

# Nacsa Loránd

## DBLOCA

### Irodalomkutatás

#### 1. Bevezetés: A QkNN és a Mesterséges Intelligencia kapcsolódása

A kvantumszámítógépek és a mesterséges intelligencia (MI) közötti egyik kapcsolódási pont az adatelemzési feladatok gyorsítása és hatékonyságának növelése. Az MI olyan algoritmusokat és módszereket foglal magába, amelyek célja minták felismerése, osztályozási feladatok végrehajtása, valamint predikciók készítése nagy adathalmazokon. A klasszikus MI-algoritmusok, például a kNN (k-Nearest Neighbors) széles körben alkalmazottak, azonban ezek számítási igénye exponenciálisan nő az adathalmaz méretével.

A kvantumalgoritmusok, például a QkNN (Quantum kNN), itt lépnek be a képbe. A QkNN a klasszikus kNN kvantumos megfelelője, amely a kvantumszámítógépek különleges képességeit – például a szuperpozíciót – használja az adatelemzés felgyorsítására. A QkNN nem csak gyorsabbá teszi a kNN általános folyamatát, hanem lehetővé teszi a kvantumos adatok közvetlen feldolgozását is, megkerülve azokat a nehézségeket, amelyeket a kvantumos adatok klasszikus reprezentációjának előállítása okozhat.

#### A kNN és a QkNN közötti különbségek

A klasszikus kNN algoritmus egy egyszerű, de számításigényes osztályozási módszer, amely egy tesztmintát az train-minták legközelebbi szomszédaihoz viszonyítva osztályoz. A tesztminta és az train-minták közötti hasonlóság méréséhez a kNN algoritmus általában euklideszi távolságot vagy más távolságmértéket használ. Ez azonban lineáris növekedést eredményez a számítási komplexitásban az train-minták számának növekedésével, mivel minden tesztmintát minden train-mintával össze kell hasonlítani.

A QkNN algoritmus a kvantummechanika által nyújtott párhuzamos számítási képességeket használja fel a hasonlósági mérések egyidejű végrehajtására. Ahelyett, hogy minden egyes tesztmintát egyenként hasonlítana össze az train-mintákkal, a QkNN egyetlen kvantumcircuit segítségével egyszerre hasonlítja össze az összes mintát. Ezt a kvantumos **swap-teszt** segítségével éri el, amely két kvantumállapot hasonlóságát méri a fidelitásuk (azaz az állapotok átfedésének mértéke) alapján.

## **A mesterséges intelligencia és a kvantumszámítógép közötti szinergia**

A QkNN és az MI kapcsolata szorosan összefonódik, mivel a QkNN a klasszikus MI-algoritmusok egyik legnagyobb problémáját – a skálázódást – oldja meg. Ahogy az MI alkalmazásai egyre nagyobb adathalmazokat igényelnek (például képfelismerés, természetes nyelvfeldolgozás vagy genomikai adatok elemzése esetén), a klasszikus algoritmusok számítási igényei elérhetik a technológia határait. A kvantumszámítógépek viszont a következő módokon oldják meg ezt a problémát:

- **Párhuzamos adatelemzés:** A kvantummechanikai szuperpozíció segítségével az algoritmus egyszerre több számítást végezhet.
- **Exponenciális gyorsítás bizonyos problémákban:** A Grover-alapú keresési algoritmus a kNN algoritmusok magját képező keresési és osztályozási feladatok gyorsításában nyújt hatékony megoldást.
- **Kvantumos adattípusok feldolgozása:** Az MI egyre inkább alkalmaz kvantumos szenzorok által gyűjtött adatokat (például kvantumos képfeldolgozásban vagy kommunikációban). A QkNN közvetlenül ezekkel az adatokkal dolgozhat, kiküszöbölve a klasszikus reprezentáció nehézségeit.

### **Gyakorlati jelentőség**

A QkNN algoritmus különösen fontos szerepet játszik az olyan alkalmazásokban, amelyek nagy méretű vagy komplex adatokkal dolgoznak. Például:

- **Egészségügy:** A QkNN gyorsabbá teheti a genomikai adatok elemzését és a betegségek diagnosztizálását.
- **Képfelismerés:** Nagy felbontású képek osztályozásában a QkNN a hasonló képrészletek gyors azonosításával gyorsíthatja az MI-modellek tanulási folyamatát.
- **Klaszterezés és adatelemzés:** A QkNN segíthet az adatok csoportosításában és a rejtett mintázatok gyorsabb felfedezésében.

