

PENGENALAN KARAKTER DENGAN MENGGUNAKAN HAMMING NETWORK

Thiang

Jurusan Teknik Elektro, Universitas Kristen Petra
Siwalankerto 121-131, Surabaya, Indonesia
E-mail: thiang@petra.ac.id

Abstrak

Makalah ini menjelaskan tentang aplikasi dari Hamming network untuk pengenalan karakter. Hamming network adalah salah satu model struktur dari Artificial Neural Network yang terdiri atas 2 layer yaitu feedforward layer dan recurrent layer. Dalam sistem ini, input dari Hamming network berupa image dari karakter berukuran 46x46 pixel dengan format image biner. Output dari Hamming network terdiri atas 26 neuron dimana masing-masing neuron mewakili satu karakter yaitu huruf A sampai Z. Ada tiga jenis karakter yang diuji dalam sistem ini, yaitu karakter Times New Roman, Arial Rounded Bold dan Courier New. Hasil pengujian menunjukkan bahwa untuk aplikasi ini, Hamming network baik sekali mengenal input data yang terdapat dalam data training dan cukup robust terhadap noise. Karakter Times New Roman menghasilkan tingkat pengenalan yang lebih baik.

Kata kunci: artificial neural network, hamming network, pengenalan karakter

1. Pendahuluan

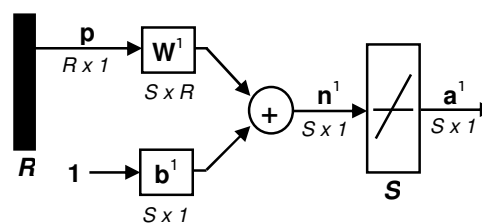
Artificial Neural Network (ANN) telah banyak diaplikasikan dalam berbagai bidang. Mulai dari bidang aerospace, otomotif, banking, financial, elektronika, manufaktur, robotika, medika, telekomunikasi, transportasi dan lain-lain (1). Tetapi pada umumnya, bila dirangkum, ANN diaplikasikan untuk 4 jenis aplikasi yaitu pengenalan pola (pattern recognition), klasifikasi (classification), estimasi sebuah fungsi (function estimation) dan pengembalian pola (pattern recall). Pada aplikasi pengenalan pola, ANN berfungsi untuk mengenal pola yang tidak diketahui berdasarkan informasi pola referensi. Pada aplikasi klasifikasi, ANN berfungsi untuk mengklasifikasi data input yang tidak diketahui ke dalam group-group (class). Pada aplikasi fungsi estimasi, ANN berfungsi sebagai fungsi pendekatan yang menyatakan relasi antara input dan output. Pada aplikasi pattern recall, ANN berfungsi untuk recall pola yang lengkap dari input pola yang mungkin hanya sebagian saja.

Model struktur dari ANN dan metode trainingnya juga telah banyak dikembangkan. Salah satu model struktur ANN tersebut adalah Hamming Network. Hamming network adalah suatu model neural network yang didisain khusus untuk menyelesaikan masalah pengenalan pola dengan input dari neural network dalam bentuk biner. Dengan demikian setiap elemen dari vektor input neural network hanya mempunyai dua kemungkinan nilai input.

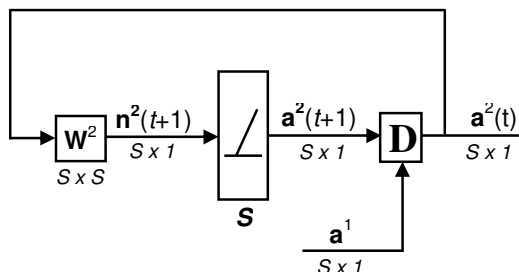
Karena itu dalam makalah ini akan dipaparkan tentang pengaplikasian dari Hamming network pada aplikasi sederhana yaitu pengenalan karakter. Dalam sistem yang telah dirancang, input data berupa image dari karakter tersebut dan output berupa jenis karakter yang dikenal. Untuk lebih

jelas, perancangan hamming network untuk sistem pengenalan karakter ini akan dijelaskan lebih detail pada bagian ketiga dari makalah ini. Sedangkan bagian kedua dari makalah ini akan menjelaskan tentang model struktur Hamming network itu sendiri. Bagian keempat dari makalah ini akan memaparkan hasil pengujian yang telah dilakukan terhadap sistem yang telah dirancang. Dan makalah ini akan ditutup dengan kesimpulan yang didapatkan dari hasil pengujian dan perancangan sistem ini.

