

KLASIFIKASI DATA DENGAN *QUANTUM PERCEPTRON*

Lipantri Mashur Gultom^{1*}

¹Program Studi Teknik Komputer, Politeknik LP3I Medan

Tel: 061-7867311 Fax: 061-7874466

*Email : lipantri@gmail.com

ABSTRAK

Klasifikasi data memiliki beberapa algoritma pengklasifikasian, salah satu algoritma yang ada menggunakan konsep pembelajaran dengan pengawasan. Algoritma pembelajaran ini membimbing input menjadi beberapa kemungkinan output dengan menggunakan konsep jaringan syaraf tiruan *multilayer perceptron*. Model *multilayer perceptron feedforward* yang digunakan memiliki perbedaan dengan algoritma *perceptron* biasa atau klasik. Algoritma ini menggunakan pendekatan komputasi kuantum dalam proses pembelajarannya. Nilai input, hidden dan output neuron yang biasanya menggunakan biner digantikan dengan qubit (*quantum bit*), Sedangkan penentuan nilai bobot dan arsitektur jaringan disesuaikan dengan nilai qubit yang ada. Dalam menentukan tingkat keberhasilan klasifikasi data ini maka digunakan *bechmark* dataset dimana dataset dibagi menjadi dua bagian yaitu dataset pembelajaran dan pengujian. Dari hasil pembelajaran dapat dilihat bahwa algoritma *quantum perceptron* dapat melakukan pembelajaran dengan cepat dan algoritma ini juga dapat mengklasifikasikan data dengan baik.

Kata Kunci : Klasifikasi, Kuantum, Perceptron, Jaringan

PENDAHULUAN

Metode jaringan syaraf tiruan banyak digunakan dalam klasifikasi data dan sangat erat kaitannya dalam *machine learning*. Metode ini digunakan untuk mengorganisasikan dataset menjadi beberapa kategori data yang lebih efektif dan efisien. Dengan hasil yang telah terorganisasi ini maka pemakaian data menjadi informasi menjadi lebih mudah. Algoritma pembelajaran dengan pengawasan adalah salah satu metode yang digunakan dalam proses pengklasifikasian data. Algoritma pembelajaran yang mudah diimplementasikan yaitu model *multilayer perceptron* dengan konsep *feedforward* dimana pemrosesan data memiliki arah bergerak maju tanpa umpan balik. Umumnya arsitektur jaringan yang digunakan terdiri dari tiga layer yaitu layer *input*, *hidden* dan *output*.

Proses pembelajaran *multi layer perceptron* menggunakan perhitungan nilai input, bobot dan output dalam menentukan hasil akhir dari pengklasifikasian data. Nilai input dan output berupa bilangan 1, 0 atau 1, -1 sedangkan nilai bobot dan bias berupa bilangan dalam rentang 0 sampai 1. Namun dalam proses pembelajaran ini dilakukan transformasi nilai input, hidden dan output dengan menggunakan pendekatan komputasi kuantum yaitu dalam bentuk qubit (*quantum bit*). Pendekatan komputasi kuantum digunakan untuk membandingkan apakah model *multi layer perceptron* dengan pendekatan komputasi kuantum masih dapat melakukan klasifikasi data dengan baik atau tidak. Kemudian seberapa cepat algoritma pembelajaran dengan pendekatan komputasi kuantum dapat melakukan klasifikasi data.

