QUANTUM CIRCUITS FOR tnreason

October 20, 2025

By its central axioms, quantum mechanics of multiple qubits is formulated by tensors capturing states and discrete time evolutions. Motivated by the structural similarity, we investigate how quantum circuits can be utilized for the tensor-network based approach towards efficient and explainable AI in the tnreason framework.

We follow two main ideas:

- **Sampling of Computation-Activation Networks:** Prepare quantum states, which measurement statistics can be utilized to prepare samples from Computation-Activation Networks.
- **Quantum Circuits as Contraction providers:** Quantum circuits are contractions of multiple tensors and therefore tensor networks, and measurement probabilities are given by contractions. Here we investigate how we can exploit these as contraction provider for tnreason.

Contents

1 Circuit Encoding Schemes

We investigate here quantum pendants to the function encoding schemes used in tnreason.

- Pendant for Coordinate Encoding: **Amplitude Encoding**, storing the function value in the amplitude of an ancilla qubit.
- Pendant for Basis Encoding: **Basis Encoding Circuit**, with composition by contraction property.

Both are defined using qubit rotations (see Section 4.2 in [Nielssen, Chuang]), where the incoming qubit variable is A_{in} and the outgoing A_{out}

$$R_Y(\alpha)\left[A_{\text{in}}, A_{\text{out}}\right] := \begin{bmatrix} \cos\left(\frac{\alpha}{2}\right) & -\sin\left(\frac{\alpha}{2}\right) \\ \sin\left(\frac{\alpha}{2}\right) & \cos\left(\frac{\alpha}{2}\right) \end{bmatrix}$$

Further we use the Pauli-X:

$$\sigma_1\left[A_{\text{in}}, A_{\text{out}}\right] := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

1.1 Amplitude Encoding

We define the angle preparing function on $p \in [0, 1]$ by

$$h(p) = 2 \cdot \cos^{-1}\left(\sqrt{1-p}\right).$$

For any $p \in [0, 1]$ we then have

$$\langle \epsilon_0\left[A_{\text{in}}\right], R_Y\left(h(p)\right)\left[A_{\text{in}}, A_{\text{out}}\right]\rangle\left[A_{\text{out}}\right] = \begin{bmatrix} \sqrt{1-p} \\ \sqrt{p} \end{bmatrix}.$$

Definition 1 (Amplitude Encoding). *Given a function*

$$q : \bigtimes_{l \in [p]} [2] \to [0, 1]$$

its amplitude encoding is the unitary tensor $\mathcal{V}^\tau\left[A_{\text{in}}, A_{\text{out}}, Y_{\text{in}, [p]}, Y_{\text{out}, [p]}\right]$ *which satisfies for each* $y_{[p]}$

$$\mathcal{V}^\tau\left[A_{\text{in}}, A_{\text{out}}, Y_{\text{in}, [p]}, Y_{\text{out}, [p]}\right] = \sum_{y_{[p]}} \epsilon_{y_{[p]}}\left[Y_{\text{in}, [p]}\right] \otimes \epsilon_{y_{[p]}}\left[Y_{\text{in}, [p]}\right] \otimes R_Y\left(h(q(y_{[p]}))\right)\left[A_{\text{in}}, A_{\text{out}}\right].$$

Tensors $\tau\left[Y_{[p]}\right]$ with non-negative coordinates can be encoded after dividing them by their maximum, that is the amplitude encoding of the function

$$q(y_{[p]}) = \frac{\tau\left[Y_{[p]} = y_{[p]}\right]}{\max_{\tilde{y}_{[p]}} \tau\left[Y_{[p]} = \tilde{y}_{[p]}\right]}$$

When the maximum of the tensor is not known, it can be replaced by an upper bound (reducing the acceptance rate of the rejection sampling).

