

Kvantno masinsko učenje

Milan Bojic

Jun 2022

Sadržaj

1	Kvantno računarstvo	3
1.1	Osnovni pojmovi	3
1.2	Kvantno računarstvo	5
1.3	Kvantna informacija	5
1.3.1	Priprema podataka	6
2	Linearne algebra za kvantno masinsko učenje	6
3	Kvantna teorija kompleksnosti	7
3.1	SWAP Test	7
4	Kvantno masinsko učenje	8
4.1	Quantum support vector machine	8
4.1.1	Klasican algoritam	8
4.1.2	Kvantni algoritam	10
4.2	Quantum principal component analysis	12
4.2.1	Klasicni algoritam	12
4.2.2	Kvantni algoritam	13
4.3	Kvantna neuralna mreža	14
4.3.1	Reprezentacija modela	16
4.3.2	Učenje modela	17
4.3.3	Učenje osobina kvantnih stanja	20

1 Kvantno racunarstvo

Pre nego sto se pocne pricati o Kvantnom masinskom ucenju, treba objasniti neki osnovni pojmovi da bi lakse razumeli ostatak rada

1.1 Osnovni pojmovi

Potrebni pojmovi su:

- Kubit (eng. Qubit)
- Kvantna kola (eng. Quantum Gates)
- Kvantna uvezanost (eng. Quantum entanglement)
- Kvantna memorija, Kvantni registri

Kubit

Kubit (eng. 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 (koristeci "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:

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

Posto stanje kubita ima dva stepena slobode sto dovodi do toga da amplitude se mogu zapisati kao:

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

Takodje mozemo da vidimo da je $|\alpha|^2$ verovatnoca da se kubit nalazi u stanju 0, isto vazi i za $|\beta|^2$ i 1. Saznanje o tomo u kom stanju se nalazi kubit ce se dobiti merenjem kubita, tade ce da kubit izadje iz superpozicije i "pasce" u

stanje 1 ili stanje 0. U tom slučaju kubit će imati ponašanje kao i običan bit, ali ovako gubimo predjasnije kvantno stanje kubita. U fizičkom svetu kubit se može predstaviti kao polarizovani foton, pre čemu se dva stanja se uzimaju kao vertikalna i horizontalna polarizacija.

Kvantna kapija

Kvantna kapije (eng. Quantum Gates) su logički predstavljaju matricama i oni rade na određenom broju kubita. Matrice su unitarne sa oblikom $2^n \times 2^n$, gde je n broj qubita na kojim radimo. Neke od poznatih kola su: Hadamardovo kolo (stavlja kubit u superpoziciju), bit flip kolo (zamenjuje amplitude na kubit), 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 radiana.

Kvantna uvezanost

Kvantna uvezanost (eng. 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 amplituda:

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

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

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

Zanimljiva stvar kod uvazanih 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 mi bi smo dobili 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 uvezanih kubita i može da se predstavlja do 2^m vrednosti stanja istovremeno. Kvantna memorija su uređaji koji čuvaju kvantna stanja fotona, bez da uništavaju kvantnu informaciju koja se nalazi u fotonu. Ovakva memorija zahteva koherentni sistem materije, jer bi u suprotnom kvantna informacija unitar uređaja bila izgubljena zbog nekoherentnosti.

1.2 Kvantno računarstvo

Kvantno računarstvo je vrsta računarstva gde se koriste kolekcije fizickih osobina kvantne mehanike kao što su superpozicija i kvantna uvezanost, tako da se izvrši neka kalkulacija. Uređaja koji izvršava kvantne kalkulacije zovu se **kvantni računari**. Kvantni računari se sastoje od kvantnih kola i elementarnih kvantnih kapija koje služe za prenosenje i manipulisanje kvantnih informacija. [10]

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

1.3 Kvantna informacija

Kvantna informacija je informacija o stanju kvantnog sistema. O njihovim svojstvima bavi se **kvantna teorija informacije**. Takođe, kvantna informacija mogu izmeriti na isti način kao i klasična informacija koristeći se Šenonovom metodom. Odnosno, postoji jedinstveno merilo to jest funkcija nad kvantnim stanjem, koje je funkcija verovatnoće, kontinuiteta i sumiranja.[12] Ova funkcija se zove **von Neumann entropy** i za neki ulazni kubit ρ postoji ekvivalent u **Shannon entropy** H za neku slučajnu promenljivu X