2. Hamming Network



Gambar 1. Layer Pertama dari Hamming Network (Feedforward Layer)



Gambar 2. Layer Kedua dari Hamming Network (Recurrent Layer)

Hamming network terdiri atas dua layer. Layer pertama disebut dengan feedforward layer.

Sedangkan layer kedua adalah recurrent layer. Gambar 1 menunjukkan gambar dari hamming network standar.

2.1 Feedforward Layer

Pada layer pertama dari Hamming network (feedforward layer), fungsi aktivasi atau fungsi transfer yang digunakan adalah fungsi transfer linier dimana $\mathbf{a} = \mathbf{n}$. Sehingga output dari feedforward layer (layer pertama) adalah sebagai berikut:

$$\mathbf{a}^1 = \mathbf{W}^1 \mathbf{p} + \mathbf{b}^1 \quad (1)$$

Dimana \mathbf{p} adalah vektor input yang akan dikenali, \mathbf{W}^1 adalah bobot koneksi antara input neural network dengan semua neuron yang ada pada feedforward layer dan \mathbf{b}^1 adalah bobot koneksi input bias pada masing-masing neuron pada feedforward layer. R adalah jumlah input dari neural network dan S adalah jumlah neuron pada feedforward layer yang juga menentukan jumlah output dari feedforward layer ini. Feedforward layer menghasilkan korelasi antara setiap pola prototipe (data training) dengan vektor input neural network. Oleh karena itu, untuk menghasilkan hal ini, bobot koneksi pada feedforward layer (\mathbf{W}^1), diset sama dengan pola prototipe (data training). Bobot koneksi untuk input bias (\mathbf{b}^1) pada feedforward layer, diset sama dengan R . Neuron pada feedforward layer yang mempunyai output terbesar menunjukkan bahwa pola prototipe yang bersesuaian dengan neuron tersebut mempunyai jarak paling dekat dengan pola vektor input.

2.2 Recurrent Layer

Layer kedua dari Hamming network adalah recurrent layer. Layer ini dikenal sebagai layer kompetisi. Setelah proses kompetisi berlangsung, hanya akan ada satu neuron di layer ini yang mempunyai output tidak nol, sedangkan neuron yang lain akan mempunyai output nol.

Fungsi aktivasi atau fungsi transfer yang digunakan dalam recurrent layer ini adalah fungsi linier positif (**poslin**) dimana persamaannya dapat ditulis sebagai berikut:

$$a = \begin{cases} 0 & \text{untuk } n < 0 \\ n & \text{untuk } n \geq 0 \end{cases} \quad (2)$$

Sehingga output dari recurrent layer dapat dituliskan dengan persamaan berikut:

$$\mathbf{a}^2(t+1) = \text{poslin}(\mathbf{W}^2 \mathbf{a}^2(t)) \quad (3)$$

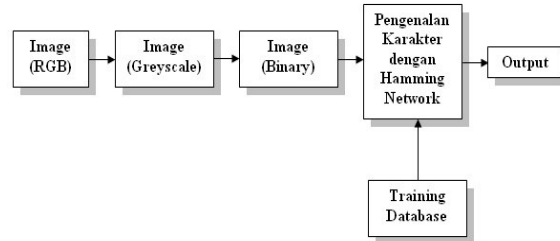
dimana $\mathbf{a}^2(0) = \mathbf{a}^1$ yang berarti bahwa inisialisasi awal output layer kedua diambil dari output layer pertama \mathbf{a}^1 . Bobot koneksi layer kedua \mathbf{W}^2 diset dengan nilai sebagai berikut:

$$w_{ij}^2 = \begin{cases} 1 & \text{jika } i = j \\ -\epsilon & \text{jika } i \neq j \end{cases} \quad (4)$$

dimana $0 < \epsilon < \frac{1}{S-1}$

3. Aplikasi pada Sistem Pengenalan Karakter

Secara umum blok diagram sistem pengenalan karakter dengan menggunakan Hamming network dapat dilihat pada gambar 3.