1.1.1 Encoding of directed tensors

Following the schemes in **?**, we can prepare any non-negative tensors by a sequence of controlled rotations. To this end, one iterates over the states of the incoming variables, and performs a controlled rotation on the outgoing variable, where the angle is given by the value of the tensor at the incoming state. This generalizes the basis encoding scheme, which demands boolean tensors
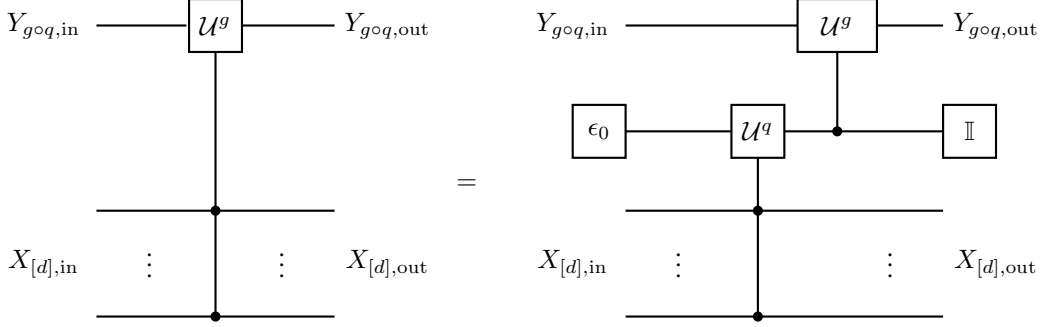
Figure 1: Exploitaition of Decomposition sparsity in basis encoding circuits.

1.2 Basis encoding

We here suggest a quantum pendant to basis encodings (see Chapter Basis Calculus), which has the decomposition by contraction property.

Definition 2 (Basis Encoding Circuit)*. Given a boolean function $q : \bigtimes_{k \in [d]}[2] \to [2]$ the basis encoding circuit is the unitary tensor*

$$
\begin{aligned}
\mathcal{U}^q &\left[Y_{\text{in},q}, Y_{\text{out},q}, X_{\text{in},[d]}, X_{\text{out},[d]} \right] \\
&= \sum_{x_{[d]} \in \bigtimes_{k \in [d]}[m_k] \,:\, q(x_{[d]})=1} \sigma_1 \left[Y_{\text{in},q}, Y_{\text{out},q} \right] \otimes \epsilon_{x_{[d]}} \left[X_{\text{in},[d]} \right] \otimes \epsilon_{x_{[d]}} \left[X_{\text{out},[d]} \right] \\
&+ \sum_{x_{[d]} \in \bigtimes_{k \in [d]}[m_k] \,:\, q(x_{[d]})=0} \delta \left[Y_{\text{in},q}, Y_{\text{out},q} \right] \otimes \epsilon_{x_{[d]}} \left[X_{\text{in},[d]} \right] \otimes \epsilon_{x_{[d]}} \left[X_{\text{out},[d]} \right] \,.
\end{aligned}
$$

Notice, that $U^{\neg} = \text{CNOT}$, which is obvious from $\mathbb{I}\left[Y_{\text{in}}, Y_{\text{out}}\right] - \delta\left[Y_{\text{in}}, Y_{\text{out}}\right]$ being the Pauli-X gate (not to be confused with X denoting distributed variables here). The basis encoding circuit is therefore a generalized controlled NOT gate, where the control is by a boolean function.

Functions with multiple output variables, i.e. $q : \bigtimes_{k \in [d]}[2] \to \bigtimes_{l \in [p]}[2]$, can be encoded image coordinate wise as a concatenation of the respective circuits.

1.2.1 Composition by Contraction - Exploiting Decomposition sparsity

The decomposition by contraction property of basis encodings is now a composition of circuits property, as stated in the next lemma.

Lemma 1. *We have for functions $q : \bigtimes_{k \in [d]}[2] \to \bigtimes_{l \in [p]}[2]$, $g : \bigtimes_{l \in [p]}[2] \to \bigtimes_{s \in [r]}[2]$ (see Figure ??)*

$$
\begin{aligned}
\mathcal{U}^{g \circ q} \left[Y_{\text{in},g \circ q}, Y_{\text{out},g \circ q}, X_{\text{in},[d]}, X_{\text{out},[d]} \right] = \Big\langle &\epsilon_0 \left[Y_{\text{in},q} \right], \\
&\mathcal{U}^q \left[X_{\text{in},[d]}, X_{\text{out},[d]}, Y_{\text{in},q}, Y_{\text{out},q} \right], \\
&\mathcal{U}^g \left[Y_{\text{out},q}, Y_{\text{out,aux}}, Y_{\text{in},g \circ q}, Y_{\text{out},g \circ q} \right] \Big\rangle \left[X_{\text{in},[d]}, X_{\text{out},[d]}, Y_{\text{in},g \circ q}, Y_{\text{out},g \circ q} \right] \,.
\end{aligned}
$$

