

Kvantno mašinsko učenje

Milan Bojić

Septembar 2022.

Sadržaj

1	Uvod	3
2	Kvantno računarstvo	4
2.1	Osnovni pojmovi	4
2.2	Kvantno računarstvo	6
2.3	Kvantna informacija	6
2.3.1	Priprema podataka	8
2.4	Linearna algebra za kvantno mašinsko učenje	8
3	Kvantna teorija kompleksnosti i kvantni algoritmi	10
3.1	Primeri klasa kompleksnosti	10
3.2	BQP	10
3.3	Primeri kvantnih algoritama	11
3.3.1	Šorov algoritam	11
3.3.2	SWAP Test	12
4	Kvantno mašinsko učenje	14
4.1	Kvantna metoda potpornih vektora	14
4.1.1	Klasičan algoritam	14
4.1.2	Kvantni algoritam	16
4.2	Kvantna analiza glavnih komponenti	18
4.2.1	Klasičan algoritam	18
4.2.2	Kvantni algoritam	19
4.3	Kvantna neuralna mreža	20
4.3.1	Reprezentacija modela	22
4.3.2	Učenje modela	24
4.3.3	Učenje osobina kvantnih stanja	27
5	Zaključak	29

1 Uvod

Prethodna decenija je bila obeležena mašinskim učenjem, sa njegovom primenom u svakodnevnom životu običnih ljudi. Međutim, u prethodnih nekoliko godina počelo je da se oseća usporenje inovacija i razvoja novih metoda, a u nekim oblastima su i dostignute granice računarskih resursa (npr. GPT-3). U rešavanju ovog problema očekuje se da pomogne razvoj kvantnog računarstva, oblasti koja se razvija gotovo pola veka, ali tek u poslednjih nekoliko godina je došlo do povećanog interesovanja. Za kvantno računarstvo se očekuje da bude sledeće veliko remećenje tehnološkog poretka, sa približavanjem kvantnoj nadmoći svakog dana.

Od kvantnog mašinskog učenja se očekuje da bude prekretnica u mašinskom učenju kakvo danas znamo. Oblast je relativno mlada i trenutno se najviše bavi teoretskim razvojem. Neki od zadataka kojim se bavi jeste obrada kvantnih sistema i “učenje” njihovih osobina, brže i bolje prepoznavanje obrazaca u klasičnom sistemu, kao i otkrivanje nekih osobina klasičnog mašinskog učenja koje se nisu mogle primetiti u klasičnom sistemu.

U ovom radu će biti uvedene u osnove kvantnog računarstva i predstavljeno nekoliko metoda kvantnog mašinskog učenja koji su bili istraživani u prethodnom periodu.

2 Kvantno računarstvo

Pre nego što se počne pričati o Kvantnom mašinskom učenju, treba objasniti neke osnovne pojmove da bi se ostatak rada mogao lakše razumeti.

2.1 Osnovni pojmovi

Potrebni pojmovi su:

- Kubit (engl. Qubit)
- Kvantna kapija (engl. Quantum Gates)
- Kvantna spletenost (engl. Quantum entanglement)
- Kvantna memorija, Kvantni registri

Kubit

Kubit (engl. Qubit) je najmanja jedinica informacije u kvantnom računarstvu, slično bit-u u klasičnom računarstvu. Razlika od bita jeste u tome što kubit pored stanja 1 i 0, može da se nalazi i u superpoziciji između oba. Oni se mogu predstaviti formulom (koristeći bra-ket notaciju):

$$|\gamma\rangle = \alpha |0\rangle + \beta |1\rangle$$

Ovde su $|0\rangle$ i $|1\rangle$ zapravo stanja kao i kod klasičnog bita, a α i β su kompleksni brojevi koji predstavljaju amplitude zadatih stanja i za njih važi:

$$|\alpha|^2 + |\beta|^2 = 1$$

Stanje kubita ima dva stepena slobode što dovodi do toga da amplitude se mogu zapisati kao:

$$\alpha = \cos \frac{\Theta}{2}$$
$$\beta = e^{i\phi} \sin \frac{\Theta}{2}$$

gde je $e^{i\phi}$ relativna faza kubita, a Θ ugao.

Takođe možemo da vidimo da je $|\alpha|^2$ verovatnoća da se kubit nalazi u stanju 0, isto važi i za $|\beta|^2$ i 1. Saznanje o tome u kom stanju se nalazi kubit

se dobija merenjem kubita, kada bi kubit izašao iz superpozicije i “pao” u stanje 1 ili stanje 0. U tom slučaju kubit će imati ponašanje kao i običan bit, ali ovako gubimo pređašnje kvantno stanje kubita. U fizičkom svetu kubit se može predstaviti kao polarizovan foton, gde će dva stanja da se uzimaju kao vertikalna i horizontalna polarizacija.

Kvantna kapija

Kvantne kapije (engl. Quantum Gates) su logički predstavljene matricama i one rade nad određenim brojem kubita. Matrice su unitarne sa oblikom $2^n \times 2^n$, gde je n broj kubita sa kojim radimo. Neke od poznatih kola su: Hademardovo kolo (stavlja kubit u superpoziciju)

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Pauli-jeva kola (zamenjuju amplitude na kubit u po osama), ali nas najviše zanima rotaciono kolo:

$$R = \begin{bmatrix} \cos \Theta & -\sin \Theta \\ \sin \Theta & \cos \Theta \end{bmatrix}$$

Ovo kolo rotira kubite u prostoru, odnosno menja njihove amplitude za Θ radijana.

Kvantna spletenost

Kvantna spletenost (engl. Quantum entanglement) je fizički pojam gde su dva, ili više, kubita povezana tako da zajedno prave novo kvantno stanje. U čistim stanjima oni su matematički zapravo proizvodi tenzora amplitude:

$$|\gamma\rangle \otimes |\delta\rangle = \alpha_1\alpha_2 |00\rangle + \alpha_1\beta_2 |01\rangle + \beta_1\alpha_2 |10\rangle + \beta_1\beta_2 |11\rangle$$

I ovako napisano kvantno stanje se može razdvojiti na dva kubita. Ali postoje i kvanta stanja koja se ne mogu razdvojiti npr.

$$\frac{1}{\sqrt{2}} |00\rangle + \frac{1}{\sqrt{2}} |11\rangle$$

Zanimljiva stvar kod spletenih kubita jeste u tome što dele informacije. Ako bi jedan kubit iz para odneli u neko veoma daleko mesto (na primer druga

galaksija), i tamo bi ga izmerili, dobili bi smo 0 ili 1, međutim drugi kubit bi takođe upao u određeno stanje i to u istom trenutku kad smo izmerili prvi daleki kubit. Ovo je zapravo gde se nalazi glavna različitost između klasičnog i kvantnog računarstva, ova pojava ne postoji u klasičnom računarstvu i ne može se "lako" simulirati.

Kvantni registri

Kvantni registri se sastoje od kvantnog stanja od m spletenih kubita i mogu predstavljati do 2^m vrednosti stanja istovremeno. Kvantna memorija su uređaji koji čuvaju kvantna stanja fotona, bez uništavanja kvantne informacije koja se nalazi u fotonu. Ovakva memorija zahteva koherentni sistem materijala, jer bi u suprotnom kvantna informacija unutar uređaja bila izgubljena zbog nekoherentnosti.

2.2 Kvantno računarstvo

Kvantno računarstvo je vrsta računarstva gde se koriste kolekcije fizičkih osobina kvantne mehanike kao što su superpozicija i kvantna spletenost, tako da se izvrši neka kalkulacija. Uređaji koji izvršavaju kvantne kalkulacije zovu se **kvantni računari** (Slika 1). Kvantni računari se sastoje od kvantnih kola i elementarnih kvantnih kapija koje služe za prenošenje i manipulisanje kvantnih informacija. [14]

Jedna od glavnih primena kvantnih računara jeste simulacija fizičkih sistema, bili oni kvantne ili klasične prirode.