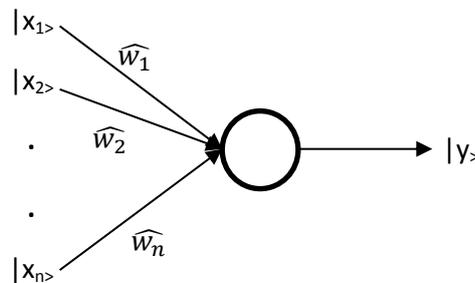
Komputasi Kuantum

Pada komputasi kuantum dikenal *quantum bit (qubit)* yang merupakan basis bilangan terkecil. *Qubit* memiliki dua keadaan yang disimbolkan dengan keadaan $|0\rangle$ dan keadaan $|1\rangle$ sedangkan tanda “ $| \rangle$ ” dikenal dengan notasi Dirac’s. Teori komputasi kuantum sangat terinspirasi oleh fenomena partikel pada mekanika kuantum. Fenomena ini disebut dengan superposisi dimana sebuah partikel dalam mekanika kuantum dapat memiliki dua keadaan sekaligus. Superposisi dari sebuah partikel ini jika ditransformasikan kedalam komputasi dapat berupa bit dimana nilai dari bit hanya terdiri dari 0 atau 1 akan tetapi dalam komputasi kuantum nilai bit dapat berupa 0 atau 1 atau kombinasi dari kedua nilai [6],[7]. Pada komputasi kuantum, qubit $|0\rangle$ memiliki nilai $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ sedangkan qubit $|1\rangle$ memiliki nilai $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$. Sedangkan qubit $\langle 0|$ memiliki nilai $[1 \ 0]$ dan qubit $\langle 1|$ memiliki nilai $[0 \ 1]$. Konsep qubit digunakan pada nilai input dan output node dalam proses pembelajaran.

Quantum Perceptron

Jaringan saraf tiruan merupakan model komputer yang dapat berpikir seperti otak manusia (Du & Swami, 2006; Heaton, 2008). Proses pembelajaran jaringan saraf dapat dibagi menjadi tiga kategori yaitu pembelajaran terawasi (*supervised learning*), tanpa pengawasan (*unsupervised learning*) dan pendekatan hybrid. Jaringan saraf tiruan memiliki arsitektur umum yang terdiri dari lapisan masukan (*input layer*), lapisan tersembunyi (*hidden layer*) dan lapisan keluaran (*output layer*) dimana setiap lapisan memiliki beberapa neuron.

Quantum Neural Network (QNN) merupakan salah satu bentuk dari jaringan saraf tiruan yang menggunakan konsep komputasi kuantum (Gaman, 2011). Dalam metode ini bentuk arsitektur jaringan masih menggunakan jaringan saraf klasik akan tetapi dari penentuan input, bobot, algoritma pembelajaran dan target sudah menggunakan pendekatan komputasi kuantum.



Gambar 1. Neuron pada *quantum perceptron*

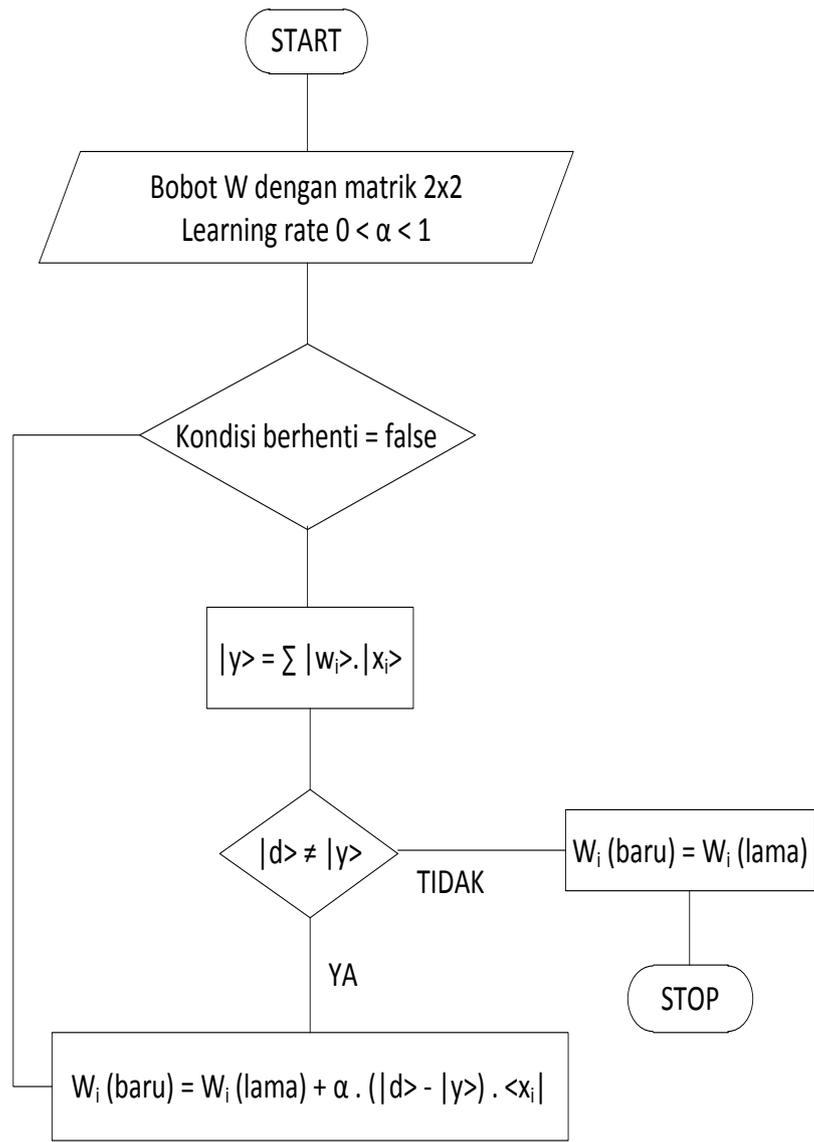
Dari gambar gambar 1 terdapat n-input $|x_1\rangle, |x_2\rangle, \dots, |x_n\rangle$ dan output $|y\rangle$ dimana