Note, that the variables $Y_{\text{out,aux}}$ are auxiliar and not left open in the contraction. This amounts to not measuring them in a computational basis measurement.

1.2.2 Construction for mod2-basis+ CP decompositions - Exploiting Polynomial sparsity

Concatenating two basis encoding circuits is the basis encoding circuit of their mod2 sum.

A basis+ elementary function can be encoded by a single controlled NOT operation with auxiliary X qubits.

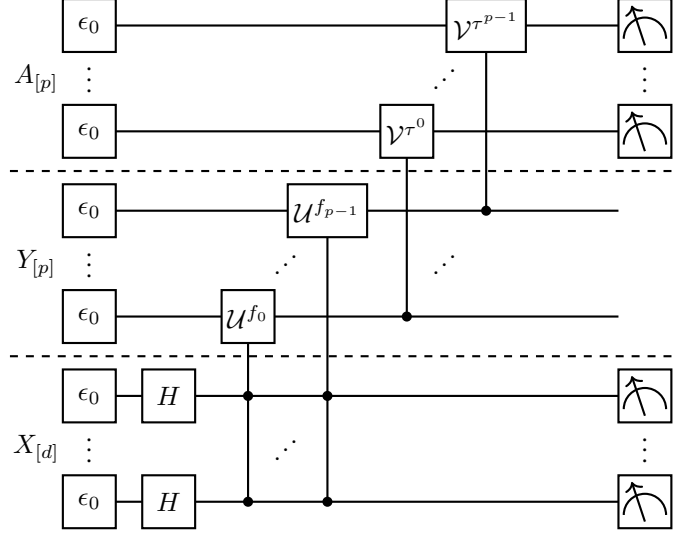This motivates the mod2-basis+ CP decomposition of tensors.

Figure 2: Quantum Circuit to reproduce a Computation-Activation Network (with elementary activation) by rejection sampling. We measure the distributed qubits $X_0, \ldots, X_{d-1}$ and the ancilla qubits $A_{[p]}$ and reject all samples, where an ancilla qubit is measured as 0.

Definition 3. *Given a boolean tensor τ, a mod2-basis+ CP decomposition is a collection $\mathcal{M}$ of tuples (A, x_A) with such that for any $x_{[d]} \in \bigtimes_{k \in [d]}[m_k]$*

$$\tau\left[X_{[d]} = x_{[d]}\right] = \bigoplus_{(A, x_A) \in \mathcal{M}} \langle \epsilon_{x_A}[X_A] \rangle \left[X_{[d]} = x_{[d]}\right] .$$

Using that basis CP decompositions are a special case of basis+ CP decompositions, we get the following rank bound.

Lemma 2. *The mod2-basis+ CP rank is bounded by the basis CP rank.*

Proof. Use $A = [d]$, and x_A to each supported state. Then the mod2-sum is a usual sum and the basis CP decomposition is also a mod2-basis+ CP decomposition. $\square$

This shows in particular, that any propositional formula can be represented by a mod2-basis+ CP decomposition.

1.2.3 Preparation by fine and coarse structure

Having a mod2-basis+ CP decomposition of rank r to a connective, we need r controlled NOT gates to prepare the basis encoding. Given a syntactical decomposition of a boolean statistics, we prepare the basis encoding as a circuit with:

- **Fine Structure:** Represent each logical connective based on its mod2-basis+ CP decomposition, as a concatenation of basis encoding circuits with the same variables.
- **Coarse Structure:** Arrange the logical connective representing circuits according to the syntactical hypergraph, where parent head variables appear as distributed variables at their children.