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

Jos jedna od merila za kvantno stanje jeste merenje "validnost" (eng. fidelity) između dva kvantna stanja $|\phi\rangle$ i $|\psi\rangle$. Neka je F funkcija koja meri osobinu, ona 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.[12]

Odnost kvantne i klasicne teorije informacije

Kvantna i klasicna informacija se u dosta stvari razlikuju. Dok se klasicna informacija prolazi kroz sisteme sa dobro definisanim stanjima, moze se kopirati i pri procesom merenje se nemenja, Kvantna informacija je enkodovana u kvantnim sistemima, ne moze se kopirati i pri procesu merenje ona se menja. Takodje kvantna informacija ima neke osobine koje se ne mogu iskazati u klasicnoj informaciji, kao sto su superpozicija i kvantna uvezanost [8] Kvantna teorija informacije se bavi:

1. Prenosjenju klasicnih informacija preko kvantnih kanala
2. Prenosjenju kvantnih informacija preko kvantnih kanala
3. Efekat kvantne uvezanosti na prenosjenje informacija

1.3.1 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

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

Ako imamo parove $|v'_j|, \Phi_j$ cuvamo kao float brojeve u QRAM-u, onda mozemo da konstruisemo $\log_2 N$ kubit kvantno stanje $|v\rangle = |\vec{v}|^{-\frac{1}{2}} \vec{v}$ u $O(\log N)$ koraka

Kada smo kreirali kompresovane kvantne vektore od ulaznih vektora, mozemo da vrsimo transformacije koristeći kvantne algoritme, za dalje koriscenje podataka za masinsko ucenje. Ovaj proces zove se **postprocessing** i u opstem obliku njemu je potrebno $O(\text{poly}(\log N))$ koraka. [6]

2 Linearne algebra za kvantno masinsko ucenje

Da bi videli kako kvantni racunari poboljsavaju masinsko ucenje, treba da se vidi kako kvantni racunari obradjuju linearnu algebru, jednu od osnova modernog masinskog ucenja.

Tokom godina razvijeni su nekoliko kvantnih algoritama koji resavaju probleme linearne algebre. Zajedno ti algoritmi se nazivaju **osnovni kvantni**

podprogrami linearne algebre (eng. qBLAS), i oni se koriste u izradi algoritama za kvantno masinsko učenje.

Primeri algoritama koji su deo qBLAS-a su:

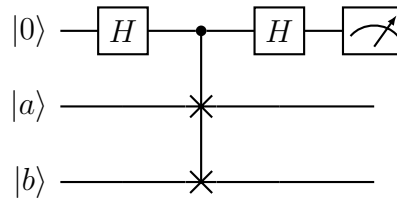
- HHL algoritam: koristi se za resavanje sistema linearnih jednačina, koristeći 2^n dimenzionalni vektorski prostor za resavanje sistema sa n promenljivih. [1]
- Kvantna Furijeova transformacije [9]
- Kvantna procena faza za eigen vrednosti i eigen vektora/stanja. [9]

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

3 Kvantna teorija kompleksnosti

3.1 SWAP Test

Ova rutina je jednostavan kvantni algoritam koji izrazava skalarni produkt za dva ulazna kvantna stanja $|a\rangle$ i $|b\rangle$. [4]



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

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

gde je $F(|a\rangle, |b\rangle) = |\langle a | b \rangle|^2$ - validnost izmedju dva kvantan stanja. Verovatnoca $P(|0\rangle) = 0.5$ znaci da su kvantan stanja $|a\rangle$ i $|b\rangle$ su medjusobno ortogonalna, a verovatnoca $P(|0\rangle) = 1$ znaci da su kvantan stanja identicna. Ova rutina treba da se ponavlja vise puta da bi se dobila dobra estimacija vrednosti validnosti.

SWAP test se može koristiti za izračunavanje Euklidske udaljenosti između kvantnih stanja u visedimenzionom prostoru kao i u velikom broju kvantnih algoritama.