$$|Y\rangle = \sum_{i=1}^n \widehat{W}_i |X_i\rangle \dots\dots\dots (1)$$

\widehat{W} merupakan matriks 2x2 dengan basis $(|0\rangle, |1\rangle)$ sedangkan aturan pembelajaran *quantum perceptron* [4],[5] menggunakan persamaan

$$\widehat{w}_i(t + 1) = \widehat{w}_i(t) + \alpha (|d \rangle - |y(t) \rangle) \cdot \langle x_i | \dots\dots\dots (2)$$

$|d \rangle$ adalah *desired output* atau hasil yang diharapkan. Kemudian perbedaan antara hasil nyata dengan yang diharapkan ($|d \rangle - |y(t) \rangle$) akan menghasilkan satu nilai yang disebut dengan error. Proses pencarian nilai error mendekati 0 ini disebut dengan konvergensi (Sagheer & Zidan, 2013). Dengan kata lain hasil konvergensi bergantung pada nilai error dari keadaan antara $|d \rangle$ dan $|y(t) \rangle$. Berikut ini algoritma *quantum perceptron*.

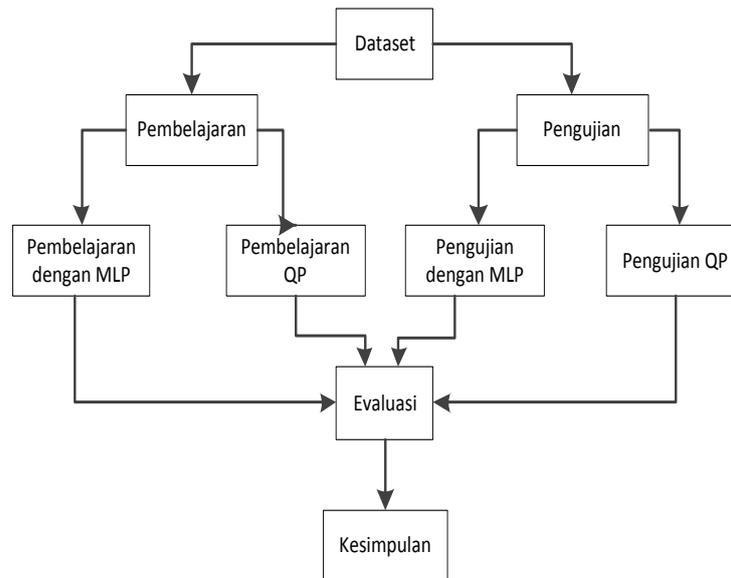


Gambar 2. Algoritma *quantum perceptron*

METODE PENELITIAN

Skema perbandingan MLP dengan QP

Gambar berikut ini merupakan skema perbandingan MLP dengan QP.