2 Sampling from Computation-Activation Networks as Quantum Circuits

tnreason provides tensor network representations of knowledge bases and exponential families following a Computation Activation architecture. Here are some ideas to utilize quantum circuits for sampling Computation-Activation Networks.

We can produce Q-samples for ancilla augmented Computation-Activation Networks using basis circuit encodings of the computation cores and ancilla encodings of the activation cores.

2.1 Generic Q-samples

In general, we define Q-samples to be quantum states, which measured in the computational basis reproduce a given probability distribution.

Definition 4 (Q-sample). *Given a probability distribution* $\mathbb{P} : \bigtimes_{k \in [d]}[2] \to \mathbb{R}$ *(i.e.* $\langle \mathbb{P} \rangle [\varnothing] = 1$ *and* $0 \prec \mathbb{P}$) *its q-sample is*

$$\psi^{\mathbb{P}}[X_0, \ldots, X_{d-1}] = \sum_{x_{[d]} \in \bigtimes_{k \in [d]}[m_k]} \sqrt{\mathbb{P}[X_{[d]} = x_{[d]}]} \cdot \epsilon_{x_{[d]}}[X_0, \ldots, X_{d-1}] \ .$$

In **?** the amplitude encoding is called q-sample. It prepares a scheme to realize property 1 (purity) + 2 (q-sampling) of a qpdf, but fails to realize property 3 (q-stochasticity). The q-sample can be prepared for Bayesian Networks, where each child qubit is prepared densely by C-NOTs conditioning on parent qubits.

Q-samples can be prepared by amplitude encodings acting on uniform quantum states (Hadamard gates acting on ground state).

Doing rejection sampling on the ancilla qubit corresponds with sampling from the normalized contraction with the activation tensor.

Lemma 3. *Given a distribution* $\mathbb{P}[X_0, \ldots, X_{d-1}]$, *we construct a circuit preparing its q-sample and add the ancilla encoding of a tensor* $\tau[X_0, \ldots, X_{d-1}]$. *The rejection sampling scheme, measuring the ancilla qubit and the* $X_0, \ldots, X_{d-1}$ *qubits, rejecting the ancilla qubit measured as* 0, *prepares samples from the distribution*

$$\langle \mathbb{P}[X_0, \ldots, X_{d-1}], \tau[X_0, \ldots, X_{d-1}] \rangle [X_0, \ldots, X_{d-1}|\varnothing] \ .$$

For more flexible sampling schemes of Computation-Activation Networks we need to introduce ancilla qubits.

Definition 5 (Ancilla Augmented Distribution). *Let* $\mathbb{P}[X_{[d]}]$ *be a probability distribution over variables* $X_0, \ldots, X_{d-1}$. *Another joint distribution of* $X_0, \ldots, X_{d-1}$ *and ancilla variables* $A_{[p]}$ *is called an ancilla augmented distribution, if*

$$\mathbb{P}[X_0, \ldots, X_{d-1}|A_{[p]} = \mathbb{I}[[p]]] = \mathbb{P}[X_{[d]}] \ .$$

Sampling from the distribution can be done by rejection sampling on the ancilla augmented distribution, measuring all variables and rejecting all samples where an ancilla variable is 0.

Given an augmented Q-sample of a distribution, we can prepare samples from the distribution by rejection sampling, measuring all variables $X_0, \ldots, X_{d-1}$ and $A_{[p]}$ and rejecting all samples where an ancilla qubit is measured as 0.

When sampling from probability distributions, we can use these samples to estimate probabilistic queries. Building on such particle-based inference schemes, we can perform various inference schemes for Computation-Activation Networks, such as backward inference and message passing schemes.

2.2 Preparation of Ancilla Qubits

Using the ancilla encoding of activation cores!

We introduce a ancilla qubit, which stores in its coefficient to the first state the probability of the configuration (this is the ancilla encoding!). When we have a probability tensor, this can be prepared, since all values are in $[0, 1]$. For the rejection sampling, only the quotients of the values are important, we can therefore scale the value by a scalar such that the mode is 1. The value qubit is initialized by the zeroth one hot encoding ($|0\rangle$) and rotated by a controlled rotation gate, which is controlled by the variable qubits.

