PENGENALAN KARAKTER ALFANUMERIK MENGGUNAKAN METODE BACKPROPAGARATION

Amriana[1]
Program Studi D1 Teknik Informatika Jurusan Teknik Elektro Fakultas Teknik UNTAD

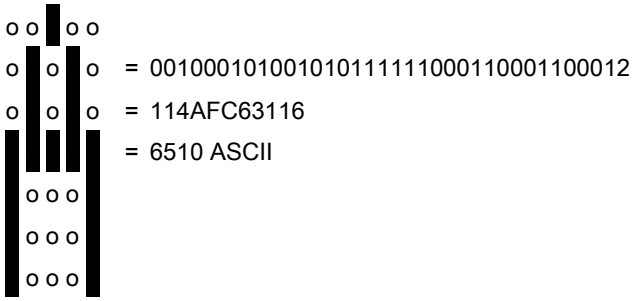
ABSTRAK

Jaringan saraf tiruan untuk aplikasi pengenalan karakter alfanumerik terdiri dari 3 lapisan yaitu lapisan input, sebuah lapisan tersembunyi dan sebuah lapisan output. Jaringan propagasi balik akan menerima input biner sehingga data harus diatur sebagai sekumpulan angka (vektor). Karakter alfanumerik diterjemahkan kedalam bentuk matriks ukuran 5 x 7 yang melambangkan pixel-pixel. Data dimasukkan dalam bentuk biner, angka 1 menunjukkan pixel berisi citra, angka 0 menunjukkan kosong. Pengujian jaringan dilakukan dengan menggunakan program aplikasi MATLAB yang akan mengenali karakter alfanumerik.

Kata Kunci: Karakter Alfanumerik, Metode Propagarasi Balik

I. Latar Belakang

Propagasi balik merupakan metode yang sangat baik dalam menangani masalah pengenalan pola-pola kompleks, salah satu contoh adalah pengenalan karakter alfanumerik. Diasumsikan akan dibuat sebuah desain untuk mengenali karakter alfanumerik dengan jalan menerjemahkan sebuah matriks 5 x 7 yang berisikan bilangan-bilangan biner yang menggambarkan citra pixel pemetaan bit dari sebuah karakter alfanumerik ke dalam kode ASCII 8 bit.



Gambar 1. Setiap citra karakter dipetakan ke dalam kode ASCII yang ditunjuknya.

Tidak tersedianya fungsi matematika yang jelas yang bisa menghasilkan translasi yang diinginkan membuat masalah penerjemahan karakter alfanumerik ini menjadi cukup memusingkan karena untuk menghasilkan translasi yang salah satu caranya adalah dengan melakukan korelasi pixel demi pixel dibutuhkan waktu yang relatif lebih lama, salah satu cara dengan menggunakan sebuah tabel *lookup* berupa sebuah *array linear*. Akan tetapi di dalam sistem nyata ada situasi yang tidak dapat ditangani dengan metoda ini karena sangat dimungkinkan sebuah citra yang dibaca ternyata mengandung *noise* atau kurang lengkap. Adanya *noise* atau citra yang kurang lengkap

mengakibatkan algoritma *lookup* akan mengembalikan kode ASCII yang salah sebab didalam pencocokan antara pola input dan pola asosiasi target tidak menghasilkan karakter yang dimaksud.

Masalah citra yang mengandung *noise* atau kurang lengkap dapat diselesaikan dengan menambah sejumlah program (berarti penambahan waktu yang dihabiskan CPU) ke dalam algoritma tabel lookup. Penambahan program ini akan memeperbaiki kemampuan komputer untuk menebak pada karakter mana yang seharusnya terletak citra yang mengandung *noise*. Pada bit tunggal adalah cukup mudah untuk menemukan dan mengoreksi *error*. Namun demikian akan menjadi sulit bila *error* terjadi bukan hanya pada bit tunggal, melainkan pada banyak bit. selain itu, pengeliminasian *noise* dari pola input untuk deiterjemahkan ke dalam kode ASCII akan menghabiskan banyak waktu CPU.