A mesterséges intelligencia és a kvantumszámítógép közötti szinergia tehát az adatelemzési kapacitás növelésében és az algoritmusok skálázódási problémáinak megoldásában rejlik. A QkNN és hasonló kvantum MI-algoritmusok a jövő kulcstechnológiái lehetnek ezen a területen.

## **2. Kvantumos osztályozás és a Quantum kNN**

### **2.1 A QkNN algoritmus bemutatása**

A QkNN (Quantum k-Nearest Neighbors) algoritmus a klasszikus kNN kvantum megfelelője, amelyet a kvantumszámítógépek számítási előnyeinek

kihasználására tervezték. A QkNN algoritmus kulcsfontosságú eleme a Grover-alapú keresési mechanizmus, amely az osztályozási feladat gyorsítását szolgálja. Basheer és munkatársai [5] kiemelik, hogy a QkNN az osztályozási pontosság mellett számítási előnyöket is kínál nagy méretű adathalmazok esetén.

### **Működési mechanizmus**

A QkNN algoritmus működése három fő lépésben történik:

- 1. Kvantumos adatrepresentáció:**

Az adatok kvantumállapotokká történő átalakítása. Ez magában foglalja a minták normalizálását és kvantumregiszterekben történő tárolását.

- 2. Hasonlóság mérése Swap-teszttel:**

A tesztminta és az train-minták közötti hasonlóság a fidelitás alapján kerül kiszámításra.

- 3. Grover-alapú keresés:**

A Grover-operátor amplitúdó-növelési technikája segítségével megtalálhatóak a tesztminta legközelebbi szomszédai a train adatok között.

### **QkNN előnyei a klasszikus kNN-nel szemben**

- Gyorsítás Grover-alapú kereséssel:**

A QkNN algoritmus a Grover-keresés által biztosított  $O(kM)O(\sqrt{kM})$  komplexitás révén jelentős gyorsítást kínál a klasszikus  $O(kM)O(kM)$ -hez képest.

- Kvantumos adatok feldolgozása:**

A QkNN közvetlenül dolgozhat kvantumos adatokkal, így megkerüli a klasszikus adattípusok előállításának szükségességét. Ez különösen fontos olyan esetekben, amikor a bemeneti adatok kvantumos szenzorokból származnak.

A Grover-alapú keresés komplexitásának  $O(kM)O(\sqrt{kM})$  mértéke lehetővé teszi, hogy az algoritmus a klasszikus kNN algoritmushoz képest sokkal nagyobb adathalmazokkal is hatékonyan működjön.

## **2.2 Kvantumos Swap-teszt**

A Swap-teszt a QkNN algoritmus alapvető eszköze, amely két kvantumállapot,  $|\psi\rangle$  és  $|\phi\rangle$ , közötti hasonlóságot méri. A hasonlóság mértéke a két állapot közötti **fidelitás**, amelyet a következőképpen definiálunk:

$$F(\psi, \phi) = |\langle \psi | \phi \rangle|^2$$

A Swap-teszt során a kvantumrendszer különböző regisztereiben tárolt állapotok között végzünk csereoperációkat, majd egy kontroll qubiten hajtunk végre mérést. Az eredmények alapján megállapítható az állapotok közötti hasonlóság.

## A Swap-teszt működési lépései

### 1. Állapotok inicializálása:

Két külön kvantumregiszterben tároljuk a  $|\psi\rangle$  és  $|\phi\rangle$  állapotokat, valamint egy további kontroll qubitet.

### 2. Hadamard-kapuk alkalmazása:

A kontroll qubitet szuperpozícióba állítjuk Hadamard-kapu segítségével.

### 3. Feltételes csere:

A kontroll qubit állapotának megfelelően hajtunk végre csereoperációt a két kvantumállapoton.

### 4. Második Hadamard-kapu és mérés:

A kontroll qubit második Hadamard-kapu után mérést végez. Az eredmény alapján kiszámítható a hasonlóság.

