Encoding

Marten Teitsma

March 8, 2024

Encoding

Quantum computing solves some problems faster than classical computing

- exponential (factoring large prime numbers)
- $\sqrt{n}$ (unstructured search)
- heuristic (e.g. quantum machine learning)

Encoding needed to fit classical data to quantum computers. Runtime complexity of data encoding depends on:

- the encoding routine
- the data itself

A quantum feature map encodes classical input to quantum states.
Six encoding patterns

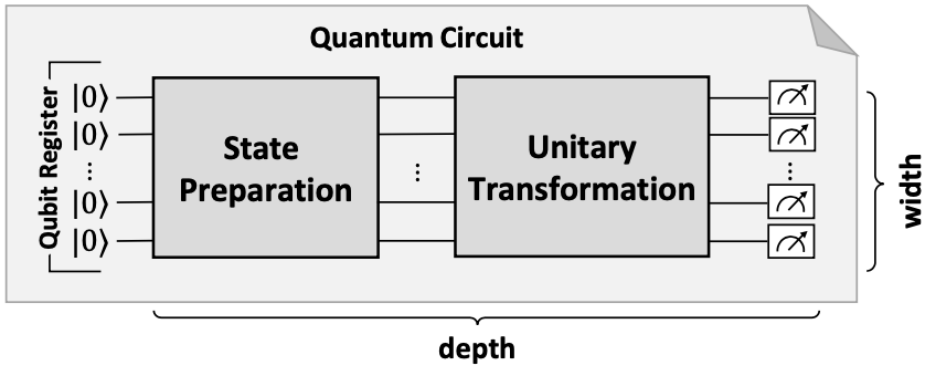


Figure: Typical structure of a quantum calculation that is represented as a quantum circuit which has a defined depth and width. From: [1]

1. The qubits of the register are initialized as $|0\rangle$
2. An initial state is prepared, including loading data which (as a result) is encoded in the state of the register
3. Quantum gates perform unitary transformations on the state of the register
4. One or multiple qubits are measured as the result of this computation

The quantum circuit is characterized by its

- depth which is the number of sequential executable gates
- width that equals the number of required qubits

Patterns for encoding

Every data encoding is essentially a trade-off between three major forces:

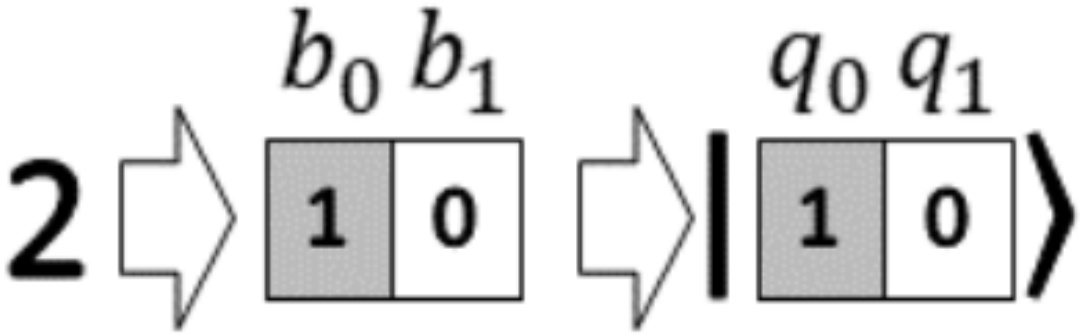
1. the number of qubits needed for the encoding should be minimal because current devices are of intermediate size and thus only contain a limited number of qubits
2. the number of parallel operations needed to realize the encoding should be minimal to minimize the width of the quantum circuit - ideally, the loading routine is of constant or logarithmic complexity
3. the data must be represented in a suitable manner for further calculations, e.g., arithmetic operations

Basis encoding I

Intent: Represent data elements in a quantum computer in order to perform calculations

Context: A quantum algorithm requires numerical input data X for further calculations

Solution: The main idea for this encoding is to use the computational basis $|0\ldots00\rangle, |0\ldots01\rangle, |1\ldots11\rangle$ to encode the input data: An input number x is approximated by a binary format $x := b_{n-1}\ldots b_1 b_0$ which is then turned into the corresponding basis vector $|x\rangle := |b_{n-1}\ldots b_1 b_0\rangle$.

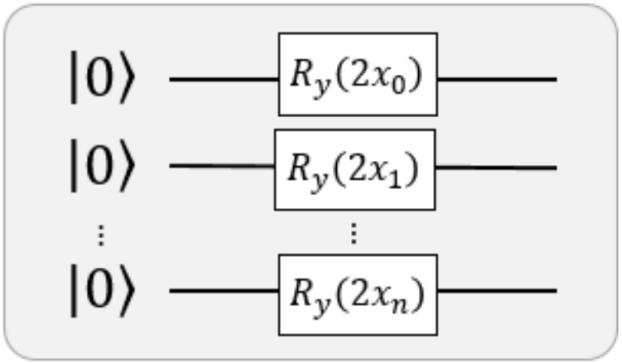


Basis encoding II

This encoding can be categorized as digital encoding because it is suitable for arithmetic computations For input numbers which are approximated by I digits, I qubits are needed for its representation. To realize this encoding, the initial $|0\rangle$ state of qubits that represent a '1' digit must be flipped into $|1\rangle$. For one qubit, this can be done by a single operation, and thus, this encoding can be prepared in linear time.

