Kvantno mašinsko učenje

Milan Bojic

Jun 2022

Sadrzaj

1 Uvod

Prethodna decenija je bila obeležena mašinskim učenjem, sa njenom primenom u svakodnevnom životu običnih ljudi. U prethodnih nekoliko godine 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 rešeavanju ovog problema očekuje se da pomogne razvoj kvantnog računarstva, oblast koja koja se razvoja 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 nadmoć svakog dana.

Od kvantno mašinsko učenje se očekuje da bude prekretnica u mašinskom učenju kakvog 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 koji se nisu mogli primetiti u klasičnom sistemu.

U ovom radu vas uvesti u osnove kvantnog računarstva i predstaviću nekoliko metoda kvantnog mašinskog učenja koji su bila istraživana u prethodnom periodu.

2 Kvantno računarstvo

Pre nego što se počne pričati o Kvantnom mašinskom učenju, treba objasniti neki osnovni pojmovi da bi lakše razumeli ostatak rada.

2.1 Osnovni pojmovi

Potrebni pojmovi su:

- Kubit (eng. Qubit)
- Kvantna kapija (eng. Quantum Gates)
- Kvantna uvezanost (eng. Quantum entanglement)
- Kvantan 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 (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 aplitude zadatih stanja i za njih važi:

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

Pošto 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 tomo u kom stanju se nalazi kubit se dobija merenjem kubita, tade bi da 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 polarizovani fotoni,gde će dva stanja da se uzimaju kao vertikalna i horizontalna polarizacija.

Kvantna kapija

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

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

bit flip kolo (zamenjuje aplitude na kubitu), 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 kvanta 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, dobili bi smo 0 ili 1, međutim drugi kubit bi takođe upao u određeno stanje i to u istom trenutni kad smo izmerili prvi daleki kubit. Ovo je zapravno gde se nalazi glavan 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. Kvantan memorija su uređaji koji čuvaju kvantna stanja fotona, bez da uništavaju kvanten informacije koja se nalazi u fotonu. Ovakva memorija zahteva koherentni sistem materijala, jer bi u suprotnom kvantna informacija unitar 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 uvezanost, tako da se izvrši neka kalkulacija. Uređaji koji izvršavaju kvante kalkulacije zovu se **kvantni računari**. Kvantni računari se sastoje od kvantnih kola i elemntarnih kvantnih kapija koje sluze za prenosenje i manipulisanje kvantnih inforamcija. [12]
Jedna od glavnih primena Kvantnih računari jeste simulacija fizičkih sistema, bili oni kvantne ili klasinče prirode.

2.3 Kvantna inforamcija

Kvantan informacija je informacija o stanju kvantnog sistema. O njihovim svojstvima bavi se **kvantna teorija informacije**. Takođe, kvantane informacije se mogu izmeriti na isti način kao i klasicna informacija koristeći se *Šenononvoj metodom*. Odnosno, postoji jednistveno merilo, to jest funkcija nad kvantnim stanjem, koje je funkcija verovatnoće, kontinuiteta i sumiranja.[15] Ova funkcija se zove **von Neumann entropija** i za neki ulazni kubit ϱ postoji ekvivalent u **Shannon entropiji** H za neku slucajnu promenljivu X

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

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.[15]

Odnost kvantne i klasicne teorije informacije

Kvantna i klasična informacija se u dosta stvari razlikuju. Dok klasična informacija prolazi kroz sisteme sa dobro definisanim stanjima, može se kopirati i pri procesu merenja se ne menja, Kvantna informacija je enkodovana u kvantnim sistemima, ne može se kopirati i pri procesu merenje ona se menja. Takođe kvantan informacija ima neke osobine koje se ne mogu iskazati u klasicnoj informaciji, kao sto su superpozicija i kvantan uvezanost [10]
Kvantana teorija informacije se bavi:

1. Prenošenje klasicnih informacija preko kvantnih kanala

2. Prenošenje kvantinih informacija preko kvantinh kanala

3. Efekat kvantne uvezanosti na prenošenje informacija

4. Informacioni aspekt kvantnog merenja, odnos 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 $\overrightarrow{v}$ 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 float brojeve u QRAM-u, onda možemo da konstruišemo $\log_2 N$ kubit kvantno stanje $|v\rangle = |\overrightarrow{v}|^{-\frac{1}{2}}\overrightarrow{v}$ 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 **postprocessing** i u opštem obliku njemu je potrebno $O(poly(\log N))$ koraka. [8]

3 Linearne algebra za kvantno mašinsko učenje

Da bi videli kako kvantni računari poboljšavaju mašinsko učenje, treba da se vidi kako kvantni računari obrađuju linearnu algebru, jednu od osnova modernog mašinskog učenja.
Tokom godina razvijeni su nekoliko kvantnih algoritama koji rešavaju probleme linearne algebre. Zajedno ti algoritmi se nazivaju **osnovni kvantni podprogrami linearne algebre** (eng. 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 dimenzijonalni vektorski prostor za rešavanje sistema sa n promenljivih. [3]

- Kvantna Furijeova transformacije [11]

- Kvantan procena faza za eigen vrednosti i eigen vektora/stanja. [11]

Ovi algoritmi su korišćeni kao osnova napredinih 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 tesko kreirani u realnom svetu (npr. QRAM).

4 Kvantna teorija kompleksnosti

U klasičnoj teoriji kompleksnoti klasifikuju se algoritamski problemi po njihovoj težini rešavanja. Problemi se klasifikuju u **klase kompleksnosti**, oni se mogu posmatrati kao kolekcija algoritamskikh problema koji dele neke zajedničke osobine vezane za komputaciona sredstva potrebna da bi se oni rešili (uglavnom vreme i prostor). [12]

4.1 Primeri klasa kompleksnosti

Među kojima su najpoznatiji i najvažniji **P** i **NP**. Zbog prirode kvantnih računara objasniću neke druge klase kompleksnosti.

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

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

4.2 BQP

Klasa komepleksnoti **BQP** (Bounded-Error Quantum Polynomial-Time) je klasa problema koji se mogu efikasno rešiti na kvantnom računaru, ako se dopusti ograničena verovatnoća greške [12]. Formalija definicija bi bila:[1]

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

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

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

Odnos sa klasnicnim klasam kompleksnosti

Prva stvar koja važi jeste da **BPP** $\subseteq$ **BQP**, odnosno da sve što može da se uradi sa klasicnom probalističkim računarom može da se uradi i na kvantnom računaru.

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

4.3 Primeri kvantnih algoritama

4.3.1 Šorov algoritam

Pošto je trenetno najoptimalniji algoritam na klasičnom računaru (*general number filed sieve*) se izvršava u sub-ekponencijalnom vremenu. Šorov algoritam rešava problem nalaženja periode, koji može da se iskoristi za problem faktorizacije. [14]
Problem: ako imamo periodicnu funkciju

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

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

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

Šorov algoritam koristi estimaciju kvantne faze na unitarnom operatoru:

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

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

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

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

Ovde eigen stanje ima eigen vredost 1, što nam ništa ne znači. Zanimljivije je kada se gledaju eigen stanja gde su faze drugačije za svako bazno stanje. Posebno se posmatraju slucajevi gde faza k-tog stanja je proporciona k:

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

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

U ovom slučaju eigen vrednost sadrzi r. Ovde r kao faktor normalizacije između r baznih stanja. Sada, možemo ova stanja da pomnožimo sa $s \in N_0$ koji ce uticati na krajnju eigne vrednost.