Solusi untuk masalah pengenalan karakter alfanumerik adalah dengan menggunakan jaringan saraf tiruan. Dengan memanfaatkan sifat paralel jaringan saraf tiruan maka waktu yang dibutuhkan oleh sebuah prosesor sekuensial untuk melakukan pemetaan dapat dikurangi.

II. Rumusan Masalah

Jaringan saraf tiruan untuk pengenalan karakter alfanumerik yang harus diperhatikan antara lain adalah ukuran jaringan yang dirancang, pembawaan input dan output, tipe pelatihan dan waktu operasi rutin dari sistem berjalan. Metode propagasi balik mempunyai pembawaan dari input dan outputnya dalam bentuk diskrit (*biner*) dan tipe pelatihan terawasi.

III. Metodologi

Perangkat lunak pengembang jaringan saraf tiruan banyak tersedia, untuk pengenalan karakter alfanumerik membangun simulator jaringan saraf tiruan dengan bahasa pemrograman Matlab (*toolbox Neural Network*). Langkah-langkah pengumpulan data untuk membangun aplikasi pengenal karakter alfnumerik adalah sebagai berikut :

1. Membuat rancangan data (*input* dan *output*) yang akan digunakan sebagai latihan.

Tabel 1. Set Pelatihan yang digunakan untuk melatih jaringan

Vektor input (35 Digit)	Vektor Output	Target
12345 6789 12345 6789 12345 6789 12345	123456	karakter
00100 01010 01010 11111 10001 10001 10001	000001	A
11110 10001 10001 11110 10001 10001 11110	000010	B
01110 10001 10000 10000 10000 10001 01110	000011	C
00100 01010 01010 11111 10001 10001 10001	000001	A
11110 10001 10001 11110 10001 10001 11110	000010	B
01110 10001 10000 10000 10000 10001 01110	000011	C
00100 01010 01010 11111 10001 10001 10001	000001	A
11110 10001 10001 11110 10001 10001 11110	000010	B
01110 10001 10000 10000 10000 10001 01110	000011	C

00100 01010 01010 11111 10001 10001 10001	000001	A
11110 10001 10001 11110 10001 10001 11110	000010	B
01110 10001 10000 10000 10000 10001 01110	000011	C
00100 01010 01010 11111 10001 10001 10001	000001	A
11110 10001 10001 11110 10001 10001 11110	000010	B
01110 10001 10000 10000 10000 10001 01110	000011	C
00100 01010 01010 11111 10001 10001 10001	000001	A
11110 10001 10001 11110 10001 10001 11110	000010	B
01110 10001 10000 10000 10000 10001 01110	000011	C
00100 01010 01010 11111 10001 10001 10001	000001	A
11110 10001 10001 11110 10001 10001 11110	000010	B
01110 10001 10000 10000 10000 10001 01110	000011	C
00100 01010 01010 11111 10001 10001 10001	000001	A
11110 10001 10001 11110 10001 10001 11110	000010	B
01110 10001 10000 10000 10000 10001 01110	000011	C
00100 01010 01010 11111 10001 10001 10001	000001	A
11110 10001 10001 11110 10001 10001 11110	000010	B
01110 10001 10000 10000 10000 10001 01110	000011	C
00100 01010 01010 11111 10001 10001 10001	000001	A
11110 10001 10001 11110 10001 10001 11110	000010	B
01110 10001 10000 10000 10000 10001 01110	000011	C

Citra karakter yang dibaca sangat mungkin mengandung derau (*noise*) atau tidak lengkap, maka jaringan propagasi balik yang dirancang harus dilatih dengan banyak contoh yang juga mengandung *noise* atau tidak lengkap. Diharapkan jaringan dapat menggeneralisasi sehingga dimasa depan dimasukkan input baru yang berbeda yang citranya juga mengandung *error* maka jaringan tetap mampu mengenalinya sebagai sebuah karakter tertentu.