Gambar 3. Skema perbandingan MLP dengan QP

Dari skema pada gambar 3, dataset yang digunakan dibagi menjadi dua bagian yaitu dataset untuk pembelajaran dan dataset untuk pengujian. Setiap dataset diproses dengan menggunakan dua algoritma yang berbeda yaitu algoritma MLP dan QP. Kemudian hasil pemrosesan dievaluasi untuk melihat kelebihan dan kekurangan dari setiap algoritma yang kemudian menjadi kesimpulan dari penelitian.

Persiapan Dataset

Dataset yang digunakan dalam penelitian ini diperoleh dari repository *UCI machine learning* dengan nama dataset “*post operative*”. *Post operative* merupakan dataset klasifikasi tempat perawatan pasien pasca operasi karena kemungkinan terjadinya hipotermia adalah suatu yang harus diperhatikan pasca operasi. Dataset ini memiliki jumlah contoh data sebanyak 87 dengan setiap contoh data memiliki 8 atribut. Berikut nama atribut dan kategori nilai dari setiap atribut :

- 1) *L-CORE* (patient's internal temperature in C) : high (> 37), mid (>= 36 and <= 37), low (< 36)
- 2) *L-SURF* (patient's surface temperature in C) : high (> 36.5), mid (>= 36.5 and <= 35), low (< 35)
- 3) *L-O2* (oxygen saturation in %) : excellent (>= 98), good (>= 90 and < 98), fair (>= 80 and < 90), poor (< 80)
- 4) *L-BP* (last measurement of blood pressure) : high (> 130/90), mid (<= 130/90 and >= 90/70), low (< 90/70)
- 5) *SURF-STBL* (stability of patient's surface temperature) : stable, mod-stable, unstable
- 6) *CORE-STBL* (stability of patient's core temperature) : stable, mod-stable, unstable
- 7) *BP-STBL* (stability of patient's blood pressure) : stable, mod-stable, unstable

8) *COMFORT* (patient's perceived comfort at discharge, measured as an integer between 0 and 20)

sedangkan klasifikasi kelas terdiri dari :

- 1) *I* (patient sent to Intensive Care Unit),
- 2) *S* (patient prepared to go home),
- 3) *A* (patient sent to general hospital floor)

Berikut ini contoh 10 data dari 87 dataset dengan 8 atribut dapat dilihat pada tabel 1.

Tabel 1. Contoh 10 data dari dataset *post operative*

1	2	3	4	5	6	7	8
mid	low	excellent	mid	stable	stable	stable	15
high	low	excellent	high	stable	stable	mod-stable	10
mid	mid	excellent	high	stable	stable	stable	10
mid	mid	good	mid	stable	stable	stable	15
mid	low	good	mid	stable	stable	unstable	10
mid	high	excellent	high	stable	stable	stable	10
high	low	good	mid	stable	stable	unstable	15
mid	low	excellent	high	stable	stable	mod-stable	05
high	mid	excellent	mid	unstable	unstable	stable	10
mid	high	good	mid	stable	stable	stable	10

sebelum dataset digunakan maka dilakukan pengkodean kedalam bentuk biner seperti pada tabel 2 berikut. setelah dilakukan pengkodean kedalam biner maka dataset awal berubah menjadi seperti tabel 3 berikut ini. sedangkan klasifikasi kelas dikodekan kedalam biner seperti tabel 4 berikut ini. Dari hasil pengkodean dataset kedalam biner dapat ditentukan arsitektur jaringan seperti gambar 4 berikut.