2.3 Kvantna informacija

Kvantna informacija je informacija o stanju kvantnog sistema. Njihovim svojstvima bavi se **kvantna teorija informacije**. Takođe, kvantne informacije se mogu izmeriti na isti način kao i klasična informacija koristeći se *Šenonovom metodom*. Postoji jedinstveno merilo, to jest funkcija nad kvantnim stanjem, koja je funkcija verovatnoće, kontinuiteta i sumiranja.[17] Ova funkcija se zove **fon Nojmanova entropija** i za neki ulazni kubit ϱ postoji ekvivalent u **Šenonovoj entropiji** H za neku slučajnu promenljivu X

$$S(\varrho) = H(X)$$



Slika 1: Slika spoljašnjosti kvantnog računara [13]

Još jedno od merila za kvantno stanje jeste merenje validnosti (engl. fidelity) između dva kvantna stanja $|\phi\rangle$ i $|\psi\rangle$. Neka je F funkcija koja meri osobinu, odnosno meri verovatnoću da merenjem stanja $|\phi\rangle$ dobijemo stanje $|\psi\rangle$. Izlaz funkcije je između 0 i 1, gde ako je izlaz 0 onda su dva stanja ortogonalna jedna od drugog, a ako je izlaz 1 onda su dva stanja jednaka [17].

Odnos kvantne i klasične teorije informacije

Kvantna i klasična informacija se u dosta stvari razlikuju. Klasična informacija prolazi kroz sisteme sa dobro definisanim stanjima, može se kopirati i pri procesu merenja se ne menja. Kvantna informacija je kodirana u kvantnim sistemima, ne može se kopirati i pri procesu merenja ona se menja. Takođe kvantna informacija ima neke osobine koje se ne mogu iskazati u klasičnoj informaciji, kao što su superpozicija i kvantna spletenost [11].

Kvantna teorija informacije se bavi:

1. Prenošnjem klasičnih informacija preko kvantnih kanala
2. Prenošnjem kvantnih informacija preko kvantnih kanala
3. Efektima kvantne spletenosti na prenošenje informacija

4. Informacionm aspektma kvantnog merenja, odnosa između distribucije kvantnog stanja i preciznog merenja

2.3.1 Priprema podataka

Za obradu podataka treba nam kvantni RAM (QRAM), koji nam dozvoljava paralelan pristup kvantnim podacima. Neka imamo kompleksan vektor \vec{v} sa $N = 2^n$ dimenzija, gde su njegove komponente oblika

$$v_j = |v'_j|e^{i\varphi_j}$$

Ako imamo parove $\{|v'_j|, \varphi_j\}$, čuvamo ih kao realne brojeve u QRAM-u, onda možemo da konstruišemo $\log_2 N$ kvantnih stanja $|v\rangle = |\vec{v}|^{-\frac{1}{2}}\vec{v}$ u $O(\log N)$ koraka

Kada smo kreirali kompresovane kvantne vektore od ulaznih vektora, možemo da vršimo transformacije koristeći kvantne algoritme, za dalje korišćenje podataka za mašinsko učenje. Ovaj proces zove se **pred procesiranje** i u opštem obliku njemu je potrebno $O(poly(\log N))$ koraka [9].

2.4 Linearna algebra za kvantno mašinsko učenje

Da bi se videlo kako kvantni računari poboljšavaju mašinsko učenje, treba da se vidi kako kvantni računari obrađuju linearnu algebru, jednu od osnova modernog mašinskog učenja.

Tokom godina razvijeno je nekoliko kvantnih algoritama koji rešavaju probleme linearne algebre. Zajedno ti algoritmi se nazivaju **osnovni kvantni podprogrami linearne algebre** (engl. qBLAS), i oni se koriste u izradi algoritama za kvantno mašinsko učenje.

Primeri algoritama koji su deo qBLAS-a su:

- HHL algoritam: koristi se za rešavanje sistema linearnih jednačina, koristeći 2^n -dimenzioni vektorski prostor za rešavanje sistema sa n promenljivih [3].
- Kvantna Furijeova transformacija [12].
- Kvantna procena faza za sopstvene vrednosti i sopstvene vektore/stanja [12].

Ovi algoritmi su korišćeni kao osnova naprednih metoda i algoritama za Kvantno mašinsko učenje. Samo treba pripaziti kod pominjanja ovih algoritama, jer neki od njih koriste neke koncepte koji su samo teorijske prirode ili su teško kreirani u realnom svetu (npr. QRAM).

3 Kvantna teorija kompleksnosti i kvantni algoritmi

U klasičnoj teoriji kompleksnosti klasifikuju se algoritamski problemi po njihovoj težini rešavanja. Problemi se klasifikuju u **klase kompleksnosti**, koje se mogu posmatrati kao kolekcija algoritamskih problema koji dele neke zajedničke osobine vezane za računске resurse potrebne da bi se rešili (uglavnom vreme i prostor) [14].

3.1 Primeri klasa kompleksnosti

Među najpoznatijim i najvažnijim klasama su **P** i **NP**. Zbog prirode kvantnih računara biće objašnjene neke druge klase kompleksnosti.

Klasa kompleksnosti **PSPASE** je klasa problema koji se mogu rešiti u polinomskom prostoru, ali sa neograničenim vremenom izvršavanja.

Klasa kompleksnosti **PP** (Probabilistic Polynomial-Time) je klasa problema za koje postoji nasumični algoritam u polinomskom vremenu koji vraća tačno rešenje sa verovatnoćom većom od $\frac{1}{2}$.

Klasa kompleksnosti **BPP** (Bounded-Error Probabilistic Polynomial-Time) je klasa problema za koje postoji nasumični algoritam u polinomskom vremenu koji vraća tačno rešenje sa verovatnoćom većom od $\frac{2}{3}$ [1].

3.2 BQP

Klasa kompleksnosti **BQP** (Bounded-Error Quantum Polynomial-Time) je klasa problema koji se mogu efikasno rešiti na kvantnom računaru, ako se dopusti ograničena verovatnoća greške [14]. Formalnija definicija bi bila [1]:

Definicija 1 ***BQP** je klasa jezika $L \subseteq \{0, 1\}^*$ za koje postoji uniformni skup kvantnih kola polinomske veličine (C_n) tako da za svako $x \in \{0, 1\}^n$:*

- *ako $x \in L$ onda C_n prihvata ulaz $|x\rangle|0\dots 0\rangle$ sa verovatnoćom većom od $\frac{2}{3}$.*
- *ako $x \notin L$ onda C_n prihvata ulaz $|x\rangle|0\dots 0\rangle$ sa verovatnoćom ne većom od $\frac{1}{3}$.*

Ovako definisano, može se primetiti da problemi iz **BQP** su dosta bliži problemima iz **BPP** nego iz **P**.

Odnos sa klasičnim klasama kompleksnosti

Prva stvar koja važi jeste da $\mathbf{BPP} \subseteq \mathbf{BQP}$, odnosno da sve što može da se uradi sa klasičnom probalističkim računarom može da se uradi i na kvantnom računar.

Kada se traži gornja granica kvantnih problema prvo se dolazi do $\mathbf{BQP} \subseteq \mathbf{EXP}$, ovo znači da kvantni računari mogu da dovedu do najviše *eksponencijalnog ubrzanja* u odnosu na klasični računar [1]. Bolja gornja granica za \mathbf{BQP} jeste $\mathbf{BQP} \subseteq \mathbf{PP}$. Ovo su dokazali Adleman, DeMarrais i Huang u [2].

3.3 Primeri kvantnih algoritama

3.3.1 Šorov algoritam

Trenutno optimalan algoritam na klasičnom računar (general number field sieve) se izvršava u sub-eksponencijalnom vremenu. Šorov algoritam rešava problem nalaženja periode, koji može da se iskoristi za problem faktORIZACIJE [16].