Angle encoding I

Intent: Represent each data point by a separate qubit
Context: An algorithm requires an efficient encoding schema to be able to perform as many operations as possible within the decoherence time after the data has been loaded.
Solution: As a first step, each data point of the input is normalized to the interval $[0, 2\pi]$. To encode the data points, a rotation around the y-axis is used for which the angle depends on the value of the normalized data point.



Angle encoding II

This creates the following separable state:

$$|\phi\rangle = \begin{bmatrix} cosx_0 \\ sinx_0 \end{bmatrix} \otimes \begin{bmatrix} cosx_1 \\ sinx_1 \end{bmatrix} \otimes \cdots \otimes \begin{bmatrix} cosx_n \\ sinx_n \end{bmatrix} \tag{1}$$

It can easily be seen that one qubit is needed per data point which is not optimal. To load the data, the rotations on the qubits can be performed in parallel, thus, the depth of the circuit is optimal.

Amplitude encoding I

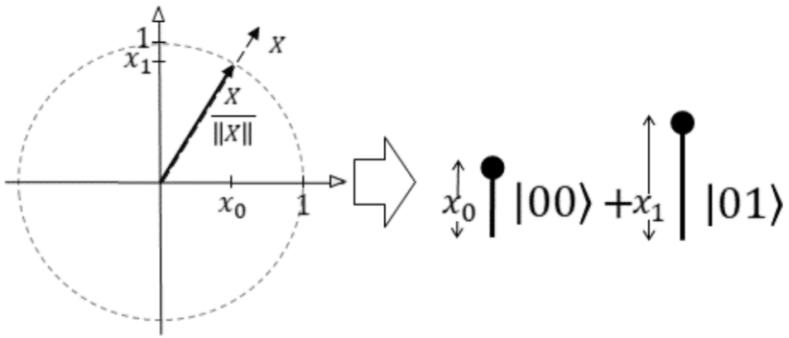
Intent: Encode data in a compact manner that do not require calculations
Context: A numerical input data vector $(x_0, \ldots, x_{n-1})^T$ must be encoded for an algorithm.
Solution: Use amplitudes to encode the data. As the squared moduli of the amplitudes of a quantum state must sum up to 1, the input vector needs to be normalized to length 1. This is illustrated in the pattern sketch for a 2-dimensional input vector that contains 2 data points. To associate each amplitude with a component of the input vector, the dimension of the vector must be equal to a power of two because the vector space of an n qubit register has dimension 2^n. If this is not the case, the input vector can be padded with additional zeros to increase the dimension of it.

Amplitude encoding II

Using a suitable state preparation routine, the input vector is encoded in the amplitudes of the quantum state as follows:

$$|\phi\rangle = \sum_{i=0}^{n-1} x_i |i\rangle \tag{2}$$

As the amplitudes depend on the data, the process of encoding the data (but not the encoding itself) is often referred to as arbitrary state preparation.



Amplitude encoding III

A data input vector of length I can be represented by $[\log_2(I)]$ qubits - this is indeed a very compact representation. For an arbitrary state represented by n qubits (which represents 2^n data values), it is known that at least 2^n parallel operations are needed. It must be noted that if the output is also encoded in the amplitude, multiple measurements must be taken to obtain a good estimate of the output result. The number of measurements scales with the number of amplitudes -as n qubits contain 2^n amplitudes, this is costly.

When 8 data are stored in 3 qubits the result is something like:

$$\frac{1}{\sqrt{8}}(\alpha_0 |000\rangle + \alpha_1 |001\rangle + \alpha_2 |010\rangle + \alpha_3 |011\rangle + \alpha_4 |100\rangle + \alpha_5 |101\rangle + \alpha_6 |110\rangle + \alpha_7 \tag{3}$$

This gives a uniform superposition for 3 qubits. But our goal is not a uniform distribution of amplitudes. How do we get the data normalised on 1 (Borne's rule)?

$$\frac{1}{\sum_{i=0}^{n} \alpha_i}(\alpha_0 |000\rangle + \alpha_1 |001\rangle + \alpha_2 |010\rangle + \alpha_3 |011\rangle + \alpha_4 |100\rangle + \alpha_5 |101\rangle + \alpha_6 |110 \tag{4}$$

Assignment

Create an example for all the three discussed encoding patterns and create a uniform superposition.
Quantum patterns


M. Weigold, J. Barzen, F. Leymann, and M. Salm.
Expanding data encoding patterns for quantum algorithms.
In *2021 IEEE 18th International Conference on Software Architecture Companion (ICSA-C)*, pages 95–101. IEEE, 2021.