4 Kvantno masinsko učenje

Kvantno masinsko učenje je spoj kvantnih računara i masinskog učenja. U programima Kvantnog masinskog učenja koriste se kvantni algoritmi (npr. qBLAS algoritmi) kao deo metoda optimizacija slične klasičnim metodama masinskog učenja.

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

1. Obrada klasičnih podataka na kvantnim masinama (**Masinsko učenje dopunjeno kvantnim računarima** eng. Quantum-enhanced machine learning)
2. Obrada kvantnim podataka na kvantnim masinama

Problem kod obrade klasičnih podataka na kvantnim masinama 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 (velicina kvantnog kola zna da poraste i na skalu oko 10^{25} za jednostavnu implementaciju HHL algoritma). [1]

4.1 Quantum support vector machine

Jedan od najjednostavnijih primera metoda Kvantnog masinskog učenja jeste **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. [1]

4.1.1 Klasicičan 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. Postoji kod većine slučajeva odvojenost između dve klase podataka nije linearizibilno odvojivo, sa SVM metodom koristi se i **Kernel**

metoda.

Pronalazenje optimalne hiper-ravni se sastoji od minimizacije $|w|^2/2$ u nejednacinu $y_j(w * x_j + b) \geq 1$ za svako j . Ovo minimizacija se moze uraditi, ako uvedemo Karush-Kuhn-Tucker mnozioca $\vec{\alpha} = (\alpha_1, \dots, \alpha_M)$ i maksimizujemo ih nad Lagranzovoj 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 sledecim ogranicenjima $\sum_{j=1}^M \alpha_j = 0$ i $\forall j \leq M \ y_j \alpha_j \geq 0$. Tako da, parametre za hiper-ravan se izvode kao: $w = \sum_{j=1}^M \alpha_j x_j$ i $b = y_j - w x_j$ (za one j gde vazi da $\alpha_j \neq 0$). Mali broj α_j je razlicitno od nule, takve promenljive se odnose na vectore x_j koji leze na ravni, ti vektori se zovu **Support vektori** [11]

Kernel metoda transformise podatke u prostor gde su dve klase linearno odvojive. Metoda se oslanja na to da se linearna funkcija moze zapisati iskljucivo kao dot produkt izmedju 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 zamenim x sa izlazom funkcije $\phi(x)$, a dot produkt sa funkcijom $k(x, x_i) = \phi(x) * \phi(x_i)$. Funkcija k se zove **kernel**, dok funkcija ψ je funkcija koja preslikava podatke iz jednog prostora u drugi. Operator $\langle * \rangle$ predstavlja untrasnji produkt ekvivalentno $\phi(x)^T \phi(x_i)$. [5]

Kada zamenimo dot produkt sa kernelom, funkciju predikcije mozemo da zapisemo kao

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

Jedan od velikih mana kernel metode jeste cena evaluacije izlaza kernel funkcije je linarna u odnosu na broj trening primera, jer i -ti bi oznacavao clana $\alpha_i k(x, x_i)$ kernel funkcije. [5]

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

Takodje krajnje resenje se je binarni klasifikator za neki vektor x :

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

4.1.2 Kvantni algoritam

Pretpostavimo da imamo metodu za treniranje (eng. Oracle) koja vraća norme $|x_j|$, labela y_j i kvanten 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 evaluaciju inner produkt priprema se kernel matrica, može se dobiti SVD algoritam kompleksnošću $O(\log(1/\epsilon)M^3 + M^2 \log(N/\epsilon))$. Kernel matrica je od velike važnosti za reformulaciju algoritma kao funkciju kvadratnog troška. Uvodimo simplifikaciju za nejednakosti, tako što uvodimo promenljivu e_j i koristimo osobinu labela da $y_j^2 = 1$

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

Porod ove jednačine imamo i implicitan uslov Lagranžove funkcije da sadrži taksanu (eng. penalty) promenljivu $\gamma/2 \sum_{j=1}^M e_j^2$ gde je definisana γ za relativne težine greske treniranja. Ako uzmemo parcijalno derivat od Lagranžove funkcije i eliminisemo promenljivu u i e_j dovodi do aproksimacije funkcije kvadratnog troška problema:

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

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 offset-a b . Promenljiva α_j ima ulogu određivanja distance od optimalnog rešenja. Tako da na rešenje, odnosno pronalazenje 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 verovatnoca uspeha pri swap testu između $|b, \vec{\alpha}\rangle$ i $|x\rangle$. Za efikasnost merenja algoritma, posebno izračunavanja interzne matrice, matrica \hat{F} mora da se razdvoji na jednostavne elemente. Tako da

matrica \hat{F} može da se razdvoji na sledeće elemente $\hat{F} = (J + K + \gamma^{-1}\mathbb{1})/trF$. Gde je matrica

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

Takodje, za estimaciju faze pravimo formulaciju Lijeovog produkta $e^{-i\hat{F}\Delta t} = e^{-i\gamma^{-1}\mathbb{1}\Delta t/trF} e^{-iJ\Delta t/trF} e^{-iK\Delta t/trF} + O(\Delta t)$

Za njega vazi da ima dve eigen vrednosti oblika $\lambda_{\pm} = \pm\sqrt{M}$, a, istovetno, eigen 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 $\gamma^{-1}\mathbb{1}$ dve eigne vrednosti su $v_1 = 0$ i $v_2 = \gamma^{-1}M$. Sada mozemo da aproksimiramo fazu za $e^{-i\hat{F}\Delta t}$.

Prvi korak, Stanje $|y\rangle$ može da se transformise u eigen state $|u_j\rangle$ matrice \hat{F} , koja ima eigen vrednost λ_j . Ono je obilka $|y\rangle = \sum_{j=1}^{M+1} \langle u_j|y\rangle |u_j\rangle$. Ako inicijalizujemo aproksimaciju eigen vrednosti na $|0\rangle$, i primenimo estimaciju faze nad stanjem dobicemo stanje blize pravoj eigen vrednosti:

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

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

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

Klasifikacije Sada imamo trenirani model kvantnog SVM-a i zelimo da klasifikujemo stanje $|x\rangle$. Od stanja $|b, \vec{\alpha}\rangle$, koriscenjem metode za treniranje, konstruisemo 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 konstruisemo 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 $N_x = M|x|^2 + 1$. Konstruisemo 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, verovatnoca dobivanja pozitivne vrednosti je $P = |\langle\psi|\phi\rangle|^2 = \frac{1}{2}(1 - \langle\tilde{u}|\tilde{x}\rangle)$. Ovde unutrasnji produkt, 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 obicno izracunava u $O(1)$ na kvatnom racunaru. Ako hocemo preciznost ϵ , treba da iteriramo kroz algoritam merenja $O(P(1 - P)/\epsilon^2)$ puta. [11]

4.2 Quantum principal component analysis

Ova je metoda koja se koristni za smanjivanje dimenzija vektora podataka gde nam je bitno da sacuvamo sto vise informacije o podatku - labava kompresija (eng. lossy compression).

4.2.1 Klasicni algoritam

Neka za svaku tacku $x^{(i)} \in \mathbb{C}^n$ zelimo da transformisemo u tacku $c^{(i)} \in \mathbb{C}^l$ gde je $l < n$. Zelimo da nadjemo funkciju enkodovanja koja za ulaz x vraca c , odnosno, $f(x) = c$. Takodje zelimo da nadjemo funkciju dekodovanja $g(f(x)) \approx x$.

Zbog jednostavnosti, uzecemo funkciju mnozenja matrica kao funkciju dekodavanja. Neka je $g(c) = Dc$, gde je $D \in \mathbb{C}^{n \times l}$ matrica definisana za dekodovanje. Takodje zbog optimalno izracunavanja funkcije enkodovanja, PCA uvodi ogranicenje da su kolone medjusobno ortogonalne. Jos jedno ogranicenje koje moze da se uvede u algoritam, i koji ce dovesti do jedinstvenog resenja, jeste da su sve kolone matrice D u unitarnoj normi. Jedan od nacina na koji hocemo da nadjemo optimalnu projekciju c za ulaz x jeste da nadjemo najmanju L2 distancu izmedju ulaza x i dekodovane vrednosti $g(c)$

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

I ova za pronalazenje minimalne distance ce dovesti do optimalnog resenje $c = D^T x [5]$. Tako da funkcija enkodovanja je oblika:

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

Takodje, mozemo 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 pronalazjenje optimalnog c za ulaz x , odnosno kao pronalazjenje minimalne L2 distance za ulazne vektore njihove rekonstrukcije.