Problem: ako imamo periodičnu funkciju

$$f(x) = a^x \bmod N$$

gde su a i N prirodni brojevi, $a < N$, koji nemaju zajedničkog faktora. Period r je najmanji prirodan broj tako da:

$$a^r \bmod N = 1$$

Šorov algoritam koristi procenu kvantne faze na unitarnom operatoru:

$$U|y\rangle = |ay \bmod N\rangle$$

Tako da bi sopstvena vrednost za U bila jednaka superpoziciji stanja $|u_0\rangle$

$$|u_0\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} |a^k \bmod N\rangle$$

$$U|u_0\rangle = |u_0\rangle$$

Ovde sopstveno stanje ima sopstvenu vrednost 1, što nam ništa ne znači. Zanimljivije je kada se gledaju sopstvena stanja gde su faze drugačije za

svako bazno stanje. Posebno se posmatraju slučajevi gde faza k -tog stanja je proporcionalna k :

$$|u_1\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-\frac{2\pi i k}{r}} |a^k \bmod N\rangle$$

$$U |u_1\rangle = e^{\frac{2\pi i}{r}} |u_1\rangle$$

U ovom slučaju sopstvena vrednost sadrži r . Ovde je r kao faktor normalizacije između r baznih stanja. Sada, možemo ova stanja da pomnožimo sa $s \in N_0$ što će uticati na krajnju sopstvenu vrednost.

$$|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-\frac{2\pi i s k}{r}} |a^k \bmod N\rangle$$

$$U |u_s\rangle = e^{\frac{2\pi i s}{r}} |u_s\rangle$$

Sada imamo jedinstveno sopstveno stanja za svako s ($0 \leq s \leq r-1$). Ako saberemo sva dobijena sopstvena stanja, razlike u fazama se poništavaju međusobno tako da se dobije stanje $|1\rangle$

$$\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle = |1\rangle$$

Pošto je bazno stanje $|1\rangle$ superpozicija za data sopstvena stanja, to znači da možemo da uradimo *kvantnu procenu faze* nad U koristeći se stanjem $|1\rangle$, u tom slučaju će izmerena faza Φ biti:

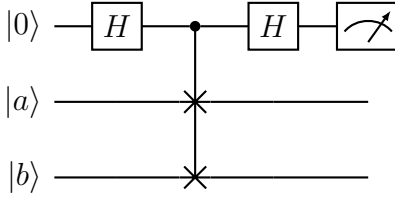
$$\Phi = \frac{s}{r}$$

3.3.2 SWAP Test

Ova rutina je jednostavan kvantni algoritam koji izražava skalarni proizvod za dva ulazna kvantna stanja $|a\rangle$ i $|b\rangle$ [7].

Verovatnoća da se pri merenju kontrolnog kubita dobije stanje $|0\rangle$ je definisana kao:

$$P(|0\rangle) = \frac{1}{2} + \frac{1}{2} F(|a\rangle, |b\rangle)$$



Slika 2: Dijagram kvantnog kola za SWAP test. $|a\rangle$ i $|b\rangle$ su ulazna stanja, $|0\rangle$ je dodatni kubit sa kojeg se čita rezultat [13]

gde je $F(|a\rangle, |b\rangle) = |\langle a | b \rangle|^2$ validnost između dva kvantna stanja. Verovatnoća $P(|0\rangle) = 0.5$ znači da su kvantna stanja $|a\rangle$ i $|b\rangle$ međusobno ortogonalna, a verovatnoća $P(|0\rangle) = 1$ znači da su kvantna stanja identična. Ova rutina treba da se ponavlja više puta da bi se dobila dobra procena vrednosti validnosti.

SWAP test se može koristiti za izračunavanje Euklidove distance između kvantnih stanja u više dimenzionom prostoru, kao i u velikom broju kvantnih algoritama.

4 Kvantno mašinsko učenje

Kvantno mašinsko učenje je spoj kvantnih računara i mašinskog učenja. U programima Kvantnog mašinskog učenja koriste se kvantni algoritmi (npr. qBLAS algoritmi, SWAP test) kao deo metoda optimizacije slične klasičnim metodama mašinskog učenja.

Prema vrsti podataka koji se obrađuju oblast možemo da delimo na dve podoblasti:

1. Obrada klasičnih podataka na kvantnim mašinama (**Mašinsko učenje dopunjeno kvantnim računarima** engl. Quantum-enhanced machine learning)
2. Obrada kvantnih podataka na kvantnim mašinama

Problem kod obrade klasičnih podataka na kvantnim mašinama jeste učitavanje podataka u sistem, kao i čitanje rezultata. Ovo dovodi da algoritmi sa teorijskim eksponencijalnim ubrzanjem, u realnom svetu budu dosta sporiji i fizički zahtevniji (veličina kvantnog kola zna da poraste i na skalu oko 10^{25} za jednostavnu implementaciju HHL algoritma) [3].

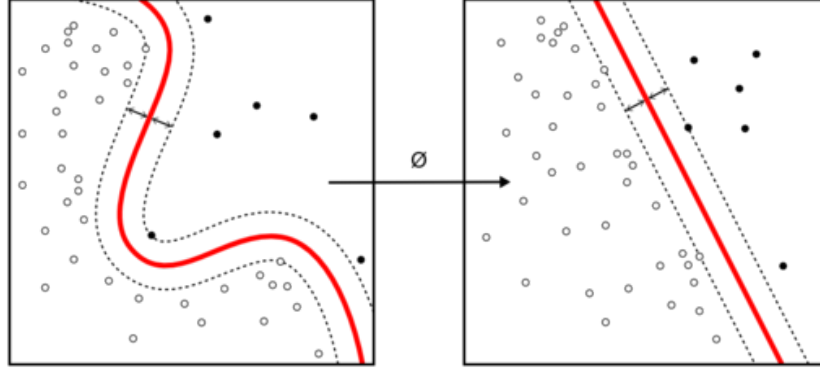
4.1 Kvantna metoda potpornih vektora

Jedan od najjednostavnijih primera metoda Kvantnog mašinskog učenja jeste **Kvantna metoda potpornih vektora** (engl. Quantum Support-vector machine - QSVM). Klasičan SVM je metoda koja pronalazi optimalnu podelu hiper-ravni između dva različita skupa podataka, tako da sa velikom verovatnoćom svi podaci iz jednog skupa podataka će se naći na jednoj polovini hiper-ravni. [3]

4.1.1 Klasičan algoritam

Ova metoda određuje klase koristeći linearnu funkciju $w^T x + b$. SVM predviđa prvu klasu ako je izlaz funkcije pozitivan, a predviđa drugu klasu ako je izlaz negativan. Pošto kod većine slučajeva pripadnost između dve klase podataka nije odvojiva linearnom hiper-ravni, sa SVM metodom koristi se i **Kernel metoda**.

Pronalaženje optimalne hiper-ravni se sastoji od minimizovanja $|w|^2/2$ u nejednačini $y_j(w * x_j + b) \geq 1$ za svako j . Ovo minimizovanje se može uraditi, ako uvedemo Karuš-Kun-Taker množioce (engl. Karush-Kuhn-Tucker



Slika 3: Ilustracija kernela za support vector mašinu [4]

multiplier) $\vec{\alpha} = (\alpha_1, \dots, \alpha_M)$ i maksimizujemo ih nad Lagranžovoj funkcijom:

$$L(\vec{\alpha}) = \sum_{j=1}^M y_j \alpha_j - \frac{1}{2} \sum_{j,k=1}^M \alpha_j \alpha_k x_j x_k$$

sa sledećim ograničenjima $\sum_{j=1}^M \alpha_j = 0$ i $\forall j \leq M \ y_j \alpha_j \geq 0$. Tako da, parametri za hiper-ravan se izvide kao: $w = \sum_{j=1}^M \alpha_j x_j$ i $b = y_j - w x_j$ (za one j gde važi da $\alpha_j \neq 0$). Mali broj α_j je različito od nule, takve promenljive se odnose na vektore x_j koji leže na ravni, ti vektori se zovu **Potporni vektori** (engl. Support vectors) [15].

Kernel metoda transformiše podatke u prostor gde su dve klase linearno odvojive. Metoda se oslanja na to da se linearna funkcija može zapisati isključivo kao skalarni proizvod između primera.

$$w^T x + b = b + \sum_{i=1}^m \alpha_i x^T x_i$$

gde je x_i trening primer, a α je vektor koeficijenata. Ovako zapisivanje funkcije nam dozvoljava da zamenimo x sa izlazom funkcije $\phi(x)$, a skalarni proizvod sa funkcijom $k(x, x_i) = \phi(x) * \phi(x_i)$. Funkcija k se zove **kernel**, dok funkcija ϕ je funkcija koja preslikava podatke iz jednog prostora u drugi. Operator $\langle * \rangle$ predstavlja unutrašnji proizvod ekvivalentno $\phi(x)^T \phi(x_i)$. [8]

Kada zamenimo skalarni proizvod sa kernelom, funkciju predikcije možemo da zapišemo kao

$$f(x) = b + \sum_i \alpha_i k(x, x_i)$$

Jedna od velikih mana kernel metode jeste da, cena procene izlaza kernel funkcije je linearna u odnosu na broj trening primera, jer i -ti indeks označava člana $\alpha_i k(x, x_i)$ kernel funkcije. [8]

Složenost SVM je $O(\log(1/\epsilon)M^2(N + M))$, gde je ϵ preciznost rešenja, N je broj dimenzija prostora nad kojim radimo, a M je broj trening primera.