Gambar 3. Blok Diagram Sistem Pengenalan Karakter dengan Hamming Network

Input dari sistem ini berupa image dengan format RGB dan berukuran 46x46 pixel. Tahap pertama yang dilakukan adalah mengubah image dari format RGB menjadi greyscale. Proses ini dilakukan dengan menggunakan persamaan berikut:

$$Gray = 0.299R + 0.587G + 0.114B \quad (5)$$

Proses selanjutnya adalah mengubah image dengan format greyscale menjadi image biner. Hal ini perlu dilakukan karena Hamming network dirancang untuk pengenalan pola dengan format data biner. Proses pengubahan image greyscale menjadi image biner dilakukan dengan menggunakan fungsi threshold yang direpresentasikan oleh persamaan berikut:

$$g(x, y) = \begin{cases} 0 & \text{if } f(x, y) < T \\ 255 & \text{if } f(x, y) \geq T \end{cases} \quad (6)$$

Image biner ini yang akan menjadi input dari Hamming network untuk dikenali sebagai salah satu karakter yang ada dalam database (data training). Karena image yang digunakan berukuran 46x46 pixel, maka dalam satu pola karakter terdapat 2116 informasi sehingga pada struktur hamming network yang dirancang terdapat 2116 input. Dalam sistem ini juga dibatasi hanya untuk pengenalan 26 karakter yaitu 26 huruf besar dari A sampai Z dan masing-masing karakter diwakili oleh satu pola prototipe (data training). Karena itu, pada Hamming network yang dirancang terdapat 26 neuron output. Jumlah neuron output akan bertambah bila jumlah data training bertambah. Model dari Hamming network yang telah dirancang dapat dilihat pada gambar 1 dan 2 dengan nilai $R = 2116$ dan nilai $S = 26$. Dengan demikian matrik \mathbf{W}^1 akan berdimensi

26x2116 dan berisikan pola prototipe (data training). Sedangkan matrik W^2 berdimensi 26x26. Matrik b^1 merupakan sebuah matrik kolom dengan semua elemennya adalah 2116. Sebagaimana telah disebutkan sebelumnya, output dari feedforward layer akan menghasilkan korelasi antara vektor input dengan masing-masing pola prototipe. Kemudian output ini akan menjadi nilai awal dari output pada layer kedua yaitu recurrent layer. Layer kedua ini akan diiterasi terus, sampai hanya satu neuron yang mempunyai output tidak nol. Neuron yang mempunyai output tidak nol tersebut menunjukkan pola yang paling sesuai dengan pola vektor input. Dalam sistem ini, ada tiga jenis karakter yang digunakan yaitu Times New Roman, Arial Rounded Bold dan Courier New. Karakter jenis Times New Roman yang dipilih untuk menjadi data training sedangkan yang lain untuk kepentingan pengujian.

4. Pengujian Sistem

Beberapa pengujian telah dilakukan antara lain:

- Pengujian dengan data training karakter Times New Roman
- Pengujian dengan data training karakter Arial Rounded Bold
- Pengujian dengan data training karakter Courier New
- Pengujian dengan variasi penambahan noise pada image

4.1 Pengujian dengan Data Training Karakter Times New Roman

Pada pengujian ini, data Data Training diambil dari karakter Times New Roman. Masing-masing karakter diwakili oleh satu data training. Jadi dalam pengujian ini ada 26 data training. Pengujian dilakukan sebanyak tiga kali untuk masing-masing jenis karakter yaitu Times New Roman, Arial Rounded Bold dan Courier New. Hasil pengujian dapat dilihat pada tabel 1, 2 dan 3.

Tabel 1. Hasil Pengujian dengan Data Training Karakter Times New Roman dan Input Karakter Times New Roman

Input Karakter	Hasil Pengenalan	Input Karakter	Hasil Pengenalan
A	A	N	N
B	B	O	O
C	C	P	P
D	D	Q	Q
E	E	R	R
F	F	S	S
G	G	T	T
H	H	U	U
I	I	V	V
J	J	W	W
K	K	X	X
L	L	Y	Y
M	M	Z	Z

Dari hasil pengujian di atas, terlihat bahwa sistem dapat mengenali karakter dengan tingkat keberhasilan 100%.