Perhitungan bobot pada QP

Perhitungan bobot pada proses pembelejaraan QP berbeda dengan perhitungan bobot pada MLP. Karena bobot pada QP menggunakan matriks 2x2 dengan nilai acak $0 < w < 1$ sedangkan input dan output node menggunakan konsep qubit. Berikut ini perhitungan sederhana dengan menggunakan algoritma pada gambar 2 dengan menggunakan contoh data pertama pada dataset tabel 3. Pertama ditentukan nilai input $|x_1\rangle$ sampai dengan $|x_{16}\rangle$ yaitu $|0\rangle, |1\rangle, |0\rangle, |0\rangle, |1\rangle, |1\rangle, |0\rangle, |1\rangle, |1\rangle, |0\rangle, |1\rangle, |0\rangle, |1\rangle, |0\rangle, |1\rangle, |1\rangle$ (data pertama pada tabel 3) dan nilai output $|d_1\rangle$ dan $|d_2\rangle$ yaitu $|0\rangle, |0\rangle$ (kelas A). Kemudian inisialisasi nilai awal bobot w dan v secara random dalam bentuk matriks 2x2, misalkan untuk bobot $w_{i,j}$ sebagai berikut $w_{1,1} = \begin{bmatrix} 0,1 & 0 \\ 0 & 0,2 \end{bmatrix}$ sampai $w_{1,16} = \begin{bmatrix} 0 & 0 \\ 0 & 0,1 \end{bmatrix}$ dan $w_{2,1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$ sampai $w_{2,16} = \begin{bmatrix} 0 & 0 \\ 0,3 & 0,1 \end{bmatrix}$. Kemudian bobot $v_{i,j}$ sebagai berikut $v_{1,1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, v_{1,2} = \begin{bmatrix} 0 & 0,1 \\ 0 & 0,1 \end{bmatrix}, v_{2,1} = \begin{bmatrix} 0 & 0,1 \\ 0 & 0,1 \end{bmatrix}$ dan $v_{2,2} = \begin{bmatrix} 0 & 0,1 \\ 0 & 0,1 \end{bmatrix}$.

Dengan persamaan (1) dapat dihitung nilai $|z_1\rangle, |z_2\rangle$ dan $|y_1\rangle, |y_2\rangle$. Jika hasil akhir dari $|y_1\rangle \neq |d_1\rangle$ atau $|y_2\rangle \neq |d_2\rangle$ maka dilakukan perubahan bobot dengan persamaan (2). Perhitungan dan perubahan bobot dilakukan sampai kondisi $|y\rangle = |d\rangle$ atau dengan kata lain nilai error = 0.

Tabel 2. Kode biner dari setiap atribut

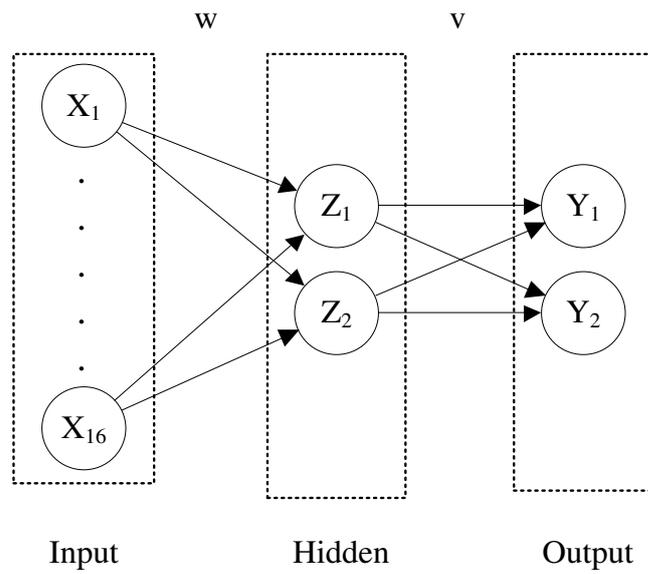
No atribut	Nilai	Kode biner
1	Low	00
	Mid	01
	High	10
2	Low	00
	Mid	01
	High	10
3	Poor	00
	Fair	01
	Good	10
	excellent	11
4	Low	00
	Mid	01
	High	10
5	Unstable	00
	Mod-stable	01
	Stable	10
6	Unstable	00
	Mod-stable	01
	Stable	10
7	Unstable	00
	Mod-stable	01
	Stable	10
8	05	00
	07	10
	10	10
	15	11