Takođe krajnje rešenje je binarni klasifikator za neki vektor x :

$$y(x) = \text{sign}\left(\sum_{j=1}^M \alpha_j k(x, x_j) + b\right)$$

4.1.2 Kvantni algoritam

Pretpostavimo da imamo metodu za treniranje (engl. Oracle) koja vraća norme $|x_j|$, oznake y_j i kvantne vektore $|x_j\rangle = \frac{1}{|x_j|} \sum_{k=1}^N (x_j)_k |k\rangle$.

Bitno nam je za algoritam da ova metoda vraća podatke pod donjom granicom, da bi se kompleksnost jezgra algoritma mogla iskazati. Koristeći procenu unutrašnjeg proizvoda za pripremu kernel matrice, može se dobiti SVM algoritam kompleksnosti $O(\log(1/\epsilon)M^3 + M^2 \log(N/\epsilon))$. Kernel matrica je od velike važnosti za preformulaciju algoritma u vidu funkcije kvadratnog troška. Uvodimo pojednostavljanje za nejednakosti, tako što uvodimo promenljivu e_j i koristimo osobinu oznaka da $y_j^2 = 1$

$$y_j(w \cdot x_j + b) \geq 1 \rightarrow (w \cdot x_j + b) = y_j - y_j e_j$$

Pored ove jednačine imamo i implicitan uslov Lagranžove funkcije da sadrži kaznenu (engl. penalty) promenljivu $\gamma/2 \sum_{j=1}^M e_j^2$, gde je definisan parametar γ za relativnu težinu greške treniranja. Ako uzmemo parcijalni izvod od Lagranžove funkcije i eliminišemo promenljive u i e_j dolazi do aproksimacije funkcije kvadratnog troška problema:

$$F \begin{bmatrix} b \\ \vec{\alpha} \end{bmatrix} \equiv \begin{bmatrix} 0 & \vec{1}^T \\ \vec{1} & K + \gamma^{-1} \vec{1} \end{bmatrix} \begin{bmatrix} b \\ \vec{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \vec{y} \end{bmatrix}$$

Ovde $K_{ij} = x_i^T \cdot x_j$ je simetrična kernel matrica, $y = (y_1, \dots, y_m)$ kao i $\vec{1} = (1, \dots, 1)$. Matrica F je dimenzija $(M + 1) \times (M + 1)$. Dodatna dimenzija (red i kolona) se sastoji od jedinica, zbog pomeraja b . Promenljiva α_j ima ulogu određivanja distance od optimalnog rešenja. Tako da rešenje, odnosno

pronalaženje promenljivih za SVM je oblika:

$$\begin{bmatrix} b \\ \vec{\alpha} \end{bmatrix} = F^{-1} \begin{bmatrix} 0 \\ \vec{y} \end{bmatrix}$$

U klasičnom algoritmu kompleksnost SVM sa funkcijom kvadratnog troška je $O(M^3)$.

U kvantnom algoritmu, zadatak je generisanje stanja $|b, \vec{\alpha}\rangle$ koja opisuju hiper-ravan i onda klasifikuju stanja $|x\rangle$. U algoritmu, rešavamo normalizovanu jednačinu $\hat{F}|b, \vec{\alpha}\rangle = |y\rangle$, gde je $\hat{F} = F/\text{tr}F$ sa ograničenjem $\|F\| \leq 1$. Klasa će biti određena kao verovatnoća uspeha pri SWAP testu između $|b, \vec{\alpha}\rangle$ i $|x\rangle$. Za izračunavanje inverzne matrice kvantnim algoritmom, matrica \hat{F} mora da se razdvoji na jednostavne elemente. Tako da, matrica \hat{F} može da se razdvoji na sledeće elemente $\hat{F} = (J + K + \gamma^{-1}\mathbb{1})/\text{tr}F$, gde je matrica

$$J = \begin{bmatrix} 0 & \vec{1}^T \\ \vec{1} & 0 \end{bmatrix}$$

Takođe, za procenu faze pravimo formulaciju Lijeovog proizvoda $e^{-i\hat{F}\Delta t} = e^{-i\gamma^{-1}\mathbb{1}\Delta t/\text{tr}F} e^{-iJ\Delta t/\text{tr}F} e^{-iK\Delta t/\text{tr}F} + O(\Delta t)$. Za njega važi da ima dve sopstvene vrednosti oblika $\lambda_{\pm} = \pm\sqrt{M}$, a istovetno, sopstvena stanja su oblika $|\lambda_{\pm}\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm \frac{1}{\sqrt{M}} \sum_{k=1}^M |k\rangle)$. Za matricu $\gamma^{-1}\vec{1}$ dve sopstvene vrednosti su $v_1 = 0$ i $v_2 = \gamma^{-1}M$. Sada možemo da aproksimiramo fazu za $e^{-i\hat{F}\Delta t}$.

Prvi korak, stanje $|y\rangle$ može da se transformiše u sopstveno stanje $|u_j\rangle$ matrice \hat{F} , koja ima sopstvenu vrednost λ_j . Ono je oblika $|y\rangle = \sum_{j=1}^{M+1} \langle u_j|y\rangle |u_j\rangle$. Ako inicijalizujemo aproksimaciju sopstvene vrednosti na $|0\rangle$, i primenimo procenu faze nad stanjem dobićemo stanje bliže pravoj sopstvenoj vrednosti:

$$|y\rangle |0\rangle \rightarrow \sum_{j=1}^{M+1} \langle u_j|y\rangle |u_j\rangle |\lambda_j\rangle \rightarrow \sum_{j=1}^{M+1} \frac{\langle u_j|y\rangle}{\lambda_j} |u_j\rangle$$

Drugi korak, invertujemo dobijeno stanje sopstvene vrednosti, pozivajući rotaciju stanja. Na kraju dobijamo novo stanje sa traženim parametrima SVM ($C = b^2 + \sum_{k=1}^M \alpha_k^2$)

$$|b, \vec{\alpha}\rangle = \frac{1}{\sqrt{C}}(b|0\rangle + \sum_{k=1}^M \alpha_k |k\rangle)$$

Klasifikacija Sada imamo trenirani model kvantnog SVM-a i želimo da klasifikujemo stanje $|x\rangle$. Od stanja $|b, \vec{\alpha}\rangle$, korišćenjem metode za treniranje, konstruišemo stanje:

$$|\tilde{u}\rangle = \frac{1}{\sqrt{N_u}}(b|0\rangle|0\rangle + \sum_{k=1}^M \alpha_k |x_k\rangle|k\rangle|x_k\rangle)$$

gde nam je $N_u = b^2 + \sum_{k=1}^M \alpha_k^2 |x_k|^2$. Pored ovoga konstruišemo i ulazno stanje $|\tilde{x}\rangle$:

$$|\tilde{x}\rangle = \frac{1}{\sqrt{N_x}}(|0\rangle|0\rangle + \sum_{k=1}^M |x\rangle|k\rangle|x\rangle)$$

gde nam je $N_x = M|x|^2 + 1$. Konstruišemo dva nova stanja $|\psi\rangle$ i $|\phi\rangle$; $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle|\tilde{u}\rangle + |1\rangle|\tilde{x}\rangle)$ i $|\phi\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. Merenjem SWAP testa, verovatnoća dobijanja pozitivne vrednosti je $P = |\langle\psi|\phi\rangle|^2 = \frac{1}{2}(1 - \langle\tilde{u}|\tilde{x}\rangle)$. Ovde dobijamo unutrašnji proizvod, odnosno $\langle\tilde{u}|\tilde{x}\rangle = \frac{1}{\sqrt{N_x N_u}}(b + \sum_{k=1}^M \alpha_k |x_k| |x| \langle x_k|x\rangle)$, koji se obično izračunava u $O(1)$ na kvantnom računaru. Ako hoćemo preciznost ϵ , treba da ponavljamo algoritam merenja $O(P(1-P)/\epsilon^2)$ puta [15].