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

Posle procesa izvodjenja [5], jendacina za optimalnu matricu D je oblika:

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

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

4.2.2 Kvantni algoritam

U kvantnom algoritmu bitno nam je da nađemo eigen vektore i eigen vrednosti za ulaz. Ovo se dosta olakša na drugi deo metode koji je opisan u Support vector machine sekciji. Ako izaberemo random vektor v_j iz skupa ulaznih vektora, kreiramo kvantno stanje $|v_j\rangle$; tada možemo da kreiramo *density* matricu $\rho = (1/N) \sum_j |v\rangle \langle v|$ gde je N veličina skupa vektora. [1] Slično **qSVM** nad *density* matricom ρ možemo da apliciramo algoritam esitacije 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 $|\chi_i\rangle$ eigen vektor od matrice ρ , \tilde{r}_i je esimacija eigen vrednosti, a $\psi_i = \langle \chi_i | v_j \rangle$. I primenom SWAP testa na 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 dobojamo eigen vrednost i eigen vektor za *density* matricu ρ . Ako uradimo ovaj proces nad većem brojem kopija matrice ρ , dobićemo preciznije estimacije eigen vrednosti i eigen vektora.

Sada kada imamo eigen vrednost i eigen vektor možemo da rekonstruisemo matricu za enkodovanje D . Vremenska složenost ovog algoritma je $O(\log d)$. [7]

4.3 Kvantna neuralna mreza

Neuralne mreze su osnova polja koji se naziva **Duboko učenje** i zato postoji veliku pažnja za razvoj istog. U papiru [3], autori su predstavili osnove algoritama za Kvantnu neuralnu mrežu (QNN). Da li su neke primere, neke prednosti i neke nedostatke kvantnog pristupa neuralnim mrežama

Neka unani 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 binarnu oznake $l(z)$ koje može da bude $+1$ ili -1 . Zbog jednostavnosti neka se u našem setu nalazi 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 podskupu ulaznih kubita i zavisi od promenljive θ . Sada izabracemo podskup od L transformacija:

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

koja zavise od L parametara $\vec{\theta} = \theta_L \theta_{L-1} \dots \theta_1$. Za svaki string z kreiramo 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 da "naucimo" proces da vraća tačne vrednosti za dati ulazni string. Posto merenje izlaznog kubita nije sigurno, odnosno merenje kubita dobijamo tačnu vrednost sa nekom verovatnoćom uvodimo transformaciju:

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

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

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

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

Mozemo primetiti da ova funkcija troska je linearna i da je minimum u 0, jer je vracema vrednost izmedju -1 i $+1$. Ako pretpostavimo da kvantna neuralna mreza radi savrseno, tako da za svaki ulazni string z , merenje uvek vraca tacnu oznaku. To onda znaci da optimalna promenljiva $\vec{\theta}$ postoji i da je minimum za funkciju troska u 0 za sve ulaze z .

Neka imamo skup stringova S za treniranje, sa njihovim oznakama. Postoji kvantni proces koji ima mogucnost da prikaze trazene labele i zavisi od parametara $\vec{\theta}$. Opisacemo proces kako da dodjemo do optimalnih parametara $\vec{\theta}$. Neka pocnemo sa random promenljivom $\vec{\theta}$ (ili ako imamo neku pretpostavku vrednosti parametara). Izaberimo neki string z^1 iz skupa za traniranje. Primenujemo kvanti proces nad izabranim stringom:

