Evaluation Sommative_V2306

June 8, 2023

UE 4268 - Examen (Evalution sommative) 2023

Noms et prenoms, matricule et email

Department of Physics - Faculty of Science - University of Yaoundé I

Nom du Laboratoire

Date

[ ]:
```
from IPython.display import Image
```

1 Algorithme de la VQE

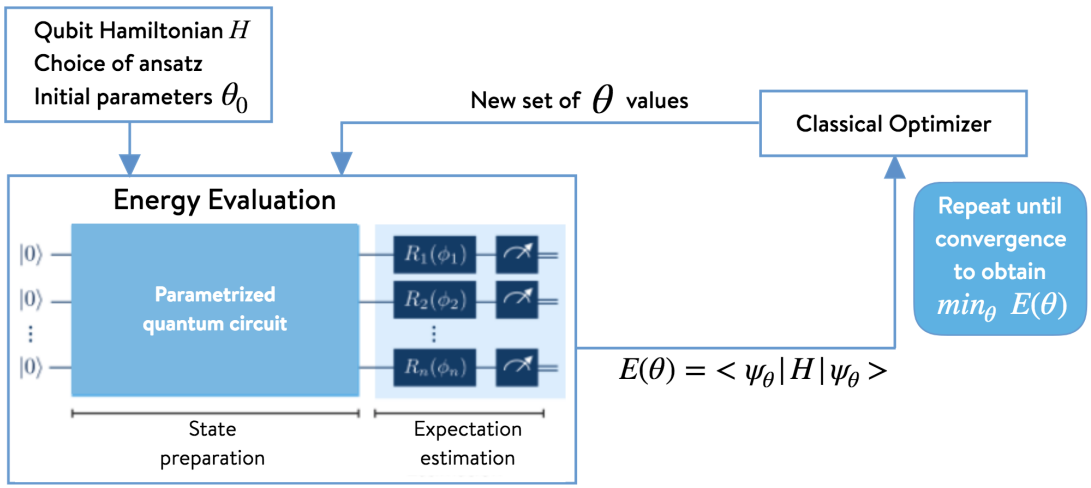
Cet exercice se traite exclusivement sur la feuille de composition

L'algorithme du VQE (Variationnal Quantum Eigensolver) se résume en deux grandes parties qu'illustre la figure ci-dessous.

1. Expliquer pourquoi on dit que c'est un **algorithme hybride** en indiquant ce que fait chaque processeur.
2. Qu'est-ce que la profondeur d'un circuit quantique et quelle dans les algorithmes du VQE?

[ ]:
```
Image("./VQE_Diagram.png", width=600)
```

[ ]:



2 Support Vector Regression to predict polarity of molecules

Write a python script that use a **support vector regression (SVR)** model to predict **TPSA** (**Topological polar surface area**, a physicochemical property describing the polarity of molecules) from molecules of the ZING Dataset contained in the file `'.Zn10.txt'`. The input - structural feature of molecules is Morgan fingerprint and the output is TPSA.

The workflow is

1. Get molecular fingerprints of each molecule
2. Split the dataset to training set and test set
3. Train a SVR model
4. Check the accuracy of prediction with R^2 and mean-square error
5. Visualize the results from the model.

The various necessaries libraries and modules will be import where they are needed.

We recall that the **molecular fingerprint** represents the substructures of a molecule as a vector of binary numbers. It is a molecular structure descriptor to use as input to reveal the relationship between molecular structure and properties, called **Quantitative Structure-Activity Relationships (QSAR)**.

[ ]:
```
import sklearn
print(sklearn.__version__)
```

[ ]:
```
import rdkit
print(rdkit.__version__)
```

[ ]:
```
# Put your code here
```

3 Calculer l'état fondamental de la molécule de phenylsulfonyl-carbazole (PSPCz)

Les calculateurs quantiques pourraient être des outils inestimables pour étudier la structure électronique et les propriétés dynamiques de molécules et de matériaux complexes, car il est plus logique de modéliser des systèmes de la physique quantique sur un dispositif quantique que sur un calculateur classique. Le phenylsulfonyl-carbazole (PSPCz) dont la formule moléculaire est $C_{18}H_{13}NO_2S$ et son SMILES isomérique est C1=CC=C(C=C1)S(=O)(=O)C2=CC=CC3=C2NC4=CC=CC=C34, a des propriétés émettrices utiles de fluorescence retardée activée thermiquement (TADF) pour les applications de diodes électroluminescentes organiques (OLED). Son Hamiltonien en représentation de Pauli {`I`, `X`, `Y`, `Z`} est

$$\mathtt{H} = h_1(\mathtt{I}\otimes\mathtt{I}) + h_2(\mathtt{Z}\otimes\mathtt{I} - \mathtt{I}\otimes\mathtt{Z}) + h_3(\mathtt{Z}\otimes\mathtt{Z}) + h_4(\mathtt{x}\otimes\mathtt{X}) + h_5(\mathtt{X}\otimes\mathtt{I} + \mathtt{I}\otimes\mathtt{X} + \mathtt{X}\otimes\mathtt{Z} - \mathtt{Z}\otimes\mathtt{X}),$$

où les coefficients

$$h_1 = -0.518418,\ h_2 = -0.136555,\ h_3 = -0.025866,\ h_4 = 0.015725,\ h_5 = -0.000296.$$