4.2 Kvantna analiza glavnih komponenti

Ovo je metoda koja se koristi za smanjivanje dimenzija vektora podataka gde nam je bitno da sačuvamo što više informacije o podatku - labava kompresija (engl. lossy compression).

4.2.1 Klasičan algoritam

Neka svaku tačku $x^{(i)} \in \mathbb{C}^n$ želimo da transformišemo u tačku $c^{(i)} \in \mathbb{C}^l$ gde je $l < n$. Želimo da nađemo funkciju kodiranja koja za ulaz x vraća c , odnosno, $f(x) = c$. Takođe želimo da nađemo funkciju dekodiranja $g(f(x)) \approx x$.

Zbog jednostavnosti, uzećemo funkciju množenja matrica kao funkciju dekodiranja. Neka je $g(c) = Dc$, gde je $D \in \mathbb{C}^{n \times l}$ matrica definisana za dekodiranje. Takođe zbog optimalnog izračunavanja funkcije kodiranja, PCA uvodi ograničenje da su kolone međusobno ortogonalne. Još jedno ograničenje koje može da se uvede u algoritam, koje će dovesti do jedinstvenog rešenja, jeste da su sve kolone matrice D u unitarnoj normi. Jedan od načina na koji hoćemo da nađemo optimalnu projekciju c za ulaz x jeste

da nađemo najmanju $L2$ distancu između ulaza x i dekodovane vrednosti $g(c)$

$$c^* = \underset{c}{\operatorname{argmin}} \|x - g(c)\|_2^2.$$

Ovo pronalaženje minimalne distance će dovesti do optimalnog rešenja problema $c = D^T x$ [8]. Tako da funkcija enkodovanja je oblika:

$$f(x) = D^T x$$

Takođe, možemo da uvedemo novu funkciju rekonstrukcije ulaza x

$$r(x) = g(f(x)) = DD^T x$$

Sada treba da se nađe optimalna matrica D . Ovo će se rešiti na isti način kao i pronalaženje optimalnog c za ulaz x , odnosno kao pronalaženje minimalne $L2$ distance za ulazne vektore njihove rekonstrukcije

$$D^* = \underset{D}{\operatorname{argmin}} \sqrt{\sum_{i,j} (x_j^i - r(x^i)_j)^2} \text{ gde važi } D^T D = I_l$$

Posle procesa izvođenja [8], jednačina za optimalnu matricu D je oblika:

$$D^* = \underset{D}{\operatorname{argmin}} \operatorname{Tr}(D^T X^T X D) \text{ gde važi } D^T D = I_l$$

gde nam je $X \in C^{m \times n}$ matrica gde su redovi ulazni vektori x . Ova jednačina se može rešiti koristeći dekompoziciju sopstvenih stanja, gde bi se pronašli sopstveni vektor za $X^T X$ za najveću sopstvenu vrednost.

4.2.2 Kvantni algoritam

U kvantnom algoritmu bitno nam je da nađemo sopstvene vektore i sopstvene vrednosti za ulaz. Ovo se dosta oslanja na drugi deo metode koji je opisan u sekciji posvećena *Kvantnoj metodi potpornih vektora*. Ako izaberemo nasumičan vektor v_j iz skupa ulaznih vektora, kreiramo kvantno stanje $|v_j\rangle$; tada možemo da kreiramo matricu gustine $\rho = (1/N) \sum_j |v\rangle \langle v|$ gde je N veličina skupa vektora. [3] Slično **qSVM** nad matricom gustine ρ možemo da apliciramo algoritam procene faze stanja. Odnosno, da primenimo $e^{-i\rho t}$, t puta nad inicijalnim stanjem:

$$|v_j\rangle |0\rangle \rightarrow \sum_i \psi_i |\chi_i\rangle |\tilde{r}_i\rangle$$

gde je $|\chi_i\rangle$ sopstveni vektor od matrice ρ , \tilde{r}_i procena sopstvene vrednosti, a $\psi_i = \langle \chi_i | v_j + \rangle$. I primenom SWAP testa nad dobijenim stanjem, dobijamo stanje:

$$\sum_i r_i |\chi_i\rangle \langle \chi_i| \otimes |\tilde{r}_i\rangle \langle \tilde{r}_i|$$

Merenjem ovog stanja mi dobijamo sopstvenu vrednost i sopstveni vektor za matricu gustine ρ . Ako uradimo ovaj proces nad većim brojem kopija matrice ρ , dobićemo preciznije procene sopstvene vrednosti i sopstvenih vektora. Sada kada imamo sopstvenu vrednost i sopstveni vektor možemo da rekonstruišemo matricu za kodiranje D . Vremenska složenost ovog algoritma je $O(\log d)$ [10].

4.3 Kvantna neuralna mreža

Neuralne mreže su osnova polja koje se naziva **Duboko učenje** i zato postoji velika pažnja za njihov razvoj. U radu [6], autori su predstavili osnove algoritama za Kvantnu neuralnu mrežu (QNN). U radu su dati neki primeri, neke prednosti i neki nedostaci kvantnog pristupa neuralnim mrežama.

Neka imamo unarni skup stringova ϕ oblika $z = z_1 z_2 \dots z_n$ gde svako z_i je bit čija vrednost može da bude $+1$ ili -1 , kao i binarne oznake $l(z)$ koje mogu da bude $+1$ ili -1 . Zbog jednostavnosti neka se u našem skupu nalaze sve permutacije ovako opisanog stringa, to jest neka $|\phi| = 2^n$. Predstavimo kvantni proces koji radi na $n + 1$ kubita (poslednji kubit služi kao izlaz procesa). Kvantni proces se sastoji od unitarnih transformacija ulaznih stanja: $U_a(\theta)$. Svaka transformacija radi nad podskupom ulaznih kubita i zavisi od promenljive θ . Izabraćemo podskup od L transformacija:

$$\mathbf{U}(\vec{\theta}) = U_L(\theta_L) U_{L-1}(\theta_{L-1}) \dots U_1(\theta_1)$$

koje zavise od L parametara $\vec{\theta} = \theta_L \theta_{L-1} \dots \theta_1$. Za svaki string z kreiraćemo početno stanje:

$$|z, 1\rangle = |z_1, z_2, \dots z_n, 1\rangle$$

Primenjivanje unitarne transformacije vraća stanje: $U(\vec{\theta}) |z, 1\rangle$. Na izlazu meri se dodati kubit sa Puali-jevim operatorom σ_y , koji se kasnije naziva i Y_{n+1} . Tako da na kraju imamo izlaz $+1$ ili -1 . Cilj je isti kao i kod klasičnih neuralnih mreža, a to je da se nauči proces da vraća tačne vrednosti za dati ulazni string. Pošto merenje izlaznog kubita nije sigurno, odnosno

merenjem kubita dobijamo tačnu vrednost sa nekom verovatnoćom, uvodimo transformaciju:

$$\langle z, 1 | U^T(\vec{\theta}) Y_{n+1} U(\vec{\theta}) | z, 1 \rangle$$

koja predstavlja prosečnu vrednost merenja, ako Y_{n+1} merimo na više kopija originalnog izlaza.

Ovde, kao i u klasičnoj neuralnoj mreži, cilj nam je da nađemo parametar $\vec{\theta}$ koja vraća tačnu vrednost sa velikom preciznošću. Slično prethodnoj postavci imamo: L unitarnih promenljivih sa korespondentnim promenljivama $\vec{\theta}$, kao i ulazni string z ; tada možemo da predstavimo funkciju troška kao:

$$\text{loss}(\vec{\theta}, z) = 1 - l(z) \langle z, 1 | U^T(\vec{\theta}) Y_{n+1} U(\vec{\theta}) | z, 1 \rangle$$

Možemo primetiti da ova funkcija troška je linearna i da je minimum u 0, jer je vraćena vrednost između -1 i $+1$. Pretpostavimo da kvantna neuralna mreža radi savršeno, tako da za svaki ulazni string z , merenje uvek vraća tačnu oznaku. To onda znači da optimalna promenljiva $\vec{\theta}$ postoji i da je minimum za funkciju troška u 0 za sve ulaze z .