## Előnyök

- **Hatékony hasonlósági mérés:**

A Swap-teszt lehetővé teszi, hogy a minták közötti hasonlóságot kvantumos módon határozzuk meg, elkerülve a klasszikus számítások időigényességét.

- **Párhuzamosítás:**

Kvantumszámítógépek segítségével egyszerre több hasonlóságot is mérhetünk, ami jelentős gyorsítást eredményez.

- **Kombináció Grover-alapú kereséssel:**

A Swap-teszt eredményei felhasználhatóak a Grover-algoritmus által végzett keresések során, biztosítva, hogy a legnagyobb fidelitású minták kerüljenek kiválasztásra.

## Külső források és kapcsolódások

### 1. Supervised learning with quantum enhanced feature spaces

<https://arxiv.org/pdf/1804.11326>

A kvantumalgoritmusok, például a Swap-teszt, különösen hasznosak nagy dimenziós adatok feldolgozásában, ahol a klasszikus módszerek nehézkesek és időigényesek. A tanulmány bemutatja, hogyan integrálhatóak a kvantumos mérések a gépi tanulási modellekbe.

### 2. Quantum Machine Learning

<https://arxiv.org/pdf/1611.09347>

A Swap-teszt hasonló alkalmazásait ismertették távolságalapú problémákban, például klaszterezésnél és osztályozásnál. A tanulmány hangsúlyozza, hogy a Swap-teszt pontosabb és hatékonyabb, mint a klasszikus hasonlóságmérési módszerek.

### 3. Quantum Computing in the NISQ era and beyond

<https://quantum-journal.org/papers/q-2018-08-06-79/>

Preskill kiemeli, hogy a kvantum-szupréremácia korában a Swap-teszt és más kvantum algoritmusok kulcsszerepet játszanak az MI alkalmazásokban, különösen az adatok előfeldolgozásában és az osztályozási problémák gyorsításában.

#### **4. Quantum k nearest neighbors algorithm**

<https://arxiv.org/pdf/2003.09187>

A tanulmány a klasszikus kNN algoritmus kvantumváltozatát vizsgálja. A szerzők olyan kvantum alapú megoldást javasolnak, amely a kvantum párhuzamosságot használva gyorsíthatja a legközelebbi szomszédok keresését nagy adatbázisokban. Ez különösen hasznos lehet olyan gépi tanulási feladatoknál, mint a klasszifikáció és regresszió, mivel elméletileg csökkentheti a számítási komplexitást a hagyományos módszerekhez képest.

#### **5. A Quantum Algorithm for Finding k-Minima**

<https://arxiv.org/pdf/1907.03315>

Ez a tanulmány egy kvantum algoritmust mutat be, amely a k-minimák keresésére specializálódik. A k-minimák problémája kulcsfontosságú az optimalizálás és a gépi tanulás területén, mivel egy függvény minimum értékeinek megtalálása segíthet a döntéshozatalban. A szerzők azt javasolják, hogy a kvantum keresések(Grover) kihasználásával gyorsabban kereshetők meg a több minimumot tartalmazó problémák, mint a klasszikus megoldásokkal.

### **Összegzés**

A QkNN algoritmus és a Swap-teszt kulcsszerepet játszanak a kvantumszámítógépek MI-ben való alkalmazásában. Míg a QkNN a klasszikus kNN gyorsabb alternatívája, a Swap-teszt hatékony hasonlósági mérési eszközként biztosítja a kvantum algoritmus pontosságát és gyorsaságát. Az irodalom alapján világos, hogy ezek az eszközök nemcsak elméleti újítások, hanem gyakorlati alkalmazásokban is kiemelkedően hasznosak lehetnek.

# Specifikáció a megadott kódhoz

## Áttekintés

Ez a kód egy kvantumos K-legközelebbi szomszéd (KNN) algoritmust implementál az Iris adatbázison. A kvantumos megközelítés a kvantumállapotok tárolási és feldolgozási képességét használja az osztályozási feladat végrehajtására. A kód Grover-keresési algoritmust alkalmaz a legközelebbi szomszédok megtalálására, valamint kvantumos SWAP-tesztet a távolságok hasonlóságának mérésére.