Pada penelitian ini belum keseluruhan karakter alfanumerik yang dicobakan, hanya terbatas pada karakter ‘A’, ‘B’, dan ‘C’ saja.

2. Mengkonfirmasikan keandalan

   Jumlah set pelatihan sebanyak 10 contoh kasus dan dianggap cukup mampu melatih jaringan untuk dapat mengenali karakter-karakter yang telah dipelajari.

IV. Pendekatan Teori

Jaringan saraf tiruan seperti layaknya otak buatan yang dapat berpikir seperti manusia dalam menyimpulkan sesuatu dari potongan-potongan informasi yang diterima. Komputer diusahakan agar

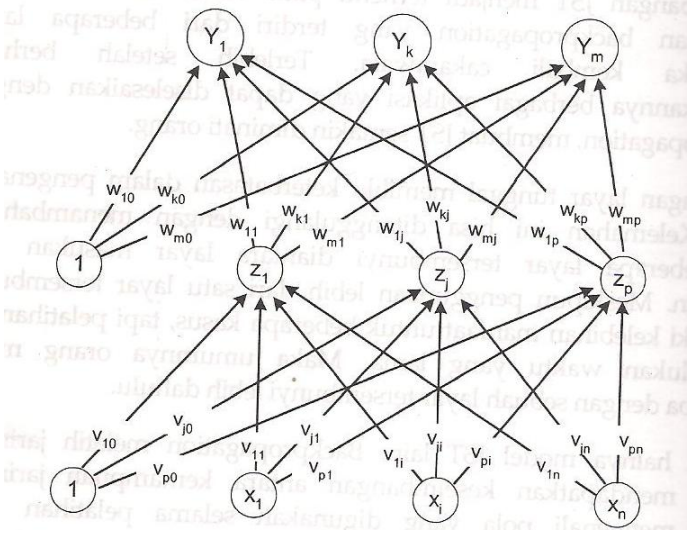
dapat berpikir sama seperti cara berpikir manusia dengan melakukan peniruan terhadap aktivitas-aktivitas yang terjadi di dalam sebuah jaringan saraf biologis.

Jaringan saraf tiruan dengan layar tunggal memiliki keterbatasan dalam pengenalan pola. Kelemahan ini bisa ditanggulangi dengan menambahkan satu/beberapa layar tersembunyi di antara layar masukan dan keluaran. Meskipun penggunaan lebih dari satu layar tersembunyi memiliki kelebihan manfaat untuk beberapa kasus, tetapi pelatihannya memerlukan waktu yang lama. Maka umumnya orang mulai mencoba dengan sebuah layar tersembunyi lebih dahulu.

Seperti halnya model jaringan saraf tiruan lain, Bacpropagation melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola yang digunakan selama pelatihan serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukan yang serupa (tapi tidak sama) dengan pola yang dipakai selama pelatihan.

Arsitektur Backpropagaration

Backpropagaration memiliki beberapa unit yang ada dalam satu atau lebih layar tersembunyi. Gambar di bawah ini adalah arsitektur backpropagaration dengan n buah masukan (ditambah sebuah bias), sebuah layar tersembunyi yang terdiri dari p unit (ditambah sebuah bias), serta m buah unit keluaran. V_{ji} merupakan bobot garis dari unit masukan Xi ke unit layar tersembunyi Zj (Vj0 merupakan bobot garis yang menghubungkan bias di unit layar tersembunyi Zj ke unit keluaran Yk (W_k merupakan bobot bias dari bias di layar tersembunyi ke unit keluaran Z_k).

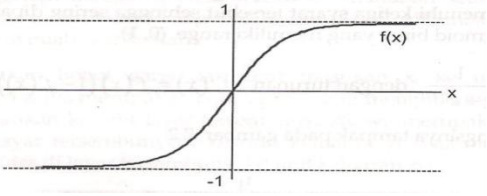


Gambar 2. Arsitektur backpropagation dengan n buah masukan

Dalam backpropagation fungsi aktivasi yang dipakai harus memenuhi beberapa syarat yaitu : kontinu, terdiferensial dengan mudah dan merupakan fungsi yang tidak turun. Bebrapa fungsi aktivasi yang dipakai dalam matlab dalam pelatihan back propagation adalah :

- Tansig (sigmoid bipolar). f(net) = $\frac{2}{1+e^{-net}} - 1$. Dengan turunan f'(x) = $\frac{(1+f(x))(1-f(x))}{2}$.