Neka imamo skup stringova S za treniranje, sa njihovim oznakama. Postoji kvantni proces koji ima mogućnost da prikaže tražene oznake i zavisi od parametara $\vec{\theta}$. Opisćemo proces kako da dođemo do optimalnih parametara $\vec{\theta}$. Neka počnemo sa nasumičnom promenljivom $\vec{\theta}$ (ili ako imamo neku pretpostavku početne vrednosti parametara). Izaberimo neki string z^1 iz skupa za treniranje. Primenujemo kvantni proces nad izabranim stringom:

$$U(\vec{\theta}) | z, 1 \rangle$$

i merimo Y_{n+1} na zadnjem kubit. Nakon nekoliko merenja možemo da imamo dobru aproksimaciju očekivane vrednosti od Y_{n+1} i tada izračunavamo $\text{loss}(\vec{\theta}, z^1)$. Nakon toga, želimo da promenimo parametar $\vec{\theta}$ tako da smanjimo funkciju troška za string z^1 . Postoje dva načina da se uradi traženo: (1) da uradimo pomeraj ka nekom izabranom uzorku u $[\vec{\theta} - \epsilon, \vec{\theta} + \epsilon]$ intervalu, (2) da izračunamo izvod funkcije troška po $\vec{\theta}$ i da se malo pomerimo ka pravcu koji minimizuje funkciju. Ovo nam daje novi parametar $\vec{\theta}^1$. Sada biramo ponovo iz skupa neki string z^2 i ponavljamo prethodni proces ali sa parametrom $\vec{\theta}^1$. Ovako dobijamo novi parametar $\vec{\theta}^2$ koji ima manju funkciju troška za string z^2 nego parametar $\vec{\theta}^1$. Ovako prolazimo kroz proces sve dok ne prođemo kroz ceo skup S . Kao rezultat ovoga generisana je sekvenca

parametara $\vec{\theta}^1, \vec{\theta}^2, \dots, \vec{\theta}^S$. Ako je učenje parametara uspešno, onda bi bilo dobijeno da operator $U(\vec{\theta}^S)$, kada se primeni na stanju $|z, 1\rangle$, vraća stanje koje kada se izmeri na izlazu vraća tačnu oznaku $l(z)$. Ako je z iz skupa za treniranje, reći ćemo da je model naučio podatke za treniranje. Ako je z izvan skupa za treniranje, možemo reći da je model naučio da generalizuje i za neviđene podatke.

Ovaj proces koji je opisan, može se primetiti, u klasičnom mašinskom učenju zove se *stohastičko učenje*. U tradicionalnom mašinskom učenju sa neuralnim mrežama, parametri se prikazuju kao promenljive unutar matrice, koja je linearna u odnosu na unutrašnje vektore. Nad komponentama tih vektora vrše se nelinearne transformacije, pre nego što se pomnože sa ostalim parametrima. Uvođenje dobre nelinearnosti je jedan od glavnih delova uspešne implementacije modela u klasičnom mašinskom učenju. Ovu osobinu klasičnih neuralnih mreža teško je prebaciti u kvantni sistem, jer je kvantna mehanika, osnova celog koncepta kvantnog računarstva, sama po sebi linearna. U metodi koja je opisana, svaka unitarna operacija se izvršava nad izlazom prethodne operacije, pri čemu se između operacija ne izvršava nikakva nelinearna transformacija. Neka nam je svaka unitarna transformacija oblika $e^{i\theta\Sigma}$, gde je Σ proizvod tenzora koji se sastoji iz skupa Paulijevih operatora, koji rade nad nekolicinom kubita. Izvod operatora po $\vec{\theta}$ je ograničen po L , to jest po broju parametara. Ovo je značajno, jer znači da gradijent ne može da ode u beskonačnost i tako izbegavamo veliki problem koji se može desiti klasičnim neuralnim mrežama.

4.3.1 Reprezentacija modela

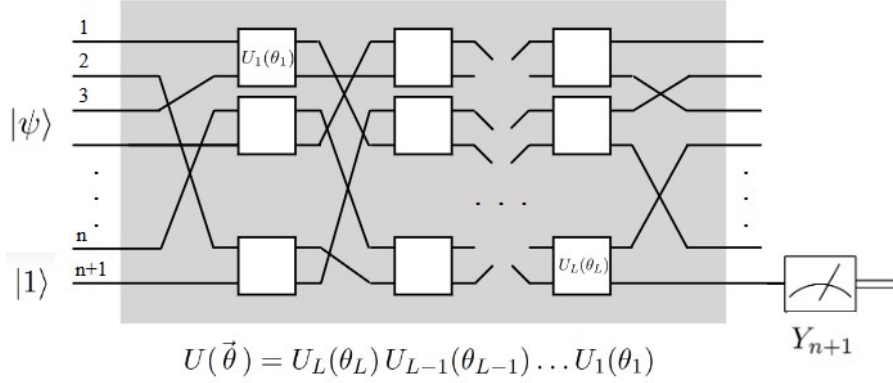
Neka imamo 2^n , n -bitnih stringova i vezano za njih $2^{(2^n)}$ funkcija oznaka $l(z)$. Ako nam je data određena funkcija oznaka, onda možemo da definišemo operator nad baznim stanjima kao:

$$U_l |z, z_{n+1}\rangle = e^{i\frac{\pi}{4}l(z)X_{n+1}} |z, z_{n+1}\rangle$$

Ovaj operator rotira ulazni kubit oko x -ose za $\frac{\pi}{4}$ puta oznaka za string z . Tako da iz toga imamo:

$$U_l^T Y_{n+1} U_l = \cos(\frac{\pi}{4}l(Z))Y_{n+1} + \sin(\frac{\pi}{4}l(Z))Z_{n+1}$$

gde u formuli, $l(Z)$ je interpretirana kao operator dijagonalan u odnosu na bazna stanja. Takođe, pošto funkcija oznaka $l(z)$ može da vrati ili $+1$ ili



Slika 4: Ilustracija kvantne neuralne mreže [6]

-1 , iz toga imamo $\langle z, 1 | U_l^T Y_{n+1} U_l | z, 1 \rangle = l(z)$. Ovo nam pokazuje da bar na nekom apstraktnom nivou imamo mogućnost da predstavimo bilo koju funkciju oznake kao kvantno kolo.

Sada ćemo objasniti kako da se napiše operator U_l kao proizvod dve kubit unitarne transformacije. Zbog ovoga, treba da se pređe na *boolean* promenljive $b_i = \frac{1}{2}(1 - z_i)$ i neka funkcija oznake l bude oblika $1 - 2b$ gde je $b \in \{0, 1\}$. Sada možemo da iskoristimo **Reed-Muller** zapisivanje bilo koje *boolean* funkcije u obliku bitova $b_1 \dots b_n$:

$$b = a_0 \oplus (a_1 b_1 \oplus a_2 b_2 \oplus \dots a_n b_n) \oplus (a_{12} b_1 b_2 \oplus a_{13} b_1 b_3 \oplus \dots) \oplus \dots \oplus a_{12\dots n} b_1 b_2 \dots b_n$$

gde su koeficijenti $a \in \{0, 1\}$. Primećuje se da imamo 2^n koeficijenta, pošto su oni ili 0 ili 1, da stvarno ima $2^{(2^n)}$ mogućih *boolean* funkcija. Naša funkcija b može biti eksponencijalno dugačka. Sada možemo da zapišemo unitarnu transformaciju koja zavisi od funkcije oznaka kao:

$$U_l = e^{i\frac{\pi}{4} X_{n+1}} e^{-1\frac{\pi}{2} B X_{n+1}}$$

gde je B operator, dijagonalan u odnosu na bazna stanja, koji odgovara datoj funkciji b . Svaka vrednosti u B se množi sa X_{n+1} tako da svaka vrednost je komutativna sa ostalim vrednostima. Svaki član, različit od nule, u **Reed-Muller** formuli dovodi do kontrolnog *bit-flip*-a u U_l na izlaznom kubit.

Ovaj rezultat kvantnog predstavljanja ima analog u klasičnoj teoriji predstavljanja [5]. Ona pokazuje da bilo koja *boolean* funkcija oznake može da

se predstavi u neuralnoj mreži dubine tri, gde srednji slog ima veličinu 2^n . Ovako velika matrica ne bi mogla da se predstavi na klasičnim računarima, ali bi mogli na kvantnim računarima, koji po prirodi rade nad Hilbertovim prostorima sa eksponencijalnim dimenzijama. Ali još nije dokazano da svaka *boolean* funkcija može da se predstavi u kvantnom kolu koje nije eksponencijalne dubine.