Tabel 2. Hasil Pengujian dengan Data Training Karakter Times New Roman dan Input Karakter Arial Rounded Bold

Input Karakter	Hasil Pengenalan	Input Karakter	Hasil Pengenalan
A	A	N	H
B	H	O	O
C	G	P	P
D	O	Q	O
E	U	R	R
F	P	S	S
G	G	T	T
H	H	U	O
I	I	V	V
J	J	W	W
K	K	X	X
L	L	Y	Y
M	M	Z	Z

Dari hasil pengujian di atas, sistem hanya dapat mengenali 18 karakter dari 26 karakter yang ada dengan tingkat keberhasilan 69,23%

Tabel 3. Hasil Pengujian dengan Data Training Karakter Times New Roman dan Input Karakter Courier New

Input Karakter	Hasil Pengenalan	Input Karakter	Hasil Pengenalan
A	A	N	R
B	P	O	D
C	C	P	P
D	P	Q	U
E	P	R	P
F	E	S	S
G	P	T	I
H	B	U	V
I	I	V	Y
J	J	W	P
K	F	X	X
L	L	Y	I
M	N	Z	I

Dari hasil pengujian di atas, sistem hanya dapat mengenali 8 karakter dari 26 karakter yang ada dengan tingkat keberhasilan 30,77%.

4.2 Pengujian dengan Data Training Karakter Arial Rounded Bold

Pada pengujian ini, data training diambil dari karakter Arial Rounded Bold. Masing-masing karakter diwakili oleh satu data training. Jadi dalam pengujian ini ada 26 data training. Pengujian dilakukan sebanyak tiga kali untuk masing-masing jenis karakter yaitu Times New Roman, Arial Rounded Bold dan Courier New. Hasil pengujian dapat dilihat pada tabel berikut.

Tabel 4. Hasil Pengujian dengan Data Training Karakter Arial Rounded Bold dan Input Karakter Times New Roman

Input Karakter	Hasil Pengenalan	Input Karakter	Hasil Pengenalan
A	I	N	X
B	S	O	O
C	L	P	F
D	S	Q	O
E	F	R	R
F	F	S	S
G	C	T	T
H	B	U	L
I	I	V	Y
J	I	W	W
K	K	X	X
L	L	Y	Y
M	M	Z	Z

Dari hasil pengujian di atas, sistem hanya dapat mengenali 13 karakter dari 26 karakter yang ada dengan tingkat keberhasilan 50%.

Tabel 5. Hasil Pengujian dengan Data Training Karakter Arial Rounded Bold dan Input Karakter Arial Rounded Bold

Input Karakter	Hasil Pengenalan	Input Karakter	Hasil Pengenalan
A	A	N	N
B	B	O	O
C	C	P	P
D	D	Q	Q
E	E	R	R
F	F	S	S
G	G	T	T
H	H	U	U
I	I	V	V
J	J	W	W
K	K	X	X
L	L	Y	Y
M	M	Z	Z

Dari hasil pengujian di atas, sistem hanya dapat mengenali semua karakter dengan tingkat keberhasilan 100%

Tabel 6. Hasil Pengujian dengan Data Training Karakter Arial Rounded Bold dan Input Karakter Courier New

Input Karakter	Hasil Pengenalan	Input Karakter	Hasil Pengenalan
A	A	N	P
B	F	O	I
C	V	P	F
D	P	Q	L
E	F	R	F
F	F	S	F
G	F	T	T
H	F	U	V
I	T	V	Y
J	J	W	V
K	F	X	X
L	A	Y	Y
M	Y	Z	I

Dari hasil pengujian di atas, sistem hanya dapat mengenali 6 karakter dari 26 karakter yang ada dengan tingkat keberhasilan 28,08%

4.3 Pengujian dengan Data Training Karakter Courier New

Pada pengujian ini, data training diambil dari karakter Courier New. Masing-masing karakter diwakili oleh satu data training. Jadi dalam pengujian ini ada 26 data training. Pengujian dilakukan sebanyak tiga kali untuk masing-masing jenis karakter yaitu Times New Roman, Arial Rounded Bold dan Courier New. Hasil pengujian dapat dilihat pada tabel berikut.

Tabel 7. Hasil Pengujian dengan Data Training Karakter Courier New dan Input Karakter Times New Roman

Input Karakter	Hasil Pengenalan	Input Karakter	Hasil Pengenalan
A	A	N	M
B	H	O	M
C	C	P	P
D	D	Q	C
E	F	R	H
F	P	S	S
G	C	T	I
H	H	U	D
I	I	V	V
J	J	W	J
K	K	X	Y
L	L	Y	V
M	V	Z	Y

Dari hasil pengujian di atas, sistem hanya dapat mengenali 10 karakter dari 26 karakter yang ada dengan tingkat keberhasilan 38,46%

Tabel 8. Hasil Pengujian dengan Data Training Karakter Courier New dan Input Karakter Arial Rounded Bold