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

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

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

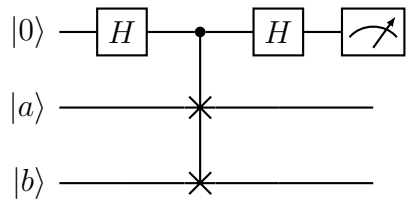
$$\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle = |1\rangle$$

Pošto je bazno stanje $|1\rangle$ superpozicija za data eigen stanja, to znači da možemo da uradimo *kvantnu esitmaciju faze* nad U koristeći se stanjem $|1\rangle$, u tom slučaju ćemo da izmerimo fazu Φ

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

4.3.2 SWAP Test

Ova rutina je jednostavan kvantni algoritam koji izražava skalarni produkt za dva ulazna kvantna stanja $|a\rangle$ i $|b\rangle$. [6]



Verovatnoća 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 između dva kvantan stanja. Verovatnoća $P(|0\rangle) = 0.5$ znači da su kvantan stanja $|a\rangle$ i $|b\rangle$ su međusobno ortogonalna, a verovatnoća $P(|0\rangle) = 1$ znači da su kvantan stanja identična. Ova rutina treba da se ponavalja vise puta da bi se dobila dobra estimacija vrednosti validnosti.
SWAP test se može koristiti za izračunavanje Euklidske distance između kvantnih stanja u višedimenzionom prostoru kao i u velikom broju kvatninh algoritama.

5 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 optimizacija slične klasičnim metodama mašinskog učenja.

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

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

2. Obrada kvantnim podataka na kvantnim mašinama

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

5.1 Quantum support vector machine

Jedan od najjednostavnijih primera metoda Kvantnog mašinskog 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 ce se naći na jednoj polovini hiper-ravni. [3]

5.1.1 Klasican algoritam

Ova metoda određuje klase koristeći linarnu funkciju $w^T x+b$. SVD predviđa prvu klasu ako je izlaz funkcije je pozitivan, a predviđa drugu klasu je izlaz negativan. Pošto kod većine slučajeva odvojenost između dve klase podataka nije linearzibilno odvojivo, sa SVM metodom koristi se i **Kernel metoda**. Pronalaženje optimalne hiper-ravni se sastoji od minimizacije $|w|^2/2$ u nejednačini $y_j(w * x_j + b) \geq 1$ za svako j. Ovo minimizicija se može uraditi, ako uvedemo Karuš-Kun-Taker množioce (eng. Karush-Kuhn-Tucker multiplier) $\overrightarrow{\alpha} = (\alpha_1, ..., \alpha_M)$ i maksimizujemo ih nad Lagranžovoj funkcijom:

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

Sa sledećim ograničenjima $\sum_{j=1}^{M} \alpha_j = 0$ i $\forall j \leq M \; y_j \alpha_j \geq 0$. Tako da, 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 važi da $\alpha_j \neq 0$). Mali broj α_j je različito od nule, takve promenljive se odnose na vectore x_j koji leže na ravni, ti vektori se zovu **Support vektori** [13]

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 produkt 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 dozvoljva da zamenim x sa izlazom funkcije $\phi(x)$, a skalarni 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 unutrašnji produkt ekvivalentno $\phi(x)^T \phi(x_i)$. [7]

Kada zamenimo skalarni produkt sa kernelom, funkciju predikcije možemo 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 člana $\alpha_i k(x, x_i)$ kernel funkcije. [7]
Složenost SVM je $O(log(1/\epsilon)M^2(N + M))$, gde je ϵ preciznost rešenja, N je broj dimenzija prostora nad kojem radimo ,a M je broj trening primera.
Takođe krajnje rešenje je binarni klasifikator za neki vektor x:

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

5.1.2 Kvantni algoritam

Pretpostavimo da imamo metodu za treniranje(eng. Oracle) koja vraća norme $|x_j|$, labele 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 kompleksost jezgra algoritma mogla iskazati. Koristeći evaluaciju unutrašnjeg produkta priprema se kernel matrica, može se dobiti SVM 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 sto uvodimo promenljivu e_j i koristimo osobinu labela da $y_j^2 = 1$

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

Pored ove jednačine imamo i implicitan uslov Lagranžove funkcije da sadrži taksanu (eng. penalty) promenljvi $\gamma/2 \sum_{j=1}^{M} e_j^2$ gde je definisana γ za relativne težinu greške treniranja. Ako uzmemo parcijalni izvod od Lagranžove funkcije i eliminišemo promenljivu u i e_j dovodi do aprokcismaciju funkcije

kvadratnog troška problema:

