SIGN OUT

QHack

Quantum Coding Challenges

RANK TEAM CHALLENGES SUBMISSIONS SUPPORT

Jump to code — Collapse text

4. Product Management

0 points

Welcome to the QHack 2023 daily challenges! Every day for the next four days, you will receive two new challenges to complete. These challenges are worth no points — they are specifically designed to get your brain active and into the right mindset for the competition. You will also learn about various aspects of PennyLane that are essential to quantum computing, quantum machine learning, and quantum chemistry. Have fun!

Tutorial #4 — Product states

The Coding Challenge Competition is now closed, but you are welcome to continue working on the challenges until QHack ends on Feb 28 @5pm ET.

×

Given a multi-qubit *pure* state (i.e., does not need to be described by a density operator), the presence of entanglement boils down to whether or not the state is a *product* state. Given a two-qubit state where the qubits are labelled by A and B, a general *pure* quantum state can be written as

$$|\psi\rangle_{AB} = \sum_{i,j} c_{ij}|i\rangle_A \otimes |j\rangle_B.$$

$|\psi\rangle_{AB}$ is said to be a *product* state for subsystems A and B if it can be written as a tensor product

$$|\psi\rangle_{AB} = |\psi\rangle_A \otimes |\psi\rangle_B.$$

For example, the well-known Bell states cannot be written as product states between the two qubits.

Your job is to create a function that can tell whether or not a pure state can be written as a product state between a subsystem and its complement (e.g., if A is the subsystem, then $B = \bar{A}$, meaning that system B is the set of qubits that are not in A).

Challenge code

In the code below, you are given a function called `is_product`. This function will output `"yes"` or `"no"` correspondingly. **You must complete this function.**

Here are some helpful resources:

- Separable quantum states
- `qml.density_matrix`

Input

As input to this problem, you are given:

The Coding Challenge Competition is now closed, but you are welcome to continue working on the challenges until QHack ends on Feb 28 @5pm ET.

- `subsystem` (`list(int)`): the subsystem that defines the subsystem of qubits A and $B = \bar{A}$. I.e., the two groups of qubits that you will determine if a state can be written as a product state.
- `wires` (`list(int)`): the wire labels associated to the qubit state of interest.

Output

This code must output `"yes"` or `"no"` if the state $|\psi\rangle_{AB}$ is a product state (with respect to A and B).

If your solution matches the correct one, the output will be `"Correct!"` Otherwise, you will receive a `"Wrong answer"` prompt.

Good luck!

Code

Help

```python
import json
import pennylane as qml
import pennylane.numpy as np
```

```python
def is_product(state, subsystem, wires):
    """Determines if a pure quantum state can be written as a product sta
    a subsystem of wires and their compliment.

    Args:
        state (numpy.array): The quantum state of interest.
        subsystem (list(int)): The subsystem used to determine if the sta
        wires (list(int)): The wire/qubit labels for the state. Use these

    Returns:
        (str): "yes" if the state is a product state or "no" if it isn't
    """

```

```python
    # Put your solution here #
    return

```

The Coding Challenge Competition is now closed, but you are welcome to continue working on the challenges until QHack ends on Feb 28 @5pm ET.

```python
# These functions are responsible for testing the solution.
def run(test_case_input: str) -> str:
    ins = json.loads(test_case_input)
    state, subsystem, wires = ins
    state = np.array(state)
    output = is_product(state, subsystem, wires)
    return output

def check(solution_output: str, expected_output: str) -> None:
    assert solution_output == expected_output

```

```python
test_cases = [['[[0.707107, 0, 0, 0.707107], [0], [0, 1]]', 'no'], ['[
```

```python
for i, (input_, expected_output) in enumerate(test_cases):
    print(f"Running test case {i} with input '{input_}'...")

    try:
        output = run(input_)

    except Exception as exc:
        print(f"Runtime Error. {exc}")

    else:
        if message := check(output, expected_output):
            print(f"Wrong Answer. Have: '{output}'. Want: '{expected_out

        else:
            print("Correct!")
```

Copy all

Submit

Open Notebook

Reset

The Coding Challenge Competition is now closed, but you are welcome to continue working on the challenges until QHack ends on Feb 28 @5pm ET.