To prepare a distribution with sufficient statistics, we prepare the statistic qubits by a basis encoding circuit and do controlled rotations on the value qubit controlled by the statistic qubits.

Maximum entropy distribution under moment constraints are a special case of distributions with sufficient statistics. In many cases (precisely by mean parameters on the interior of cube faces), they have a factored representation by the features. It is unclear, how to exploit this factored representation in the preparation of the controlled value rotations.

Open Question: How to exploit the factored representation of maximum entropy distributions in the preparation of the value qubit? - By a collection of ancilla qubits on each statistic, and doing amplitude amplification on the state of all qubits 1.

2.3 Polynomial Sparsity

This is exactly the decomposition of boolean polynomials into monomials (which can include 1-x terms)! Decompositions is called into terms (products of x or (1-x) factors), and minterms if all variables appear.

Each monomial can be prepared by a multiple-controlled NOT gate, where the control qubits are the affected variables and the target qubit is the value qubit. When we sum monomials wrt modulus 2 calculus, then the preparation is a sequence of such circuits. In such way, we can prepare the basis encoding circuit to any propositional formula. This encoding strategy exploits a modified (by mod2 calculus) polynomial sparsity.

2.4 Decomposition Sparsity

When having a syntactical decomposition of a propositional formula, we can iteratively apply the basis encoding circuit decomposition theorem and prepare each connective by a circuit. We can decompose any propositional formula into logical connectives and prepare to each a modulus 2 circuit implementation. This works, when the target qubit of one connective is used as a value qubit of another.

2.5 Quantum Rejection Sampling

Note, that the variable qubits are uniformly distributed when only the computation circuit is applied. When sampling the probability distribution, we need the value qubit to be in state 1 in order for the sample to be valid. Any other states will have to be rejected.

Classically, this can be simulated in the same way: Just draw the variables from uniform, calculate the value qubit by a logical circuit inference and accept with probability by the computed value.

For this procedure to be more effective (and in particular not having an efficient classical pendant), we need amplitude amplification on the value qubit. This can provide a square root speedup in the complexity compared with classical rejection sampling.

Open Question: Is there a way to avoid amplitude amplification and use a more direct circuit implementation of the activation network? - Cannot be the case, when the encoding is determined by the activation tensor alone: Needs to use the computated statistic as well.

3 Sampling from proposal distributions

We can prepare basis circuit encodings to selection augmented formulas, in this way introducing formula selecting networks.

Idea for an inductive reasoning scheme: Prepare a q-sample from the empirical distribution and the current distribution. Then prepare the basis circuit encodings, where the selection variables are shared and the distributed variables assigned to the prepared samples. Now, the ancilla qubits can be designed to ϵ_1 and ϵ_0 accordingly. The rejection sampling scheme on both ancillas being 1 and the measurement of L prepares then the distribution

$$\left\langle \left\langle \mathbb{P}^D\left[X_{[d]}\right], \sigma^{\mathcal{F}}\left[X_{[d]}, L\right]\right\rangle [L], \left\langle \tilde{\mathbb{P}}\left[X_{[d]}\right]\right\rangle [L]\right\rangle [L|\varnothing]$$

That is, the probability of selecting l is proportional to
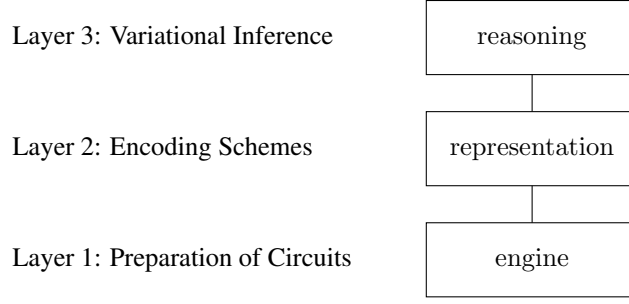
$$\mu_D\left[L = l\right] \cdot \left(1 - \tilde{\mu}\left[L = l\right]\right)$$

and thus prefers formulas, which have a large empirical mean, but a small current mean.