Input Karakter	Hasil Pengenalan	Input Karakter	Hasil Pengenalan
A	A	N	M
B	R	O	C
C	C	P	D
D	Q	Q	Q
E	H	R	H
F	F	S	R
G	C	T	I
H	M	U	M
I	I	V	U
J	J	W	A
K	W	X	X
L	Q	Y	V
M	V	Z	X

Dari hasil pengujian di atas, sistem hanya dapat mengenali 7 karakter dari 26 karakter yang ada dengan tingkat keberhasilan 26,92%

Tabel 9. Hasil Pengujian dengan Data Training Karakter Courier New dan Input Karakter Courier New

Input Karakter	Hasil Pengenalan	Input Karakter	Hasil Pengenalan
A	A	N	N
B	B	O	O
C	C	P	P
D	D	Q	Q
E	E	R	R
F	F	S	S
G	G	T	T
H	H	U	U
I	I	V	V
J	J	W	W
K	K	X	X
L	L	Y	Y
M	M	Z	Z

Dari hasil pengujian di atas, sistem dapat mengenali semua karakter dengan tingkat keberhasilan 100%

4.4 Pengujian dengan Variasi Noise Image

Pengujian ini dilakukan untuk melihat bagaimana keberhasilan dari Hamming network bila dalam input image terdapat noise. Pengujian ini hanya dilakukan untuk data training yang diambil dari karakter Times New Roman karena bila dilihat hasil pengujian sebelumnya, data ini yang memberikan hasil yang terbaik.

Penambahan noise dilakukan dengan memberikan noise uniform dan noise gaussian pada input image.

Tabel 10. Hasil Pengujian dengan Data Training Karakter Times New Roman dan Input Karakter Times New Roman dengan Noise Uniform

Input Karakter	Hasil Pengenalan	Input Karakter	Hasil Pengenalan
A	A	N	N
B	B	O	O
C	C	P	P
D	D	Q	Q
E	E	R	R
F	F	S	S
G	G	T	T
H	H	U	U
I	I	V	V
J	J	W	W
K	K	X	X
L	L	Y	Y
M	M	Z	Z

Dari hasil pengujian di atas, sistem dapat mengenali semua karakter dengan tingkat keberhasilan 100% walaupun input image telah ditambah dengan noise uniform.

Berikut rangkuman hasil pengujian yang telah dilakukan untuk input image yang sudah ditambah dengan noise.

Tabel 11. Hasil Pengujian dengan Data Training Karakter Times New Roman dan Input Karakter dengan Noise Uniform dan Gaussian

Input	Tingkat Pengenalan		
	Noise Uniform	Noise Gaussian	Tanpa Noise
Times New Roman	100%	100%	100%
Arial Rounded Bold	57,69%	61,53%	69,23%
Courier New	38,46%	26,92%	30,77%

Dari tabel di atas terlihat bahwa untuk data input karakter yang ada dalam data training, tingkat keberhasilan pengenalan tidak dipengaruhi oleh noise.

5. Kesimpulan

Dari hasil pengujian yang telah dilakukan, dapat diambil kesimpulan bahwa untuk aplikasi yang telah dirancang ini, Hamming network baik sekali untuk mengenali karakter yang ada dalam data training dan cukup robust terhadap noise. Akan tetapi untuk data yang tidak terdapat dalam data training, Hamming network menghasilkan tingkat pengenalan yang dapat dikatakan masih kurang baik.

Dari ketiga jenis karakter yang diuji, Times new Roman menghasilkan tingkat pengenalan yang lebih baik. Tentunya untuk menghasilkan tingkat pengenalan 100% untuk ketiga jenis karakter, maka dalam data training harus ada setiap data dari ketiga jenis karakter tersebut. Untuk pengembangan lebih lanjut, dalam rangka memperbaiki hasil pengenalan dari data yang tidak ada dalam data training, mungkin dapat dilakukan dengan menambah metode ekstraksi fitur data.

Daftar Pustaka

- [1] Hagan, Martin T. *Neural Network Design*. Boston, USA: PWS Publishing Company, 1996.
- [2] Haykin, Simon. *Neural Networks, A Comprehensive Foundation*, Prentice Hall, 1999.
- [3] Patterson, Dan W. *Artificial Neural Networks: Theory And Applications*. Singapore: Prentice Hall, 1996.
- [4] Zalzala, A.M.S. *Neural Networks for Robotic Control: Theory and Applications*, London: Ellis Horwood, 1996.