PennyLane 101: Order Matters [100 points]

Version: 1

PennyLane 101

The **PennyLane 101** challenges will introduce quantum computing concepts with PennyLane. Whether you're coming from an advanced quantum computing background, or you've never evaluated a quantum circuit before, these challenge questions will be a great start for you to learn how quantum computing works using PennyLane. Beyond these five questions in this category, there are well-developed demos and tutorials on the PennyLane website that are a good resource to fall back on if you are stuck. We also strongly recommend consulting the PennyLane documentation to see an exhaustive list of available gates and operations!

Problem statement [100 points]

In this challenge question, you need to construct two circuits that perform two rotations on one qubit/wire and then measure the expectation value of the `qml.PauliX` quantum operator. The two circuits you need to implement are pictured in Figure 1.

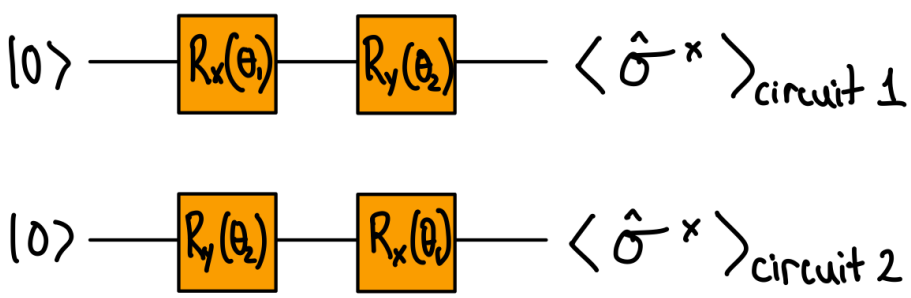

Figure 1: The circuits you need to implement.

The provided template file `order_matters_template.py` contains a function `compare_circuits` that you need to complete. Specifically, the completed `compare_circuits` function should:

- Define a quantum device.

- Create two separate quantum functions that define the two circuits in Figure 1.
- Process the results from both circuits.

The circuits will do the following:

- Circuit 1: Rotate the qubit via the gate $R_x(\theta_1)$ (`qml.RX`) then via the gate $R_y(\theta_2)$ (`qml.RY`), then output $\langle \hat{\sigma}^x \rangle_{\text{circuit 1}}$ (using `qml.expval(qml.PauliX(0))`).
- Circuit 2: Rotate the qubit via the gate $R_y(\theta_2)$ (`qml.RY`) then via the gate $R_x(\theta_1)$ (`qml.RX`), then output $\langle \hat{\sigma}^x \rangle_{\text{circuit 2}}$ (using `qml.expval(qml.PauliX(0))`).

Make your `compare_circuits` function return the absolute value of the difference between the circuit outputs: $|\langle \hat{\sigma}^x \rangle_{\text{circuit 1}} - \langle \hat{\sigma}^x \rangle_{\text{circuit 2}}|$.

Input

- `list(float)`: The angles θ_1 and θ_2 in that order.

Output

- `float`: The absolute value of the difference between the circuit outputs.

Acceptance Criteria

In order for your submission to be judged as "correct":

- The outputs generated by your solution when run with a given `.in` file must match those in the corresponding `.ans` file to within the `0.0001` tolerance specified below. To clarify, your answer must satisfy

$$\text{tolerance} \geq \left| \frac{\text{your solution} - \text{correct answer}}{\text{correct answer}} \right|.$$

- Your solution must take no longer than the `60s` specified below to produce its outputs.

You can test your solution by passing the `#.in` input data to your program as stdin and comparing the output to the corresponding `#.ans` file:

```
python3 {name_of_file}.py < 1.in
```

WARNING: Don't modify the code outside of the `# QHACK #` markers in the template file, as this code is needed to test your solution. Do not add any print statements to your solution, as this will cause your submission to fail.

Specs
Tolerance: **0.0001** Time limit: **60 s**

Version History

Version 1: Initial document.