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

Ovde $K_{ij} = x_i^T \cdot x_j$ je simetrična kernel matrica, $y = (y_1, ..., y_m)$ kao i $\overrightarrow{1} = (1, ..., 1)$. 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 rešenje, odnosno pronalaženje promenljivih za SVM je oblika:

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

U klasicnom algoritmu kopleksnost SVM sa funkcijom kvadratnog troška je $O(M^3)$

U kvantnom algoritmu, zadatak je generisanje stanja $|b, \overrightarrow{\alpha}\rangle$ koja opisuju hiper-ravan i onda klasifikuju stanja $|x\rangle$. U algoritmu, rešavamo normalizovanu jednačinu $\hat{F}\,|b, \overrightarrow{\alpha}\rangle = |y\rangle$, gde je $\hat{F} = F/trF$ sa ogranicenjem $\|F\| \leq 1$. Klasa će biti određenja kao verovatnoća uspeha pri swap testu između $|b, \overrightarrow{\alpha}\rangle$ i $|x\rangle$. Za efikasnost merenje algoritma, posebno izračunavanja inverzne 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

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

Takođe, za estimaciju faze pravimo formulaciju Lijevog 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 važi 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 možemo da aproksimiramo fazu za $e^{-i\hat{F}\Delta t}$.

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

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

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

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

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

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

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

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

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 dobivanja pozitivne vrendosti je $P = |\,\langle\psi|\phi\rangle\,|^2 = \frac{1}{2}(1 - \langle\tilde{u}|\tilde{x}\rangle)$. Ovde unutrašnji 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 obično izračunava u $O(1)$ na kvatnom računaru. Ako hoćemo preciznost ϵ, treba da iteriramo kroz algoritam merenja $O(P(1-P)/\epsilon^2)$ puta. [13]

5.2 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 kompešija (eng. lossy compression).

5.2.1 Klasicni algoritam

Neka za svaku tacku $x^{(i)} \in \mathbb{C}^n$ želimo da transformisemo u tacku $c^{(i)} \in \mathbb{C}^l$ gde je $l < n$. Zelimo da nađemo funkciju enkodovanja koja za ulaz x vraća c, odnosno, $f(x) = c$. Takođe želimo da nađemo 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. Takođe zbog optimalno izračunavanja funkcije enkodovanja, PCA uvodi ogranicenje da su kolone međusobno ortogonalne. Jos jedno ogranicenje koje može da se uvede u algoritam, i koji ce dovesti do jedinstvenog rešenja, jeste da su sve kolone matrice D u unitarnoj normi. Jedan od nacina na koji hoćemo da nađemo optimalnu projekciju c za ulaz x jeste da nađemo najmanju L2 distancu između ulaza x i dekodovane vrednosti $g(c)$

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

I ova za pronalazenje minimalne distance ce dovesti do optimalnog rešenje $c = D^T x$[7]. 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 ce se rešiti na isti nacin kao i pronalazenje optimalnog c za ulaz x, odnostno kao pronalazenje 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 [7], jendacina za optimalnu matricu D je oblika:

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

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

5.2.2 Kvantni algoritam

U kvantnom algoritmu bitno nam je da neđenmo eigen vektore i eigen vrednosti za ulaz. Ovo se dosta olanja na drugi deo metode koji je opisan u

Support vector machine sekciju. Ako izaberemo random vektor v_j iz skupa ulaznih vektora,kreiramo kvanto stanje $|v_j\rangle$; tada možemo da kreiramo *density* matricu $\rho = (1/N)\sum_j |v\rangle \langle v|$ gde je N velinica skupa vektora. [3] Slicno **qSVM** nad *density* matricom ρ možemo da apliciramo algoritam esitacije faze stanja. Odnosnto, da primenimo $e^{-i\rho t}$, t puta nad inicijalnim stanjem:

$$|v_j\rangle |0\rangle \to \sum_i \psi_i |\chi_i\rangle |\widetilde{r_i}\rangle$$