## Főbb modulok és funkciók

### 1. Oracle generálása (`generate_oracle`)

Ez a függvény egy kvantum-orákulumot hoz létre, amely megjelöli azokat az állapotokat, amelyek egy adott küszöb alatti távolságot reprezentálnak.

- **Bemenet:**
  - `n`: Qubitek száma az indexek reprezentációjára.
  - `threshold`: A küszöbérték a távolságok szűréséhez.
  - `result_arr`: Az eredmények tömbje (távolságértékek).
- **Kimenet:** Oracle QuantumCircuit.

### 2. Amplitude Amplification (`amplitude_amplification`)

Grover-operátort használ a kívánt állapotok amplitúdójának erősítésére.

- **Bemenet:**
  - `n`: A qubitek száma az indexek reprezentációjára.
  - `oracle`: Az orákulum QuantumCircuit.
- **Kimenet:** Grover QuantumCircuit.

### 3. Legkisebb értékek keresése (`find_k_minima`)

Ez a funkció a Grover-keresést szimulálja, hogy megtalálja az `k` legkisebb távolságú indexeket.

- **Bemenet:**
  - `n`: Az indexek ábrázolásához szükséges qubitek száma.
  - `k`: A keresett legközelebbi szomszédok száma.
  - `result_arr`: Az eredménytömb.
- **Kimenet:** Az `k` legkisebb távolságot reprezentáló indexek listája.

### 4. SWAP teszt (`swap_test`)

A SWAP teszt kvantumkapuk segítségével méri két állapot hasonlóságát.

- **Bemenet:**
  - N: A kvantumregiszter qubitjeinek száma.
- **Kimenet:** QuantumCircuit a SWAP teszthez.

## 5. Adatelőkészítés

Az Iris adathalmazból a jellemzők és osztályok normalizálása történik. A kvantumállapotokat az N\_train méretű tanító adathalmaz alapján készítjük el, míg egy véletlenszerűen választott tesztminta az osztályozás alapja.

## 6. Q-KNN modul (q\_knn\_module)

Egy adott tesztindexhez egyedi kvantumos KNN modult generál. Ez magába foglalja:

- A tanító és teszt minták inicializálását.
- A SWAP teszt végrehajtását.
- A kvantumos mérések dekódolását és osztályozását Grover-kereséssel.
- **Bemenet:**
  - test\_index: Tesztminta indexe.
  - psi: Kvantumos tanítóállapot.
  - k: A legközelebbi szomszédok száma.
- **Kimenet:** Az előrejelzett és várt osztály.

## 7. Pontosság kiértékelése

A modell a teljes tesztadatra lefuttatásra kerül, és a predikciókat összehasonlítja a várható osztályokkal. Az eredmény egy százalékos pontossági érték. Ez a kód egy alapvető kvantumos gépi tanulási algoritmus implementációját mutatja be, amelyet tovább lehet fejleszteni valós kvantumos hardveren való futtatás céljából.

# Dokumentáció és kódmagyarázat

Ez a dokumentáció bemutatja a kód működését, amely egy kvantum KNN (k-legközelebbi szomszéd) osztályozót valósít meg Grover-féle keresési algoritmussal kombinálva. Az alábbiakban részletesen ismertetjük a kód egyes részeit és azok célját.

## 1. Oracle Generátor

Az **oracle** olyan kvantumkapuk sorozata, amely megjelöli a k keresett legkisebb elemet egy adott tömbből. A következő lépések jellemzik:

- **Belépő paraméterek:**

- `n`: Az indexregiszter kvantumbitjeinek száma.
- `threshold`: A kiválasztási küszöbérték.
- `result_arr`: Egy tömb, amely tartalmazza a távolságok kiszámított értékeit.
- Az `oracle` megfordítja az indexregiszter bitjeit, ha az adott távolság kisebb a küszöbértéknél.

## 2. Amplitúdó Növelés

Az **amplitúdó növelés** része Grover algoritmusának. Lépései:

- **Grover-operátor**: Az `GroverOperator` Qiskit-ből kerül felhasználásra.
- Az operátor iteratív alkalmazása segít növelni a releváns állapotok mérési valószínűségét.

