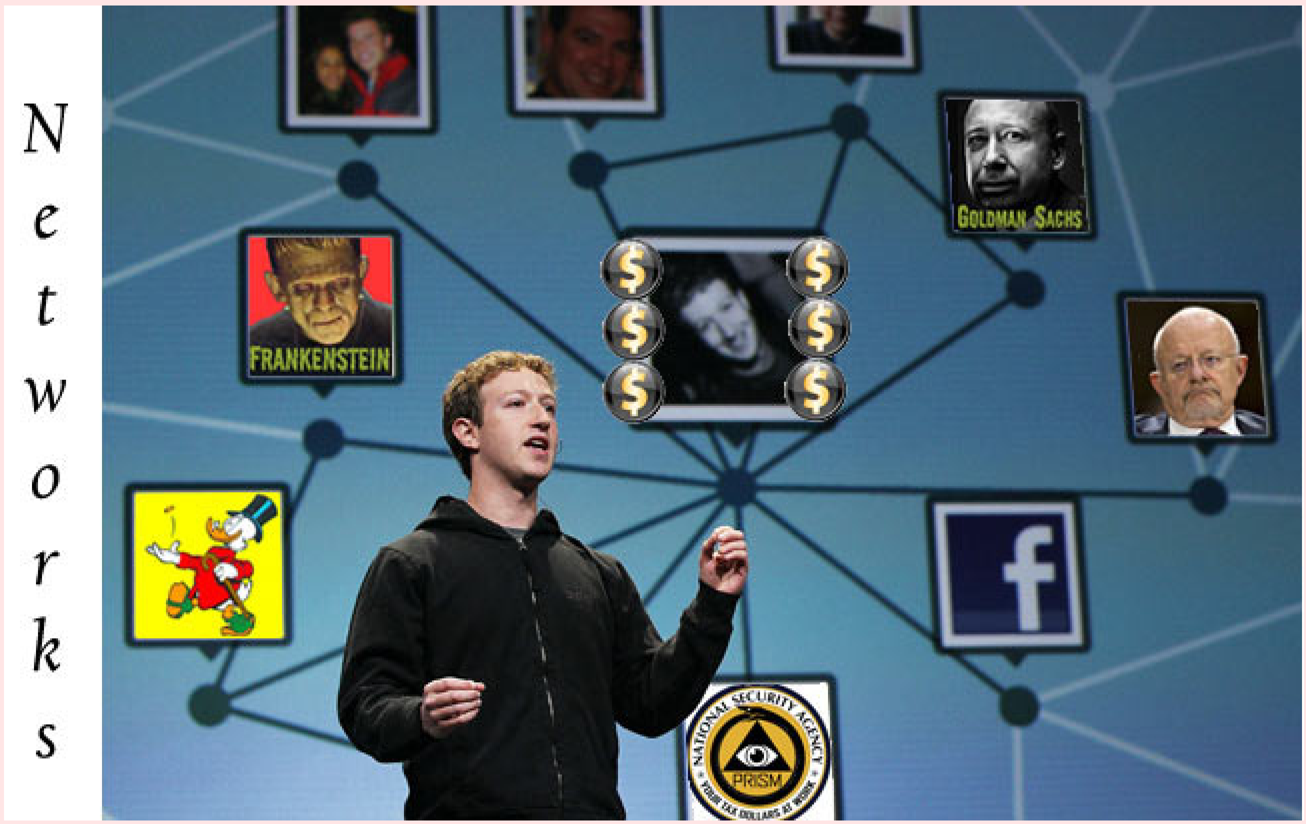
Bayesian



Neon sign from Autonomy Corp. HQ in Cambridge, UK

Artiste-qb.net

Networks



Relationship between Edward and Quantum Machine Learning (2/2)