1. Utiliser `rdkit.Chem` pour représenter en 2D de molécule PSPCz à partir de son SMILES.

[ ]:
```
# Put your code here
```

2. Utiliser pour calculer l'état fondamental de la molécule PSPCz,
   - `qiskit.opflow.operator_globals` et
   - `qiskit.algorithms.minimum_eigensolvers.NumPyMinimumEigensolver`.

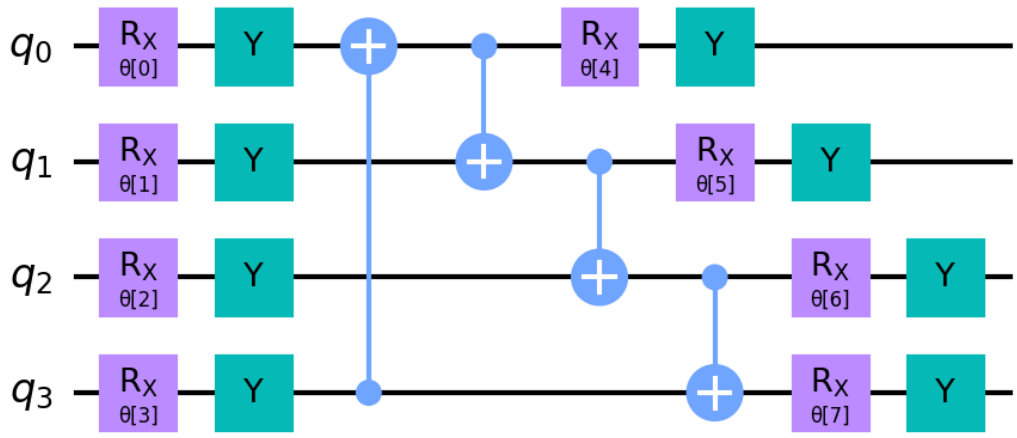
[ ]:
```
# Put your code here
```

4 VQE avec un Hardware Efficient Ansatz (HEA)

1. Utiliser `qiskit.circuit.library import EfficientSU2` pour reproduire le Hardware Efficient Ansatz (HEA) suivant :

[ ]:
```
Image(filename='./SU2_Ansatz.png', width=400)
```

[ ]:



2. Utiliser ce HEA (que l'on nommera `SU2ansatz`) dans l'algorithme VQE, définit ci-dessous par la fonction `algorithm(problem, mapper, optimizer)`, pour calculer l'état fondamental de la molécule H_2.

[ ]:
```
from qiskit.primitives import Estimator
from qiskit.algorithms.minimum_eigensolvers import VQE
from qiskit_nature.second_q.algorithms import GroundStateEigensolver

import numpy as np

def algorithm(problem, mapper, optimizer):
    """ Setup VQE solver algorithm

    Args:
        problem : Electronic Structure Problem
```

```
        mapper : qubit mapper
        optimizer : optimizer

    Returns: vqe solver algorithm
    """

    #ansatz
    ansatz = SU2ansatz # previously constructed

    #VQE algorithm solver
    vqe_solver = VQE(Estimator(), ansatz, optimizer)
    vqe_solver.initial_point = np.zeros(ansatz.num_parameters)

    # Ground state computation using a minimum eigensolver
    algorithm = GroundStateEigensolver(mapper, vqe_solver)

    # Compute Ground State properties.
    algorithm = algorithm.solve(problem)

    return algorithm
```

[ ]:
```
from qiskit_nature.units import DistanceUnit
from qiskit_nature.second_q.drivers import PySCFDriver
from qiskit_nature.second_q.transformers import FreezeCoreTransformer

H2_driver = PySCFDriver(
    atom='H .0 .0 -0.3625; H .0 .0 0.3625',
    basis="sto3g",
    charge=0,
    spin=0,
    unit=DistanceUnit.ANGSTROM,
)
# Electronic structure problem
H2_problem = H2_driver.run()
transformer = FreezeCoreTransformer()
H2_problem = transformer.transform(H2_problem)
```

[ ]:
```
from qiskit_nature.second_q.mappers import JordanWignerMapper
from qiskit_nature.second_q.mappers import TaperedQubitMapper
from qiskit.algorithms.optimizers import SLSQP

# Put your code here
```