Measurements and Expectation Values

The Singularity

February 2020

1 Introduction

In this document we will look more closely at measurements.

1.1 Pauli Gates

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

1.2 The "CNOT" Gate

$$\mathbf{CNOT} = \begin{pmatrix} I & 0 \\ 0 & X \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

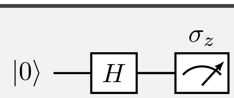
1.3 The Hadamard Gate

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

2 More Linear Algebra from Circuit Diagrams

Convert the following diagrams into linear algebra, i.e. change the circuit diagrams to matrices operating on vectors.

2.1 Circuit Diagram 1



Solution. In this circuit we are trying to calculate the **expectation value** $\langle \sigma_z \rangle$, after the Hadamard gate has operated on the prepared state $|0\rangle$. To do this, first calculate the action of the Hadamard gate on the qubit:

$$H|0\rangle = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Next, we compute the expectation value, using the Pauli-Z gate, i.e.

$$\langle \psi | \sigma_z | \psi \rangle,$$

where

$$\langle\psi| = \begin{pmatrix} 1, & 1 \end{pmatrix}.$$

So, we have

$$\langle\psi|\sigma_z|\psi\rangle = \begin{pmatrix} 1, & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 0.$$

■

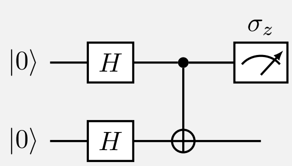
We can compute this using PennyLane via the following code:

```
dev1 = qml.device("default.qubit", wires=1)

@qml.qnode(dev1)
def circuit():
    qml.Hadamard(wires=0)
    return qml.expval(qml.PauliZ(0))

print(circuit())
```

2.2 Circuit Diagram 2



Solution. Referring to the document from the Qubit Rotation Tutorial, we need to compute the following,

$$H|0\rangle \otimes H|0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \tag{1}$$

Next, we apply the CNOT gate,

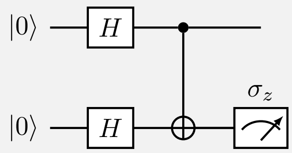
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \\ 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} = |\psi\rangle.$$

Now that we have the state $|\psi\rangle$, we compute

$$\langle\sigma_z \otimes I\rangle = \langle\psi|\sigma_z \otimes I|\psi\rangle.$$

■

2.3 Circuit Diagram 3



Solution. The previous computation provides an almost complete solution, except this time we wish to compute

$$\langle I \otimes \sigma_z \rangle = \langle \psi | \sigma_z \otimes I | \psi \rangle.$$

■

The following PennyLane code will compute and print the results of the previous two circuits, i.e. it will compute:

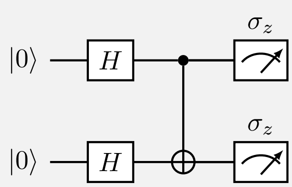
$$\langle \sigma_z \otimes I \rangle, \quad \text{and} \quad \langle I \otimes \sigma_z \rangle.$$

```
dev = qml.device("default.qubit", wires=2)

@qml.qnode(dev)
def circuit():
    qml.Hadamard(wires=0)
    qml.Hadamard(wires=1)
    qml.CNOT(wires=[0, 1])
    return qml.expval(qml.PauliZ(0)), qml.expval(qml.PauliZ(1))

print(circuit())
```

2.4 Circuit Diagram 4



Solution. Using the state computed in the previous two circuits:

$$|\psi\rangle = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \\ 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}$$

we want to compute:

$$\langle \sigma_z \otimes \sigma_z \rangle = \langle \psi | \sigma_z \otimes \sigma_z | \psi \rangle.$$

■

The following PennyLane code will compute the expectation value:

$$\langle \sigma_z \otimes \sigma_z \rangle.$$

```
dev = qml.device("default.qubit", wires=2)

@qml.qnode(dev)
def circuit():
    qml.Hadamard(wires=0)
    qml.Hadamard(wires=1)
    qml.CNOT(wires=[0, 1])
    return qml.expval(qml.PauliZ(0) @ qml.PauliZ(1))

print(circuit())
```