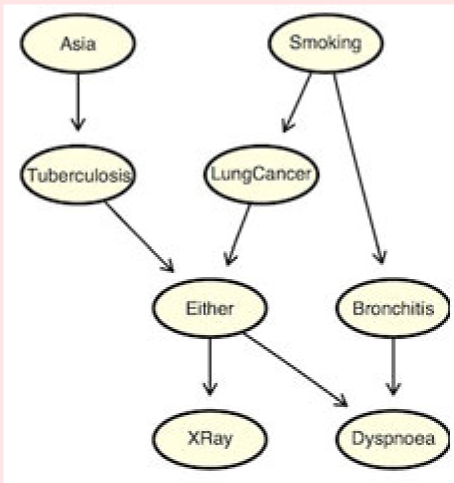
Edward → Quantum Edward

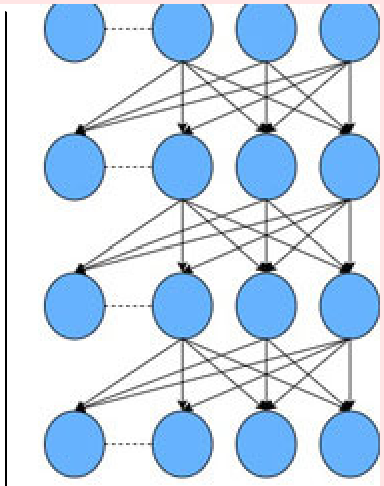
https://github.com/artiste-qb-net/Quantum_Edward

Quantum Edward at this point is just a small library of Python tools for doing classical supervised learning by Quantum Neural Networks (QNNs).

Artiste-qb.net

TensorFlow Versus TensorLayer





Directed Acyclic Graphs, Bayesian Networks	Layers, Artificial Neural Networks	
TensorFlow, Edward, PyMC	TensorLayer, PyTorch, Keras	Classical
Quantum Bayesian Networks Quantum Fog	Quantum Circuits Qubiter Quantum Edward	Quantum

Quantum Edward (Prior & Future work)

An analytical model of a QNN is entered as input into QEdward and the training is done on a classical computer, using training data already available (e.g., MNIST), and using the famous BBVI (Black Box Variational Inference) method described in Reference 1 below. The BBVI method is a mainstay of the "Edward" software library.

This first version of Quantum Edward does not do distributed computing. The hope is that it can be used as a kindergarten to learn about these techniques, and that then the lessons learned can be used to write a library that does the same thing, classical supervised learning by QNNs, but in a distributed fashion using Edward/TensorFlow on the cloud.

References

1. R. Ranganath, S. Gerrish, D. M. Blei, "Black Box Variational Inference", https://arxiv.org/abs/1401.0118

Artiste-qb.net

Quantum Edward (Strategy)

The input analytical model of the QNN is given as a sequence of gate operations for a gate model quantum computer. The hidden variables are angles by which the qubits are rotated. The observed variables are the input and output of the quantum circuit. Since it is already expressed in the qc's native language, once the QNN has been trained using QEdward, it can be run immediately on a physical gate model qc such as the ones that IBM and Google have already built. By running the QNN on a qc and doing classification with it, we can compare the performance in classification tasks of QNNs and classical artificial neural nets (ANNs).

Quantum Edward (Noisy Quantum Hardware)

Other workers have proposed training a QNN on an actual physical qc. But current qc's are still fairly quantum noisy. Training an analytical QNN on a classical computer might yield better results than training it on a qc because in the first strategy, the qc's quantum noise does not degrade the training.

Quantum Edward (Two Examples)

The first version of Quantum Edward analyzes two QNN models called NbTrolsModel and NoNbTrolsModel. These two models were chosen because they are interesting to the author, but the author attempted to make the library general enough so that it can accommodate other akin models in the future. The allowable models are referred to as QNNs because they consist of 'layers', as do classical ANNs (Artificial Neural Nets). TensorFlow can analyze layered models (e.g., ANN) or more general DAG (directed acyclic graph) models (e.g., Bayesian networks).

Quantum Edward, Docstring for class NbTrolsModel

https://github.com/artiste-qb-net/Quantum_Edward/blob/master/NbTrolsModel.py

Docstring for class NbTrolsModel

1 / 4

```
class NbTrolsModel(Model):
    """
    In this class and its twin class NoNbTrolsModel, we consider Quantum
    circuits of the following kind. Below we represent them in Qubiter ASCII
    picture notation in ZL convention, for nb=3 and na=4

    [--nb---]   [----na-----]

    NbTrols (nb Controls) model:
    |0> |0> |0> |0> |0> |0> |0>
    NOTA P(x) next
    |---|---|---|---|---|---Ry
    |---|---|---|---|---Ry--%
    |---|---|---|---Ry--%---%
    |---|---|---Ry--%---%---%
    NOTA P(y|x) next
    |---|---Ry--%---%---%---%
    |---Ry--%---%---%---%---%
    Ry--%---%---%---%---%---%
    M   M   M
```