Gde je $|\chi_i\rangle$ eigen vektor od matrice ρ, $\widetilde{r_i}$ je esimacija eigen vrednosti, a $\psi_i = \langle \chi_i | v_j + \rangle$. I primenom SWAP testa na dobijenim stanjem dobijamo stanje:

$$\sum_i r_i |\chi_i\rangle \langle \chi_i| \otimes |\widetilde{r_i}\rangle \langle \widetilde{r_i}|$$

Merenjem ovog stanja mi dobojamo eigen vrednost i eigen vektor za *density* matricu ρ. Ako uradimo ovaj proces nad vecem brojem kopija matrice ρ, dobicemo preciznije estimacije eigen vrednosti i eigen vektora.
Sada kada imamo eigen vrednost i eigen vektor možemo da rekonstruišemo matricu za enkodovanje D. Vremenska slozenost ovog algoritma je $O(\log d)$. [9]

5.3 Kvantna neuralna mreza

Neuralne mreze su osnova polja koji se naziva **Duboko učenje** i zato postoji veliku paznja za razvoj istog. U papiru [5], autori su predstavili osnove algoritama za Kvantnu neuralnu mrezu (QNN). Da li su neke primere, neke prednosti i neke nedostatke kvantnog pristupa neuralnim mrezema

Neka unani skup stringova ϕ oblika $z = z_1 z_2 \dots z_n$ gde svako z_i je bit cija 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 nasem setu nalazi sve permutacije ovako opisanog stringa, to jest neka $|\phi| = 2^n$. Predstavicemo kvantni proces koji radi na $n+1$ kubita (poslednji kubit sluzi 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:

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

koja zavise od L parametara $\overrightarrow{\theta} = \theta_L \theta_{L-1} \dots \theta_1$. Za svaki string z kreiracemo pocetno stanje:

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

Primenjivanje unitarne transformacije vraća stanje: $U(\overrightarrow{\theta})\,|z, 1\rangle$ 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 klasicnih neuralnih mreze da "naucimo" proces da vraća tacne vrednosti za dati ulazni string. Pošto merenje izlaznog kubita nije sigurno, odnosno merenje kubita dobijamo tacnu vrednost sa nekom verovatnoćom uvodimo transformaciju:

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

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

Ovde, kao i u klasicnoj neuralnoj mrezi, cilj nam je da nađemo parametar $\overrightarrow{\theta}$ koja vraća tacnu vrednost sa velikom preciznoscu. Slicno kao i prethodnoj postavci imamo: L unitarnih promenljivi sa korespodentnim promenljivama $\overrightarrow{\theta}$, kao i ulazni string z; tada možemo da predstavimo funkciju troška:

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

Mozemo primetiti da ova funkcija troška je linearna i da je minimum u 0, jer je vracema vrednost između -1 i $+1$. Ako pretpostavimo da kvantna neuralna mreza radi savrseno, tako da za svaki ulazni string z, merenje uvek vraća tacnu oznaku. To onda znači da optamalna promenljiva $\overrightarrow{\theta}$ postoji i da je minimum za funkciju trosa 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 $\overrightarrow{\theta}$. Opisacemo proces kako da dođemo do optimalnih parametara $\overrightarrow{\theta}$. Neka pocnemo sa random promenljivom $\overrightarrow{\theta}$ (ili ako imamo neku pretpostavku vrednosti parametara). Izaberimo neki string z^1 iz skupa za traniranje. Primenjujemo kvanti proces nad izabranim stringom:

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

i merimo Y_{n+1} na zadnjem kubitu. Nakon nekoliko merenja možemo da imamo dobru aproksimaciju ocekivane vrednosti od Y_{n+1} i tada izračunavamo