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

i merimo Y_{n+1} na zadnjem kubit. Nakon nekoliko merenja mozemo da imamo dobru aproksimaciju ocekivane vrednosti od Y_{n+1} i tada izracunavamo $loss(\vec{\theta}, z^1)$. Nakon toga, zelimo da promenimo parametar $\vec{\theta}$ tako da smanjimo funkciju troska za string z^1 . Postoje dva nacina da se uradi trazeno: (1) da uradimo pomeraj po nekom uzimanju uzorka u $[\vec{\theta} - \epsilon, \vec{\theta} + \epsilon]$ intervalu. (2) da izracunamo derivat funkcije troska po $\vec{\theta}$ i da se malo pomerimo ka pravcu koji smanjuje funkciju. Ovo nam daje novi parametar $\vec{\theta}^1$. Sada biramo ponovo iz skupa neki string z^2 i ponovimo prethodni proces ali sa parametrom $\vec{\theta}^1$. Ovako dobijamo novi parametar $\vec{\theta}^2$ koji ima manju funkciju troska za string z^2 nego parametar $\vec{\theta}^1$. Ovako prolazimo kroz proces sve dok ne prodjemo kroz ceo skup S . Kao rezultat ovoga generisali smo sekvencu parametara $\vec{\theta}^1, \vec{\theta}^2, \dots, \vec{\theta}^S$. Ako nam je "ucenje" parametara uspesno onda bi smo dobila da operator $U(\vec{\theta}^S)$, kada se primeni na stanju $|z, 1\rangle$, vratice stanje koje kada se izmeri na izlazu vraca tacnu oznaku $l(z)$. Ako je z iz skupa za traniranje, reci cemo da je model fitovao podatke za treniranje. Ako je z izvan skupa za treniranje, mozemo raci da je model naucio da generalizuje i za nevidjenje podatke.

Ovaj proces koji je opisan, primeti ce te, u klasicnom masinskom ucenju zove se "Stohasticko uvenje". U tradicionalnom masinskom ucenju sa neuronskim mreza, parametri se prikazuju kao promenljive unutar matrice, koja je linarna u odnosu na unutrasnje vektore. Nad Komponentama tih vektora vrsi se nelinearne transformacije, pre nego sto se mnoze sa ostalim parametrima. Uvedjenje dobre ne linearnosti je jedan od glavnih delova

uspesne implementacije modela u klasicnom masinskom učenju. Ovu osobinu klasnih neuralnim mreza tesko je prebaciti u kvantni sistem, jer je kvantna mehanika, osnova celog koncepta kvantnog racunarstva, samo po sebi linearna. U metodi koja je opisana, svaka unitarna opearcija se izvsava nad izlazom prethodne operacije, pri cemu se izmedju operacija ne izvsava nikakva nelinearna transformacije. Neka name je svaka unitarne transformacija oblika $e^{i\theta\Sigma}$, gde je Σ produkt tenzora koji se sastoji iz skupa Paulijevih operatora, i rade nad nekolicinom kubita. Derivat operatora po $\vec{\theta}$ je ograničen po L , to jest po broju parametara. Ovo je znacajno, jer znaci da gradijent ne moze da ode u beskonacno i tako izbegavamo veliko problem koji se moze desiti klasicnim neuralnim mreza.

4.3.1 Reprezentacija modela

Neka imamo 2^n , n -bitnih stringova i vezano za njih postoje $2^{(2^n)}$ funkcija oznaka $l(z)$. Ako nam je data odredjena funkcija oznaka onda mozemo da definisemo operator nad komputacionim osnovama 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 komputaciona osnovna stanja. Takodje, posto funkcija oznaka $l(z)$ moze da vrati ili $+1$ ili -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 abstraktnom nivou imamo mogucnost da predstavimo bilo koju funkciju oznake kao kvantno kolo.

Objasnjenje kako da se napise operator U_l kao produkt dve kubit unitarne transformacije. Zbog ovoga treba da se predje 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 mozemo da iskoristimo **Reed-Muller** iskazivanje 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 koeficijenati $a \in 0, 1$. Primecuje se da imamo 2^n koeficijenata i posto su oni ili 0 ili 1 da stvarno imao $2^{(2^n)}$ mogucih *boolean* funkcija. Nasa funkcija

b može biti ekponencijalno dugačka. Sada možemo da zapisemo unitarnu transformaciju koja zavisi od funkcije oznaka kao:

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

gde je B operator, dijagonalan u odnosu na komputacione baze, koji odgovara nama data funkcija b . Svaka vrednost u B se množi sa X_{n+1} tako da svaka vrednost je komutativna sa ostalim vrednostima. Svaka član, razlicit od nule, u **Reed-Muller** formuli utice u U_l na kontrolni *bit flip* na izlaznom kubit.