Open Question: Since the distribution is "similar" to $\exp\left[\mu_D\left[L = l\right] - \tilde{\mu}\left[L = l\right]\right)]$ (terms appear in Taylor of first order), can we tune the distribution with an inverse temperature parameter β?

4 Implementation

The introduced quantum circuit preparation schemes have been implemented in the python package tnreason, which consists in three layers:

Layer 3: Variational Inference	reasoning
Layer 2: Encoding Schemes	representation
Layer 1: Preparation of Circuits	engine

A   Comparing tensor networks and quantum circuits

First of all, we need to extend to complex tensors, which are maps

$$\tau : \bigtimes_{k \in [d]} [2] \to \mathbb{C}$$

with image in $\mathbb{C}$ instead of $\mathbb{R}$ as in the report.

A coarse comparison of the nomenclature used for quantum circuits and tensor networks:

Quantum Circuit	**Tensor Network**
Qubit	Boolean Variable
Quantum Gate	Unitary Tensor
Quantum Circuit	Tensor Network on a graph

Some constraints appear for a tensor network to be a quantum circuit

- **Unitarity of each gate:** That is the variables of each tensor are bipartite into sets A^{in} and A^{out} of same cardinality and the basis encoding with respect to this bipartition, that is

$$T_{\mathrm{in}\to\mathrm{out}}[X_{\mathrm{in}}, X_{\mathrm{out}}] : \bigotimes_{k \in A^{\mathrm{in}}} \mathbb{C}^2 \to \bigotimes_{k \in A^{\mathrm{out}}} \mathbb{C}^2 \,,$$

  is a unitary map, that is

$$(T_{\mathrm{in}\to\mathrm{out}})^H \circ (T_{\mathrm{in}\to\mathrm{out}}) = \left\langle T_{\mathrm{in}\to\mathrm{out}}[X_{\mathrm{in}}, Y], \overline{T}_{\mathrm{in}\to\mathrm{out}}[Y, X_{\mathrm{out}}] \right\rangle [X_{\mathrm{out}}, X_{\mathrm{in}}] = \delta [X_{\mathrm{out}}, X_{\mathrm{in}}] \,.$$

- **Incoming-Outgoing structure:** Variable appear at most once as incoming and at most once as outgoing variables. Those not appearing as outgoing (respectively as incoming) are the input and the output variables of the whole circuit.

- **Acyclicity:** Incoming and outgoing variables of each tensor core provide a direction of each edge tensor. With respect to this directionality the graph underlying the tensor network has to be acyclic.

The unitary tensors can be aligned layerwise, if and only if the last two assumption hold, i.e. the directed graph is acyclic and each variable appears at most once as an incoming and at most once as an outgoing variable.

B   POVM measurements as contractions

The main difficulty of using quantum circuits as contraction providers is that we can only extract information through measurements. Therefore measurement is the only way to execute contractions of the circuit, which come with restrictions when interested in contraction with open variables.

The most general measurement formalism is through a POVM, a set $\{E_y : y \in [r]\}$ of positive operators with

$$\sum_{y \in [r]} E_y = I$$

Measuring a pure state $|\psi\rangle$ We then get outcome m with probability

$$\langle \psi | E_y | \psi \rangle$$

We define a measurement variable Y taking indices $y \in [r]$ and a measurement tensor

$$E[Y, X_{\text{in}}, X_{\text{out}}]$$

with slices

$$E[Y = y, X_{\text{in}}, X_{\text{out}}] = E_y \, .$$

Repeating the measurement asymptotically on a state $|\psi\rangle$ prepared by a quantum circuit $\tau^{\mathcal{G}}$ acting on the trivial start state $\mathbb{I}$, we denote the measurement outcome by y^j. In the limit $m \to \infty$ we get almost surely

$$\frac{1}{m} \sum_{j \in [m]} \epsilon_{y^j} [Y] \to \left\langle \tau^{\mathcal{G}}[X_{\text{in}}], E[Y, X_{\text{in}}, X_{\text{out}}], \tau^{\tilde{\mathcal{G}}}[X_{\text{out}}] \right\rangle [Y] \, .$$

POVMs to computational basis measurements of subsets of qubits are constructed as products with delta tensors on the non-measured qubits.