Phirio

IA - programmation quantique

IA090

Durée: 1 jour

850 €

14 février
23 mai

19 septembre
19 décembre

Public :

Développeurs, data scientists, ingénieurs logiciels souhaitant découvrir l'informatique quantique par la pratique.
Professionnels curieux des nouvelles technologies quantiques et de leur potentiel concret.

Objectifs :

- Comprendre les différences clés avec l'informatique classique
- Comprendre intuitivement les concepts de qubit et superposition d'états
- Créer et exécuter des circuits quantiques sur simulateur et matériel réel
- Évaluer les possibilités et limitations actuelles de l'informatique quantique

Connaissances préalables nécessaires :

Programmation Python de base, notions de base en probabilités, curiosité pour les mathématiques et la physique

Programme :

Le générateur aléatoire quantique

Premier contact avec Qiskit : Installation et configuration de l'environnement (Jupyter + Qiskit). Premier circuit quantique : |0⟩ → H → mesure. Observation des résultats : vraiment 50/50 ?

Atelier : Générateur de nombres aléatoires quantique simple

Atelier : Comparaison avec random.choice() - même résultat, même mécanisme ?

Du bit au qubit : le qubit comme unité de calcul. Analogie avec le bit classique. Opérations de base.
États |0⟩, |1⟩ et la superposition d'états. Visualisation sur la sphère de Bloch avec Qiskit.
La mesure quantique : effondrement de la fonction d'onde.

Atelier : Visualiser les états avec plot_bloch_multivector()

Atelier : Expérimenter avec différents angles de rotation

Introduction aux portes quantiques, porte de Hadamard : créer la superposition parfaite.
Portes X, Y, Z : rotations sur la sphère de Bloch.
Composition de portes dans Qiskit.

Atelier : Créer des superpositions avec différentes amplitudes

Atelier : Visualiser les circuits avec circuit.draw('mpl')

Circuits multi-qubits et intrication

Passage à deux qubits.Systèmes à deux qubits, espace d'états en 4 dimensions.
Porte CNOT : création d'intrication. États de Bell : |00⟩ + |11⟩ et ses variants.

Atelier : Créer et mesurer les 4 états de Bell

Atelier : Mesurer les corrélations quantiques

Manipulation avancée : portes contrôlées : CNOT, CZ, Toffoli.
Circuits réversibles : propriété fondamentale du quantique.

Atelier : Construire une porte SWAP avec des CNOT

Atelier : Téléportation quantique (circuit complet)

Transition vers le matériel réel. Plateformes physiques : ions piégés, qubits supraconducteurs, photons, avantages et inconvénients
Différences entre simulation et matériel réel : bruit et décohérence.
Architecture des machines Rigetti.

Atelier : Même circuit sur simulateur local et machine Rigetti

Atelier : Observer et quantifier les effets du bruit

Algorithme de Simon

Problème de Simon : motivation et enjeux. Trouver la période cachée s dans f(x) = f(x⊕s).
Interférences pour révéler la structure.

Atelier : Cas simple s="11"

Construction progressive de l'algorithme, implémentation de f(x) ⊕ f(x⊕s).
Registres d'entrée et de sortie. Interférences quantiques : amplifier les bonnes réponses.

Atelier : Implémenter l'oracle pour s="11"

Atelier : Circuit complet Simon s="11" sur simulateur

Passage à la complexité réelle : s="1111". Scaling à 4 qubits : 16 possibilités classiques.
Construction de l'oracle 4 qubits. Extraction de l'information : résolution du système linéaire.

Atelier : Algorithme de Simon complet s="1111" sur simulateur

Atelier : Analyser les résultats et probabilités de mesure

Validation sur matériel quantique réel : adaptation pour les contraintes matérielles Rigetti.
Gestion du bruit et stratégies de mitigation. Optimisation du circuit (profondeur, connectivité).

Atelier : Simon s="1111" sur machine quantique physique

Atelier : Comparaison quantitative simulateur / matériel réel

Bilan et perspectives

Analyse des résultats obtenus. Avantage quantique observé : 16 essais classiques contre 4-5 mesures quantiques.
Impact du bruit sur l'algorithme : succès / échecs.
Techniques de correction d'erreur.

Atelier : Mesurer précisément l'impact du bruit

Écosystème quantique actuel : plateformes disponibles (IBM, Google, Rigetti, IonQ).
Langages et frameworks (Qiskit, Cirq, Forest).
Applications prometteuses à court terme.
Ressources pour continuer : projets pour approfondir.
Communautés et formations avancées. Veille technologique.

Atelier : Créer son propre défi quantique