$loss(\overrightarrow{\theta}, z^1)$. Nakon toga, želimo da promenimo parametar $\overrightarrow{\theta}$ tako da smanjimo funkciju troška za string z^1. Poštoje dva nacina da se uradi trazeno: (1) da uradimo pomeraj po nekom uzimanju uzorka u $[\overrightarrow{\theta} - \epsilon, \overrightarrow{\theta} + \epsilon]$ intervalu. (2) da izračunamo izvod funkcije troška po $\overrightarrow{\theta}$ i da se malo pomerimo ka pravcu koji smanjuje funkciju. Ovo nam daje novi parametar $\overrightarrow{\theta^1}$. Sada biramo ponovo iz skupa neki string z^2 i ponovimo prethodni proces ali sa parametrom $\overrightarrow{\theta^1}$. Ovako dobijamo novi parametar $\overrightarrow{\theta^2}$ koji ima manju funkciju troška za string z^2 nego parametar $\overrightarrow{\theta^1}$. Ovako prolazimo kroz proces sve dok ne prođemo kroz ceo skup S. Kao rezultat ovoga generisali smo sekvencu parametara $\overrightarrow{\theta^1}, \overrightarrow{\theta^2}, \ldots \overrightarrow{\theta^S}$. Ako nam je "učenje" parametara uspesno onda bi smo dobila da operator $U(\overrightarrow{\theta^S})$, kada se primeni na stanju $|z, 1\rangle$, vratice stanje koje kada se izmeri na izlazu vraća 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, možemo raci da je model naucio da generalizuje i za neviđenje podatke.

Ovaj proces koji je opisan, primeti ce te, u klasicnom mašinskom učenje zove se "Stohasticko uvenje". U tradicijonalnom mašinskom učenju sa neuronskim mrezama, parametri se prikazuju kao promenljive unutar matrice, koja je linarna u odnosu na unutrašnje vektore. Nad Komponentama tih vektora vrsi se nelinearne transformacije, pre nego sto se mnoze sa ostalim parametrima. Uveđenje dobre ne linearnosti je jedan od glavnih delova uspesne implementiacije modela u klasicnom mašinskom učenju. Ovu osobinu klasnih neuralnim mreza tesko je prebaciti u kvantni sistem, jer je kvantna mehanika, osnova celog koncepta kvantnog računarstva, samo po sebi linearna. U metodi koja je opisana, svaka unitarna opearcija se izvrsava nad izlazom prethodne operacije, pri cemu se između operacija ne izvrsava 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 $\overrightarrow{\theta}$ je ogranica po L, to jest po broju parametara. Ovo je znacajno, jer znači da gradijent ne može da ode u beskonacno i tako izbegavamo veliko problem koji se može desiti klasicnim neuralnim mrezama.

5.3.1 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 određena funkcija oznaka onda možemo da definisemo operator nad komputacionim osnovama kao:

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

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. Takođe, posto funkcija oznaka l(z) može da vrati ili $+1$ ili -1 iz toga imamo $\langle z, 1| U_l^T Y_{n+1} U_l \left|z, 1\right\rangle = l(z)$. 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 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** iskazivanje bilo koje *boolean* funkcije u obliku bitova $b_1 \ldots b_n$:

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

gde su koeficijenati $a \in 0, 1$. Primecuje se da imamo 2^n koeficijenta i posto su oni ili 0 ili 1 da stvarno imao $2^{(2^n)}$ mogucih *boolean* funkcija. Nasa funkcija b može biti ekoponencijalno dugacka. 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 odnostu na kompuntacione baze, koji odgovara nama data funklcija b. Svaka vrednostu u B se mnozi sa X_{n+1} tako da svaka vrednost je komutativna sa ostalim vrednostima. Svaka član, različit od nule, u **Reed-Muller** formuli utice u U_l na kontrolni *bit flip* na izlaznom kubitu.

Ovaj rezultat kvantno reprezentacije ima analog u klasicnoj teoriji reprezentacije [4]. Ona pokazuje da bilo koja *boolean* funkcija ozneke 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 računarima, ali na kvantnim računarima, oni po prirori rade nad Hilbertovim prostorim sa eksponencijalnim dimenzijama. Ali jos nije dokazano da svaka *boolean* funkcija može da se prestavi u kvantno kolo koje nije eksponencionalne dubine. Na tome se trenutno dosta radi u naucnim krugobima.

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, najvise, n operatora nad dva kubita koji su komutativni međusobno, gde je pridodati kubit u svim operatorima nad dva kubita.