Reprezentacija parnosti podskupa Neka imamo datu funkciju oznaka koja vraća parnost podskupa bitova datog stringa. Neka je podskup S i neka je $a_j = 1$ ako je bit j u podskupu i $a_j = 0$ ako j nije u podskupu. Reed-Muller formula za parnost podskupa je:

$$P_S(z) = \sum_j \oplus a_j b_j$$

Ovo nam dozvoljava da napravimo unitarnu transformaciju koja implementira parnost podskupa:

$$U_{P_S} = e^{i\frac{\pi}{4}X_{n+1}} e^{-i\frac{\pi}{2}\sum_j a_j B_j X_{n+1}}$$

Kolo se sastoji od, najviše, n operatora nad dva kubita koji su međusobno komutativni, gde je pridodati kubit u svim operatorima nad dva kubita.

4.3.2 Učenje modela

U ovoj podsekciji će biti objašnjene dve potencijalne metode kako da se menja parametar $\vec{\theta}$ tako da se funkcija troška smanjuje. Ako su nam dati parametri $\vec{\theta}$ i trening primer z , prvo procenjujemo vrednost troška od $loss(\vec{\theta}, z)$. Da bismo ovo uradili treba da napravimo više merenja Y_{n+1} za $U(\vec{\theta})|z, 1\rangle$. Da bismo ovo uspeali sa verovatnoćom većom od 99% za procenu od funkcije troška koja je u δ intervalu od prave vrednosti funkcije troška treba da napravimo najmanje $2/\delta^2$ merenja ($\delta \in (0, 1)$).

Nakon što procenimo vrednost funkcije troška želimo da izračunamo gradient funkcije troška u odnosu na $\vec{\theta}$. Jedan od načina jeste da menjamo jednu po jednu promenljivu u $\vec{\theta}$. Nakon svake promene treba da se izračuna $loss(\vec{\theta}', z)$, gde $\vec{\theta}'$ je različit od $\vec{\theta}$ za neku malu vrednost u jednoj promenljivoj. Ako bi se koristio simetričan izvod funkcije troška svaku promenljivu parametra bismo mogli da izračunamo do preciznosti η u oko $1/\eta^3$ merenja. Ovaj proces bi trebalo da se ponavlja L puta da bi se dobio puni gradient.

Alternativna strategija jeste da se menja svaka promenljiva gradijenta, što se koristi kada su sve unitarne transformacije oblika $e^{i\theta\Sigma}$. Ako posmatramo izvod za funkciju troška $loss(\vec{\theta}, z)$ za parametar θ_k , koji je vezan za transformaciju $U_k(\theta_k)$ (koja ima i generalni Pauli-jev operator Σ_k). Sada:

$$\frac{dloss(\vec{\theta}, z)}{d\theta_k} = 2Im(\langle z, 1 | U_1^T \dots U_L^T Y_{n+1} U_L \dots U_{k+1} \Sigma_k U_k \dots U_1 | z, 1 \rangle)$$

Ako primetimo da su Y_{n+1} i Σ_k unitarni operatori, tada definišemo unitarni operator:

$$\mathcal{U}(\vec{\theta}) = U_1^T \dots U_L^T Y_{n+1} U_L \dots U_{k+1} \Sigma_k U_k \dots U_1$$

tako da izvod možemo da zapišemo kao:

$$\frac{dloss(\vec{\theta}, z)}{d\theta_k} = 2Im(\langle z, 1 | \mathcal{U} | z, 1 \rangle)$$

$\mathcal{U}(\vec{\theta})$ se može posmatrati kao kvantno kolo koje sadrži $2L + 2$ unitarnih transformacija. Sada možemo da primenimo $\mathcal{U}(\vec{\theta})$ nad stanjem $|z, 1\rangle$. Ako koristimo dodatni kubit, možemo da merimo imaginarni deo izvoda funkcije. Započecemo sa stanjem $|z, 1\rangle \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ i primenićemo $i\mathcal{U}(\vec{\theta})$ nad dodatim kubitom vrednosti 1. Ovo kreira:

$$\frac{1}{\sqrt{2}}(|z, 1\rangle |0\rangle + i\mathcal{U}(\vec{\theta}) |z, 1\rangle |1\rangle)$$

Ako primenimo Hademardovu kapiju na dodatim kubitom dobijamo:

$$\frac{1}{2}(|z, 1\rangle + \mathcal{U}(\vec{\theta}) |z, 1\rangle |0\rangle) + \frac{1}{2}(|z, 1\rangle - i\mathcal{U}(\vec{\theta}) |z, 1\rangle |1\rangle)$$

Sada kada izmerimo dodatni kubit, verovatnoća da se dobije 0 je:

$$\frac{1}{2} - \frac{1}{2}Im(\langle z, 1 | \mathcal{U}(\vec{\theta}) | z, 1 \rangle)$$

tako da, ponavljanjem ovog merenja možemo da dobijemo dobru procenu imaginarnog dela stanja iz kojeg možemo da izvučemo procenu k -te komponente traženog gradijenta. Ovaj metod izbegava numeričke nepreciznosti prethodne strategije. Cena ove metode je potreba za dodatnim kubitom kao i za kvantnim kolom $2L + 2$ dubine.

Pošto smo izračunali gradijent, treba nam metod za izmenu $\vec{\theta}$. Neka je \vec{g} gradijent funkcije troška po parametru $\vec{\theta}$. Sada menjamo $\vec{\theta}$ u pravcu \vec{g} . Sa veličinom koraka γ imamo

$$\text{loss}(\vec{\theta} + \gamma \vec{g}) = \text{loss}(\vec{\theta}, z) + \gamma \vec{g}^2 + O(\gamma^2)$$

Pošto želimo da smanjimo trošak na 0 možemo da napravimo da:

$$\gamma = -\frac{\text{loss}(\vec{\theta}, z)}{\vec{g}^2}$$

Ovako nešto bi dovelo da trošak bude 0 za trenutni primer, ali može da dovede do loših efekata za ostale primere. Ovde se u klasičnom mašinskom učenju obično uvodi promenljiva, stepen učenja $r \in (0, 1]$ i onda imamo sledeće:

$$\vec{\theta} \rightarrow \vec{\theta} - r\left(\frac{\text{loss}(\vec{\theta}, z)}{\vec{g}^2}\right)\vec{g}$$

Deo uspešne implementacije mašinskog učenja je da racionalno odaberemo vrednost stepena učenja.

Učenje parnosti podskupa Za dati podskup \mathbb{S} , unitarna transformacija $U_{P_{\mathbb{S}}}$ može da prikaže parnost podskupa za sve ulazne stringove. Da bi se naučio skup unitarnih operacija koje zavise od parametara, sa tim da za svaki podskup postoje parametri koji opisuju $U_{P_{\mathbb{S}}}$, najlakši način da se ovo postigne jeste da se koriste n parametara

$$U(\vec{\theta}) = e^{i\frac{\pi}{4}X_{n+1}}e^{-i\sum_j^n \theta_j B_j X_{n+1}}$$

Ovde se vidi da je reprezentacija savršena kada je $\theta_j = \frac{\pi}{2}$ ako je j u podskupu i $\theta_j = 0$ ako j nije u podskupu. Posle eksperimenta sa malim brojem kubita gde su uspeli da nauče model, njihov argument je da sa povećanjem veličine sistema postaje nemoguće da se nauči kvantni model. Da bi to pokazali, izračunali eksplicitnu formulu za očekivanu vrednost za Y_{n+1}

$$\langle z, 1 | U^T(\vec{\theta}) Y_{n+1} U(\vec{\theta}) | z, 1 \rangle = \cos(2 \sum_j \theta_j b_j)$$

Sa oznakom $l(z)$, ovo može se ubaciti u funkciju troška, ali sada može da se izračuna prosek troška za sve 2^n stringove, jer imamo eksplicitnu formulu

za oznake i njihovo očekivanje. Postoji više verzija izračunate funkcije, koja zavisi od izlaza $n \bmod 4$ i koliko bitova se nalazi u podskupu \mathbb{S} . Za prikaz uzeli su primer gde je n deljiv sa 4 i skup \mathbb{S} sadrži svih n bitova. U tom slučaju prosečan trošak za sve stringove je

$$1 - \cos(\theta_1 + \theta_2 + \dots \theta_n) \sin(\theta_1) \sin(\theta_2) \dots \sin(\theta_n)$$

Iz formule se vidi da je u minimumu kada su sve $\theta = \frac{\pi}{2}$. Zamislite kakva bi bila pretraga minimuma (pored ovog primera) funkcije nad intervalom $[0, \pi]^n$. Funkcija bi samo prikazivala vrednosti eksponencijalno blizu 1, sem u eksponencijalno malim intervalima oko optimalnih uglova. Isto tako gradijent bi bio veoma mali sem oko optimalnih uglova. Zato čak i ako imamo pristup prosečnom trošku, nijedan metod koji se oslanja na gradijentni pristup ne bi mogao biti korišćen za pronalaženje optimalnog ugla za bilo koji primer sa povećim n , gradijent bi radio izvan preciznosti mašine u tom slučaju.