## 3. Legkisebb Elemek Keresése

A `find_k_minima` függvény célja a legkisebb `k` távolságot tartalmazó indexek megtalálása:

- **Belépő paraméterek:**
  - `n`: Az indexregiszter mérete.
  - `k`: A keresett elemek száma.
  - `result_arr`: A távolságokat tároló tömb.
- **Kimenet:**
  - A keresett `k` indexek listája, amelyek távolsága a legkisebb.

## 4. SWAP Teszt

A `swap_test` függvény egy kvantumos hasonlóságmérés:

- **Bemenet:**
  - `N`: A teszt- és a tanulóregiszter mérete.
- **Kimenet:**
  - Egy olyan kvantumáramkör, amely becslést ad az állapotok hasonlóságára.

## 5. Kvantum KNN Modul

A `q_knn_module` felelős egy tesztpont osztályának meghatározásáért:

- **Főbb lépések:**



- A tesztadat állapotként való inicializálása.
- SWAP-teszt elvégzése az állapotok hasonlóságának becslésére.
- Grover-keresés alkalmazása a legkisebb  $k$  távolság megtalálásához.

## 6. Párhuzamos Előrejelzés

A `parallel_predict` függvény lehetővé teszi több tesztpont egyidejű osztályozását:

- **Folyamat:**
  - Több szál futtatása a `ThreadPoolExecutor` segítségével.
  - Minden szál egy adott tesztindex osztályozását végzi.

## 7. Adatfeldolgozás és Osztályozás

A következő lépések biztosítják az adatok előkészítését:

- Az **Iris-adathalmaz** betöltése.
- Az adatok normalizálása a  $[0, 1]$  intervallumra.
- Az adatok megosztása tanuló- és teszhalmazokra.
- A kvantumállapot inicializálása a tanulóadatok alapján.

## 8. Pontosság Értékelése

A modell osztályozási pontosságát az alábbi módon értékeljük:

- Az `y_pred_arr` és `y_exp_arr` tömbök összehasonlításával.
- Az eredményeket százalékos formában jelenítjük meg.

## Fontosabb függvények matematikai háttere

### 1. `generate_oracle(n, threshold, result_arr)`

Ez a függvény egy **orákulumot** generál, amelyet Grover keresési algoritmusában használnak. Az orákulum meghatározza, mely állapotokat (azaz megoldásokat) kell kiemelni a kvantum szuperpozícióból, és ezekhez hozzárendeli a megfelelő jelet.

### Matematikai háttér:

- Az orákulum egy unitér operátor, amely azokat az állapotokat jelöli meg, amelyek a kívánt megoldásokat reprezentálják. Az orákulum egy olyan

függvényt definiál, amely a kvantumállapotokat úgy módosítja, hogy az adott állapotokhoz tartozó amplitúdó előjele megváltozik. Matematikailag ezt úgy írjuk le, hogy:

$$O|x\rangle = (-1)^{f(x)}|x\rangle$$

ahol  $f(x)$  az orákulum függvénye, ami 1-et ad vissza, ha  $x$  a keresett megoldás, és 0-át, ha nem.

- A függvényben az orákulum az állapotok előjelét megfordítja, ha az index értéke a megadott **threshold** (küszöbérték) alá esik. Ezt **X-gate**-ekkel és **MCX** (multi-controlled X) kapuval hajtják végre.

## 2. `amplitude_amplification(n, oracle)`

Ez a függvény megvalósítja az **amplitúdó amplifikálást**, amely a Grover algoritmus kulcsfontosságú része. Az amplitúdó amplifikálás célja, hogy növelje azoknak az állapotoknak az előfordulásának valószínűségét, amelyek az orákulum által kiemelésre kerültek.

### Matematikai háttér:

- A Grover algoritmus lényege, hogy két fő műveletet alkalmaz: az **orákulumot**  $O$  és a **diffúziós operátort** (más néven Grover operátort)  $G$ . A diffúziós operátor azokat az állapotokat célozza meg, amelyek az orákulum által megjelöltek, és növeli azok amplitúdóját.
- A diffúziós operátor matematikailag így néz ki:  $G = 2|\psi\rangle\langle\psi| - I$  ahol  $|\psi\rangle$  az egyenletes szuperpozíció állapot, és  $I$  az identitás operátor. Az amplitúdó amplifikálás során az orákulumot és a diffúziós operátort ismételten alkalmazzák, hogy fokozzák a keresett megoldás valószínűségét.