5.3.2 Ucenje modela

U ovoj podsekciji ce se objasniti dve potencijalne metode kako da menja parametar $\overrightarrow{\theta}$ tako da se funkcija troška smanjejue. Ako su nam dati parmteri $\overrightarrow{\theta}$ i trening primer z, prvo procenjujemo vrednost trosak od $loss(\overrightarrow{\theta}, z)$. Da bi smo ovo uradili treba da napravimo vise merenje Y_{n+1} za $U(\overrightarrow{\theta})\,|z, 1\rangle$. Da bi smo ovo uspeli sa verovatnoćom većom od 99%, procena of funkcije trosa koja je δ intervaluj od prave vrednosti funkcije troška treba da napravimo najmanje $2/\delta^2$ merenja($\delta \in (0, 1)$).

Nako sto procenimo vrednost funkcije troske želimo da izračunamo gradijent od funkcije trosa u odnosu na $\overrightarrow{\theta}$. Jedan od nacina jeste da menjamo jedenu po jednu promenljivu u $\overrightarrow{\theta}$. Nakon svake promene treba da se izračuna $loss(\overrightarrow{\theta'}, z)$, gde $\overrightarrow{\theta'}$ je različit od $\overrightarrow{\theta}$ za neku malu vrednost u jednoj promenljivi. Ako bi se koristio simetrican izvod funkcije troška 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 izvod za funkciju troška $loss(\overrightarrow{\theta}, z)$ za parametar θ_k, koji je vezan za transformaciju $U_k(\theta_k)$ (koja ima i generalni Pauli-jer operator Σ_k). Sada:

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

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

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

tako da izvod možemo da zapisemo kao:

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

$\mathcal{U}(\overrightarrow{\theta})$ se možemo posmatrati kao kvanto kolo koji sadrzi $2L+2$ unitarnih transformacija. Sada možemo da primenimo $\mathcal{U}(\overrightarrow{\theta})$ nad stanjem $|z,1\rangle$. Ako koristimo dodati kubit, možemo da merimo imaginarni deo izvoda funkcije. Zapocecemo sa stanjem $|z,1\rangle \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ i primenicemo $i\mathcal{U}(\overrightarrow{\theta})$ na dodatim kubitom vrednosti 1. Ovo kreira:

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

Ako primenimo Hademardovu kapiju na dodatim kubitom dobijamo:

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

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

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

tako da, ponavljanjem ovog merenja možemo da dobijemo dobru procenu imaginarnog dela stanja iz kojeg možemo da izvucemo procenu k-te komponente trazenog gradijenta. Ovaj metod izbegava numbricne nepreciznosti

prethodne strategije. Cena ove metode je potreba da dodatim kubitom kao i kvatno kolo $2L + 2$ dubine.

Pošto samo izračunali gradijent, sada treba metod za izmenu $\overrightarrow{\theta}$. Neka je $\overrightarrow{g}$ gradijent funkcije troška po parametru $\overrightarrow{\theta}$. Sada menjamo $\overrightarrow{\theta}$ u pravcu $\overrightarrow{g}$. Sa velicinom "koraka" γ imamo

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

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

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

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

$$\overrightarrow{\theta} | \overrightarrow{\theta} - r(\frac{loss(\overrightarrow{\theta}, z)}{\overrightarrow{g}^2})\overrightarrow{g}$$

Deo uspesne implementacije mašinskog učenja je racionalno odaberemo vrednost stepena učenja.

Ucenje parnosti podskupa Za dati podskup $\mathbb{S}$, unitarna transformacija $U_{P_\mathbb{S}}$ može 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 $U_{P_\mathbb{S}}$. Najlaksi nacin da se ovo postigne jeste da se koriste n parametara

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

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 uspeli da nauce model, njihov argument je da sa povecanjem velicine sistema postaje nemoguce da se nauci kvantni model. Da bi to pokazali, izračunali eksplicitnu formulu su ocekivanu vrednost za Y_{n+1}

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

Sa oznakom $l(z)$ 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 ocekivanje. Poštoje vise verzija izračunate funkcije, koja zavise 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 $\mathbb{S}$ sadrzi svih n bitova. U tom slučaju prosecan trosak za sve stringove je

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

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 cak i ako imamo pristup prosecnom trosku, nijedan metod koji se oslanja na gradijentalni pristup bi mogao biti koriscen za pronalazenje optimalnog ugla za bilo koji primer sa povecim n, gradijent bi radio izvan preciznosti masine u tom slučaju.