Ovaj rezultat kvantno reprezentacije ima analog u klasicnoj teoriji reprezentacije [2]. Ona pokazuje da bilo koja *boolean* funkcija oznake može da se prestavi u neuralnoj mrezi dubine tri, gde srednji slog ima velicinu 2^n . Ovako velika matrica ne bi mogla da se prestavi na klasicnim racunarima, ali na kvantnim racunarima, oni po priori rade nad Hilbertovim prostorim sa ekponencijalnim dimenzijama. Ali jos nije dokazano da svaka *boolean* funkcija može da se prestavi u kvantno kolo koje nije ekponencionalne dubine. Na tome se trenutno dosta radi u naucnim krugovima.

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

$$P_{\mathbb{S}}(z) = \sum_j \oplus a_j b_j$$

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

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

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

4.3.2 Učenje modela

U ovoj podsekciji ce se objasniti dve potencijalne metode kako da menja parametar $\vec{\theta}$ tako da se funkcija troska smanje. Ako su nam dati parametri $\vec{\theta}$ i trening primer z , prvo procenjujemo vrednost trosak od $loss(\vec{\theta}, z)$.

Da bi smo ovo uradili treba da napravimo vise merenje Y_{n+1} za $U(\vec{\theta})|z, 1\rangle$. Da bi smo ovo uspeali sa verovatnocom vecom od 99%, procena of funkcije troska koja je δ intervaluj od prave vrednosti funkcije troska treba da napravimo najmanje $2/\delta^2$ merenja ($\delta \in (0, 1)$).

Nako sto procenimo vrednost funkcije troske zelimo da izracunamo gradijent od funkcije troska u odnosu na $\vec{\theta}$. Jedan od nacina jeste da menjamo jedenu po jednu promenljivu u $\vec{\theta}$. Nakon svake promene treba da se izracuna $loss(\vec{\theta}', z)$, gde $\vec{\theta}'$ je razlicit od $\vec{\theta}$ za neku malu vrednost u jednoj promenljivi. Ako bi se koristio simetrican derivat funkcije troska svaku promenljivu parametra bi mogli da izrazunamo do preciznosti η u oko $1/\eta^3$ merenja. Ovaj proces bi trebao da se ponavlja L puta da bi se dobio puni gradijent.

Alternativna strategija jeste da se menja svaka promenljiva gradijenta, sto se koristi kada su sve unitarne transformacije oblika $e^{i\theta\Sigma}$. Ako posmatramo derivat za funkciju troska $loss(\vec{\theta}, z)$ za parametar θ_k , koji je vezan za transformaciju $U_k(\theta_k)$ (koja ima i generalni Pauli-er 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 definisemo 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 derivat mozemo da zapisemo kao:

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

$\mathcal{U}(\vec{\theta})$ se mozemo posmatrati kao kvanto kolo koji sadrzi $2L + 2$ unitarnih transformacija. Sada mozemo da primenimo $\mathcal{U}(\vec{\theta})$ nad stanjem $|z, 1\rangle$. Ako koristimo dodati kubit, mozemo da merimo imaginarni deo derivata funkcije. Zapocecemo sa stanjem $|z, 1\rangle \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ i primenicemo $i\mathcal{U}(\vec{\theta})$ na 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 dodati kubit, verovatnoća da se dobije 0 je:

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

tako da, ponavljanjem ovog merenja mozemo da dobijemo dobru procenu imaginarnog dela stanja iz kojeg mozemo da izvucemo procenu k-te komponente trazeneog gradijenta. Ovaj metod izbegava numbricne nepreciznosti prethodne strategije. Cena ove metode je potreba da dodatim kubitom kao i kvatno kolo $2L + 2$ dubine.

Posto samo izracunali gradijent, sada treba metod za izmenu $\vec{\theta}$. Neka je \vec{g} gradijent funkcije troska po parametru $\vec{\theta}$. Sada menjamo $\vec{\theta}$ u pravcu \vec{g} . Sa velicinom "koraka" γ imamo

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

Posto zelimo da smanjimo trosak na 0 mozemo da napravimo da:

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

Ovako nesto bi dovelo da trosak bude 0 za trenutni primer, ali moze da dovede do losih efekata za ostale primera. Ovde se u klasicnom masinskom ucenju obicno uvodi promenljiva, stepem ucenja $r \in (0, 1]$ i onda imamo sledece:

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

Deo uspesne implementacije masinskog ucenja je racionalno odaberemo vrednost stepena ucenja.