4.3.3 Učenje osobina kvantnih stanja

Sa kvantnom neuralnom mrežom, očekuje se da na ulazu može da ima bilo koje kvantno stanje (koje nije izvedeno iz nekog klasičnog podatka) i da može da nauči neke njegove osobine i da ih izbaci u obliku nekih oznaka. Ne postoji ni jedna klasična neuralna mreža koja može to da uradi, jer klasični računari ne mogu da prihvate kvantno stanje kao ulaz. Osnovna ideja je da se n -kubitno stanje $|\psi\rangle$ ubaci u kvantnu neuralnu mrežu sa dodatim kubitom, koji služi za čitanje rezultata, koji je postavljen na 1. Neka nam je data unitarna transformacija $U(\vec{\theta})$ tako da imamo stanje

$$U(\vec{\theta}) |\psi, 1\rangle$$

i onda merimo Y_{n+1} . Cilj ovoga je da namestimo da izlaz ovog merenja bude ekvivalentan nekim dvema oznakama koje označavaju neke osobine kvantnog stanja. To je prikazano u sledećem primeru. Posmatrajmo Hamiltonov operator H (engl. Hamiltonian), koji je suma lokalnih vrednosti sa dodatnom osobino da ima i pozitivne i negativne sopstvene vrednosti. Sa datim kvantnim stanjem $|\psi\rangle$, obeležava se oznakom koja pokazuje da li je očekivana vrednost Hamiltonovog operatora pozitivna ili negativna:

$$l(|\psi\rangle) = \text{sign}(\langle\psi| H |\psi\rangle)$$

Posmatrajmo operator $U_H(\beta) = e^{i\beta H X_{n+1}}$, gde je β mala i pozitivna vrednost. Sada

$$\langle \psi, 1 | U_H^T(\beta) Y_{n+1} U_H(\beta) | \psi, 1 \rangle = \langle \psi | \sin(2\beta H) | \psi \rangle$$

tako da za dovoljno malo β ovo je približno jednako $2\beta \langle \psi | H | \psi \rangle$ i tako imamo znak očekivane vrednosti za predikciju oznake, koja je jednaka tačnoj oznaci. Ovako prikazana, ovo je funkcija oznake sa kvantnim kolima koja ima malu grešku. Mala greška dolazi iz toga što je $\langle \psi | \sin(2\beta H) | \psi \rangle$ samo približno jednako $2\beta \langle \psi | H | \psi \rangle$. Ako uzmemo da β bude dosta manje od $1/\|H\|$ (inverzno od norme matrice H), možemo da napravimo da nam greška bude mala.

Posmatrajmo graf gde na svakoj ivici imamo ZZ uparivanje sa koeficijentom ili $+1$ ili -1 . Hamiltonov operator je oblika: $H = \sum_{i,j} J_{ij} Z_i Z_j$ gde prvobitna suma je ograničena na ivice grafa i J_{ij} je ili $+1$ ili -1 . Neka postoje M vrednosti u H . Prvo, treba da izaberemo M uglova θ_{ij} i neka je kvantno kolo koje implementira transformaciju oblika:

$$U(\vec{\theta}) = e^{i \sum_{i,j} J_{ij} Z_i Z_j X_{n+1}}$$

Ako izaberemo $\theta_{ij} = \beta J_{ij}$ imamo operator $U_H(\beta)$ koji osigurava da možemo da označimo traženu oznaku ako izaberemo malo β .

Kvantno stanje $|\psi\rangle$ je u Hilbertovom prostoru sa 2^n dimenzija i ne možemo da očekujemo da naučimo oznake za svako stanje. Posmatrani Hamiltonov operator ima bitovsku strukturu, tako da možemo da se ograničimo kvantna stanja sa bitskim strukturama. Tako je predloženo da se za treniranje koriste stanja samo sa ovom formom, kao i za testiranje.

5 Zaključak

U ovom radu je prikazano kraći uvod u kvantno računarstvo i kvantno mašinsko učenje. Takođe je dato nekoliko primera metoda kvantnog mašinskog učenja, koji poboljšavaju svoje klasične ekvivalente. Glavni izazov trenutno je praktična primena, koja je strogo vezana za razvoj funkcionalnih kvantnih računara.

Ostavljeno je dosta metoda kvantnog mašinskog učenja, koje čitalac može da istraži, ali su ove prikazane verovatno dovoljne za razumevanje oblasti. Trenutak kada se dokaže kvantna nadmoć dovešće do većeg zainteresovanja ka svim oblastima povezane za kvantno računarstvo, uključujući i kvantno mašinsko učenje. Svakako se savetuje praćenje daljeg razvoja ove oblasti, pošto je ona relativno mlada i ima dosta inovacija i poboljšavanja.

References

- [1] Scott Aaronson. *Quantum computing since Democritus*. Cambridge University Press, 2013.
- [2] Leonard M Adleman, Jonathan Demarrais, and Ming-Deh A Huang. “Quantum computability”. In: *SIAM Journal on Computing* 26.5 (1997), pp. 1524–1540.
- [3] Pancotti N. et al. Biamonte J. Wittek P. “Quantum machine learning”. In: *Nature* 549 (2017).
- [4] Wikimedai Commons. *Kernel Machine.png*. 2011. URL: https://commons.wikimedia.org/wiki/File:Kernel_Machine.png.
- [5] George V. Cybenko. “Approximation by superpositions of a sigmoidal function”. In: *Mathematics of Control, Signals and Systems* 2 (1989), pp. 303–314.
- [6] Edward FarZXhi and Hartmut Neven. “Classification with Quantum Neural Networks on Near Term Processors”. In: (2018). DOI: 10.48550/ARXIV.1802.06002. URL: <https://arxiv.org/abs/1802.06002>.
- [7] Dmitriy V Fastovets et al. “Machine learning methods in quantum computing theory”. In: *International Conference on Micro-and Nano-Electronics 2018*. Vol. 11022. SPIE. 2019, pp. 752–761.
- [8] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. Adaptive Computation and Machine Learning series. MIT Press, 2016. ISBN: 9780262035613. URL: <https://books.google.rs/books?id=Np9SDQAAQBAJ>.
- [9] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. “Quantum algorithms for supervised and unsupervised machine learning”. In: *arXiv preprint arXiv:1307.0411* (2013).
- [10] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. “Quantum principal component analysis”. In: *Nature Physics* 10.9 (July 2014), pp. 631–633. DOI: 10.1038/nphys3029. URL: <https://doi.org/10.1038%2Fnphys3029>.
- [11] Dan C. Marinescu. *Classical and quantum information*. Academic Press, 2012. Chap. 8. ISBN: 9780123838742; 0123838746.
- [12] Dan C. Marinescu. *Classical and quantum information*. Academic Press, 2012. Chap. 5. ISBN: 9780123838742; 0123838746.

- [13] Christine Middleton. *What's under the hood of a quantum computer?* 2021. URL: <https://physicstoday.scitation.org/doi/10.1063/pt.6.1.20210305a/full/>.
- [14] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.
- [15] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. “Quantum support vector machine for big data classification”. In: *Physical review letters* 113.13 (2014), p. 130503.
- [16] *Shor's Algorithm*. 2021. URL: <https://qiskit.org/textbook/ch-algorithms/shor.html>.
- [17] Vlatko Vedral. *Introduction to Quantum Information Science*. Oxford University Press, 2006.