5.3.3 Ucenje osobina kvantnih stanja

Sa kvantnom neuralnom mrezom, ocekuje sa da na ulazu može da ima bilo koje kvantno stanje (koje nije izvedeno iz nekog klasicnog podatkja) i da može da nauci neke njegov osobine i da ih izbaci u obliku nekih oznaka. Ne postoji ni jedna klasicna neuralna mreza koja može to da uradi, jer klasicni računari ne mogu da prihvate kvantno stanje kao ulaz. Osnovna ideja je da se n-kubitno stanje $|\psi\rangle$ ubaci u kvantnu neuralnu mrezu sa dodatim kubitom, koji sluzi za citanje rezultata, koji je postavljen na 1. Pa neka nam je data unitarna transformacija $U(\overrightarrow{\theta})$ tako da imamo stanje

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

i onda merimo Y_{n+1}. Cilj ovoga je da namestimo da izlaz ovog merenje bude ekvivalntan nekim dvema oznaka koje oznacavaju neke osobine kvantog 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 negetivne eigen vrednosti. Sa datim kvantnim stanjem $|\psi\rangle$, obelezava se oznakom koja pokazuje da li je ocekivana vrednost Hamiltoijevog operatora pozitivna ili negativna:

$$l(|\psi\rangle) = sign(\langle\psi|\, H\, |\psi\rangle)$$

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 grešku. Mala greška 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), možemo da napravimo da nam greška bude mala.

Posmatrajmo graf gde na svakoj ivici imamo ZZ uparivanje sa koeficijenatom 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:

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

Ako izaberemo $\theta_{ij} = \beta J_{ij}$ imamo operator $U_H(\beta)$ koja osigurava da možemo da označimo trazenu oznaku ako izaberemo malo β
Kvantno stanje $|\psi\rangle$ je u Hilbertovom prostoru sa 2^n dimenzija i ne možemo da ocekujemo da naucimo oznake za svako stanje. Posmatrani Hamiltojev operator ima bitovnu strukturum, tako da možemo 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.

6 Zakljucak

U ovom sam vam prikazao kraci uvod u kvantno računarstvo i kvantno mašinsko učenje. Takođe je dato primer nekoliko metoda kvantnog mašinskog učenja, koja poboljšavaju svoje klasične ekvivalente. Glavni izazov trenutno je praktična primena, koja je strogo vezana za razvoj funkcionalnih kvantnih računara.

Ostavljeno je dosta nepokrivenih metoda kvantnog mašinskog učenja, koje čitalac može da istraži. Savatovao bih 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] George V. Cybenko. "Approximation by superpositions of a sigmoidal function". In: *Mathematics of Control, Signals and Systems* 2 (1989), pp. 303–314.

[5] 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.

[6] 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.

[7] 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.

[8] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. "Quantum algorithms for supervised and unsupervised machine learning". In: *arXiv preprint arXiv:1307.0411* (2013).

[9] 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.

[10] Dan C. Marinescu. *Classical and quantum information*. Academic Press, 2012. Chap. 8. ISBN: 9780123838742; 0123838746.

[11] Dan C. Marinescu. *Classical and quantum information*. Academic Press, 2012. Chap. 5. ISBN: 9780123838742; 0123838746.

[12] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

[13] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. "Quantum support vector machine for big data classification". In: *Physical review letters* 113.13 (2014), p. 130503.

[14] *Shor's Algorithm*. 2021. URL: `https://qiskit.org/textbook/ch-algorithms/shor.html`.

[15] Vlatko Vedral. *Introduction to Quantum Information Science*. Oxford University Press, 2006.