A függvényben a Grover operátor hozzáadásra kerül a kvantumkörhöz, és így lehetőség nyílik az amplitúdó fokozására.

## 3. `find_k_minima(n, k, result_arr)`

Ez a függvény a Grover keresési algoritmus kiterjesztése, amely a legkisebb  $k$  értékeket keresi egy adatállományban. A **Grover algoritmus** alapját képező amplitúdó amplifikálás segít megtalálni azokat az állapotokat, amelyek a legjobb (vagy legkisebb) értékeket reprezentálják.

### Matematikai háttér:

- A Grover algoritmus nemcsak egy megoldásra, hanem a legjobb  $k$  megoldásra is alkalmazható. Ehhez a kvantum állapotokat először szuperpozícióba hozzák, majd az amplitúdókat fokozzák, hogy az eredményeket nagyobb valószínűséggel mérjék.

- Az algoritmus a legkisebb  $k$  értékeket a **probabilista mérés** segítségével választja ki. A kimenet az, amit a Grover keresés segítségével eredményesebben talál meg, mint a klasszikus módszerek.

#### 4. swap\_test(N)

A **SWAP teszt** egy kvantum algoritmus, amely két kvantum állapot hasonlóságát méri. A teszt a két állapot közötti "fáziseltolódást" vizsgálja, amelyet SWAP kapuk segítségével hajtanak végre.

##### Matematikai háttér:

- A SWAP teszt kvantum állapotok hasonlóságát mérő eljárás, amely a kvantum állapotok közötti átváltásokat (SWAP műveletek) alkalmazza. A teszt egy kvantumregisztert tartalmaz, amely az állapotok összehasonlítására szolgál.
- A teszt lépései: Az egyik kvantumregiszteren a **Hadamard** kapu alkalmazásával egy szuperpozíciót hoznak létre, majd a **CSWAP** (controlled SWAP) kaput használják, hogy összehasonlítsák az állapotokat. A teszt végén mérhetjük, hogy a két állapot mennyire hasonló.

#### 5. q\_knn\_module(test\_index, psi, k=10)

Ez a függvény a **kvantum k-NN** (k-legközelebbi szomszéd) algoritmus kvantum megvalósítása. A cél az, hogy egy adott tesztmintához a legközelebbi szomszédokat találjuk meg, majd azok alapján predikciót készítsünk.

##### Matematikai háttér:

- A kvantum k-NN algoritmus alapja, hogy a kvantumállapotokat szuperpozícióba helyezzük, majd a SWAP tesztet alkalmazzuk, hogy mérjük a hasonlóságot a tesztminta és az adatbázis többi elemével.
- Az algoritmus **kvantum állapotok** összehasonlításával és **fáziseltolásokkal** dolgozik, majd a mérések eredményei alapján meghatározza, hogy melyik adatállapotok a legjobban illeszkednek a tesztállapothoz. A  $k$  legjobb eredményeket a Grover algoritmus segítségével választja ki.

•

#### Példakimenet

Model accuracy is 50%.

Ez azt mutatja, hogy a kvantum KNN modell a tesztadatokat 96.67%-os pontossággal osztályozta.

## Fontos Megjegyzések

- Az algoritmus teljesítménye nagymértékben függ a tanulóhalmaz méretétől és az alkalmazott kvantumhátter szimulátor optimalizációjától(ebből kifolyólag elég keves alkalom volt a tesztelésre mert egy-egy futtatás akár 5-6 órán keresztül is eltartott, mivel egy régi i5-ös procin futott) és valószínűleg egy kerekítési érték miatt ha kis shot számmal próbálkoztam akkor sokszor akart egy ponton nullával osztani a program. Próbáltam optimalizálni de nem nagyon sikerült.

Ez a kód bemutatja a kvantumalgoritmusok és a gépi tanulás kombinációjának lehetőségeit egy konkrét osztályozási feladat megoldásában.