Ucenje parnosti podskupa Za dati podskup S , unitarna transformacija U_{P_S} moze da prikaze parnost podskupa za sve ulazne stringove. Na bi se "naucio" skup unitarnih operacija koji zavise od parametara, sa tim da za svaki podskup postoje parametri koji opisuju U_{P_S} . Najlaksi nacin 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 savrsena 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 uspeali 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 su 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 $l(z)$ može se ubaciti u funkciju troska, ali sada može da se izračuna prosek troska za sve 2^n stringove, jer imamo eksplicitnu formulu za oznake i njihovo očekivanje. Postoje više verzija izračunate funkcije, koja zavise od izlaza $n \bmod 4$ i koliko bitova se nalazi u podskupu S . Za prikaz uzeli su primer gde je n deljiv sa 4 i skup S sadrži svih n bitova. U tom slučaju prosečan trosak 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 minimum 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 trosku, nijedan metod koji se oslanja na gradijentni pristup bi mogao biti koriscen za pronalazenje optimalnog ugla za bilo koji primer sa povećim n , gradijent bi radio izvan preciznosti masine 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 citanje rezultata, koji je postavljen na 1. Pa 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 merenje bude ekvivalentan nekim dvema oznaka koje označavaju neke osobine kvantnog stanja. To je prikazano u sledecem primeru. Posmatrajmo Hamiltojev

operator H (eng. Hamiltonian), koji je suma lokalnih vredosti sa dodatnom osobino da ima i pozitivne i negativne eigen vrednosti. Sa datim kvantnim stanjem $|\psi\rangle$, obelezava se oznakom koja pokazuje da li je ocekivana vrednost Hamiltonijevog 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 ze dovoljno malo β ovo je priblizno jednako $2\beta \langle\psi| H |\psi\rangle$ i tako imamo znak ocekivane vrednosti za predikciju oznake koje je jednata sa tacnom oznakom. Ovako prikazana, ovo je funkcija oznake sa kvantnim kolima koja ima malu gresku. Mala greska dolazi iz toga sto $\langle\psi| \sin(2\beta H) |\psi\rangle$ samo priblizno jednako $2\beta \langle\psi| H |\psi\rangle$ Ako uzmemo da β bude dosta manje od $1/\|H\|$ (inverz of norme matrice H), mozemo da napravimo da nam greska bude mala.

Posmatrajmo graf gde na svakoj ivici imamo ZZ uparivanje sa koeficijentom ili $+1$ ili -1 . Hamiltonijev operator je oblika: $H = \sum_{i,j} J_{ij} Z_i Z_j$ gde prvobitna suma ogranicena 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 $U_H(\beta)$ koja osigurava da mozemo da oznacimo trazenu oznaku ako izaberemo malo β

Kvantno stanje $|\psi\rangle$ je u Hilbertovom prostoru sa 2^n dimenzija i ne mozemo da ocekujemo da naucimo oznake za svako stanje. Posmatrani Hamiltonijev operator ima bitovnu strukturum, tako da mozemo da se ogranicimo kvantna stanja sa bitnovnim strukturama. Tako da je predlozeno da se za treniranje koriste stanja samo sa ovom formom i za testiranje isto.

References

- [1] Pancotti N. et al. Biamonte J. Wittek P. “Quantum machine learning”. In: *Nature* 549 (2017).
- [2] George V. Cybenko. “Approximation by superpositions of a sigmoidal function”. In: *Mathematics of Control, Signals and Systems* 2 (1989), pp. 303–314.
- [3] 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>.
- [4] 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.
- [5] 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>.
- [6] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. “Quantum algorithms for supervised and unsupervised machine learning”. In: *arXiv preprint arXiv:1307.0411* (2013).
- [7] 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>.
- [8] Dan C. Marinescu. *Classical and quantum information*. Academic Press, 2012. Chap. 8. ISBN: 9780123838742; 0123838746.
- [9] Dan C. Marinescu. *Classical and quantum information*. Academic Press, 2012. Chap. 5. ISBN: 9780123838742; 0123838746.
- [10] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.
- [11] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. “Quantum support vector machine for big data classification”. In: *Physical review letters* 113.13 (2014), p. 130503.
- [12] Vlatko Vedral. *Introduction to Quantum Information Science*. Oxford University Press, 2006.