←P(q0)
←P(q1 | q0)
←P(q2 | q1, q0)
←P(q3 | q2, q1, q0)

x = (q3, q2, q1, q0)
←product = P(x)

←P(q4 | q3, q2, q1, q0)
←P(q5 | q4, q3, q2, q1, q0)
←P(q6 | q5, q4, q3, q2, q1, q0)

y= (q6, q5, q4)
←product = P(y | x)

Chain Rule for probabilities

Artiste-qb.net

Docstring for class NbTrolsModel

2 / 4

```
NoNbTrols (no nb Controls) model:
|0> |0> |0> |0> |0> |0> |0>
NOTA P(x) next
|---|---|---|---|---|---Ry
|---|---|---|---|---Ry--%
|---|---|---|---Ry--%---%
|---|---|---Ry--%---%---%
NOTA P(y|x) next
|---|---Ry--%---%---%---%
|---Ry--|---%---%---%---%
Ry--|---|---%---%---%---%
M   M   M

A gate |---|---Ry--%---%---%---% is called an MP_Y Multiplexor,
or plexor for short. In Ref.1 (Qubiter repo at github), see Rosetta Stone
pdf and Quantum CSD Compiler folder for more info about multiplexors.

In NbTrols and NoNbtrols models, each layer of a list1 corresponds to a
single plexor. We list plexors (layers) in a list1 in order of
increasing distance between the Ry target qubit and the 0th qubit.
```

Docstring for class NbTrolsModel

3 / 4

Note that the expansion of a multiplexor into elementary gates (cnots and single qubit rotations) contains a huge number of gates (exp in the number of controls). However, such expansions can be shortened by approximating the multiplexors, using, for instance, the technique of Ref.2.

Ref.3 explains the motivation for choosing this model. This model is in fact guaranteed to fully parametrize P(x) and P(y|x).

The circuits given above are for finding a fit of both P(x) and P(y|x). However, if one wants to use a physical hardware device as a classifier, then one should omit the beginning part of the circuits (the parts that represent P(x)), and feed the input x into the first na qubits. In other words, for classifying, use the following circuits instead of the ones above:

Artiste-qb.net

Docstring for class NbTrolsModel


```
[--nb---]   [----na-----]

NbTrols (nb Controls) model:
|0> |0> |0>
|---|---Ry--%---%---%---%
|---Ry--%---%---%---%---%
Ry--%---%---%---%---%---%
M   M   M

NoNbTrols (no nb Controls) model:
|0> |0> |0>
|---|---Ry--%---%---%---%
|---Ry--|---%---%---%---%
Ry--|---|---%---%---%---%
M   M   M
```

References
----

1. https://github.com/artiste-qb-net/qubiter

2. Oracular Approximation of Quantum Multiplexors and Diagonal Unitary Matrices, by Robert R. Tucci, https://arxiv.org/abs/0901.3851

3. Code Generator for Quantum Simulated Annealing, by Robert R. Tucci, https://arxiv.org/abs/0908.1633 , Appendix B

4. "Quantum Edward Algebra.pdf", pdf included in this repo

BBVI (Theory)

R. Ranganath, S. Gerrish, D. M. Blei, "Black Box Variational Inference", https://arxiv.org/abs/1401.0118

BBVI uses ELBO maximization. Before reading the original BBVI paper, you might fortify yourself with the following background material from the Tutorials of the Edward home site:

Variational Inference
http://edwardlib.org/tutorials/variational-inference

KLqp minimization (same as ELBO maximization)
http://edwardlib.org/tutorials/klqp

KL = Kullback Liebler, ELBO = Evidence Lower Bound

BBVI (Long History)

https://en.m.wikipedia.org/wiki/Variational_Bayesian_methods

Variational Inference: A Review for Statisticians
https://arxiv.org/abs/1601.00670

Berkeley Univ. Prof. Michael Jordan (same name as famous basketball player) was first to apply Variational Inference to Bayesian nets to do medical diagnosis.

Prof. M. Jordan was thesis advisor to David Blei, who became Prof. at Columbia Univ. Prof. D. Blei was thesis advisor to Dustin Tran. So BBVI spans at least 3 generations and + 20 years