Tabel 3. Contoh 10 data setelah dilakukan pengkodean biner

1	2	3	4	5	6	7	8
01	00	11	01	10	10	10	11
10	00	11	10	10	10	01	10
01	01	11	10	10	10	10	10
01	01	10	01	10	10	10	11
01	00	10	01	10	10	00	10
01	10	11	10	10	10	10	10
10	00	10	01	10	10	00	11
01	00	11	10	10	10	01	00
10	01	11	01	00	00	10	10
01	10	10	01	10	10	10	10

Tabel 4. Klasifikasi kelas dalam kode biner

Kelas	Kode biner
A	00
I	01
S	10



Gambar 4. Arsitektur jaringan syaraf

HASIL DAN PEMBAHASAN

Hasil perbandingan pembelajaran MLP dengan QP

Dari hasil pembelajaran dengan nilai *learning rate* α yang bervariasi dan dataset pembelajaran yang sama dapat dilihat perbedaan proses pembelajaran MLP dengan QP pada tabel 5 berikut ini.

Tabel 5. Perbandingan pembelajaran MLP dan QP dengan α

α	Epoch		
	MLP	QP	Selisih
0,1	9917	8546	1371
0,2	8123	7918	205
0,3	7129	6521	608
0,4	6813	5291	1522
0,5	5816	4296	1520
0,6	4529	3785	744
0,7	3621	2943	678
0,8	2715	1821	894
0,9	1927	1294	633

B. Hasil perbandingan pengujian MLP dengan QP

Proses pengujian dilakukan untuk melihat keberhasilan algoritma dalam mengklasifikasikan data kedalam 3 kelas yaitu kelas A, I dan S. Hasil pengujian dapat dilihat pada tabel 6 berikut ini.

Tabel 6. Perbandingan hasil pengujian MLP dengan QP

Algoritma	Persentase
MLP	93,33 %
QP	95,55 %

KESIMPULAN DAN SARAN

Dari hasil pengujian, algoritma *multi layer perceptron* dengan pendekatan kuantum dapat mengklasifikasikan data dengan baik. Pada proses pembelajaran dapat dibuktikan bahwa *quantum perceptron* (QP) lebih cepat daripada *multi layer perceptron* (MLP) sedangkan pada proses pengujian QP juga masih lebih baik dibandingkan MLP. Namun demikian hasil pengujian ini masih terdapat keterbatasan dimana perlu dilakukan pengujian yang lebih mendalam terhadap arsitektur jaringan syaraf terutama pada penentuan jumlah *hidden node*. Serta penentuan nilai learning rate yang lebih bervariasi lagi.

DAFTAR PUSTAKA

- Du, K.L & M.N.S. Swamy. (2006). *Neural Networks in a Softcomputing Framework*. Springer-Verlag London Limited.
- Garman, J.A. (2011). *A Heuristic Review of Quantum Neural Networks*. Tesis. Imperial College London.
- Heaton, J. (2008). *Introduction to Neural Networks with Java*. 2nd Edition. Heaton Research, Inc.
- Nayak, S, Nayak, S & Singh, J.P. (2011). *Computational Power of Quantum Artificial Neural Network*. IJCST Vol. 2(2).
- Sagheer, A & Zidan, M. (2013). *Autonomous Quantum Perceptron Neural Network*.(Online) <http://arxiv.org/abs/1312.4149v1> (10 Juni 2014).
- Stenholm, S & Suominen, K.A. (2005). *Quantum Approach To Informatics*. John Wiley & Sons, Inc., Hoboken, New Jersey.
- Yanofsky, N.S & Manucci, M.A. (2008). *Quantum Computing For Computer Scientists*. Cambridge University Press 32 Avenue of the Americas, New York, NY 10013-2473, USA.