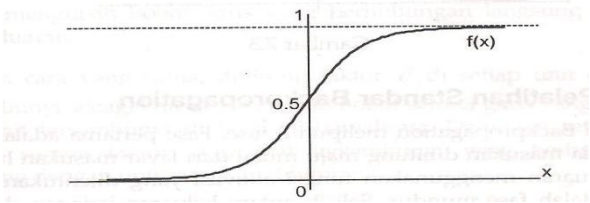
  Fungsi sigmoid bipolar memiliki range [-1,1]. Grafik fungsinya tampak pada gambar 3



Gambar 3. Fungsi aktivasi sigmoid bipolar

- Logsig (sigmoid biner). f(net) = $\frac{1}{1+e^{-net}}$ dengan turunan f'(x) = f(x)(1-f(x)). Fungsi sigmoid biner memiliki bentuk serupa dengan sigmoid bipolar , hanya rangenya adalah [0,1]

  Grafik fungsinya tampak pada gambar 4.



Gambar 4. Fungsi aktivasi sigmoid biner

V. Pembahasan

Pembangunan simulator jaringan saraf tiruan dimulai dengan perancangan arsitektur jaringan yang terdiri atas :

1. Unit input = 36 node
2. Unit tersembunyi = 36 node
3. Unit output = 6 node

Pembentukan jaringan adalah net = newff ([minmax(p)],[7,6]). Dalam pembentukan jaringan matlab akan memberi nilai bobot dan bias awal dengan bilangan acak, akan tetapi pada pelatihan ini memberi nilai bobot dan bias awal dengan nilai tertentu. Pembentukan jaringan adalah net = newff ([minmax(p)],[7,6]). net.IW{1,1}yang berarti unit masukan hanya terhubung dengan layar tersembunyi paling bawah. net.LW{k,j} dipakai untuk menyimpan bobot dari unit di layar tersembunyi ke-j ke unit dilayar tersembunyi ke-k.

Untuk melatih jaringan digunakan perintah train. Pelatihan dilakukan untuk meminimumkan kuadrat kesalahan rata-rata (mse = mean square error). Metode paling sederhana untuk merubah bobot adalah metode penurunan gradien (gradien descent). Bobot dan bias diubah pada arah dimana unjuk kerja fungsi menurun paling cepat , yaitu dalam arah negatif gradiennya. Dari pelatihan pengenalan karakter alfanumerik menggunakan metode propagasi balik didapatkan hasil sebagai berikut :

input

p1 = [0 0 1 0 0 0 1 0 1 0 0 0 0 1 0 1 1 1 1 1 1 0 0 0 1 1 0 0 0 1 0 0 0 0 1]

p2 = [1 1 0 1 0 1 0 0 0 1 1 0 0 0 1 1 1 1 1 1 0 1 0 0 0 1 1 1 1 1 1 0 1 1 1 1 0]

p3 = [0 1 1 1 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 1 1 0 0]...dst

target

t =

0	0	0	0	0	1
0	0	0	0	1	0
0	0	0	0	1	1
0	0	0	0	0	1
0	0	0	0	1	0
0	0	0	0	1	1

Net.IW{1,1}

ans =

39.2000
-39.2000
-39.2000
39.2000
-39.2000
39.2000
39.2000

Net.b{1}

ans =

-39.2000
32.6667
26.1333
-19.6000
13.0667
-6.5333
0

Net.LW{2,1}

ans =

-0.5362	0.6877	-0.3716	-0.0828	-0.5242	-0.7260	-0.3078
-0.5214	-0.6522	-0.2698	0.7397	0.2917	0.6375	-0.6679
-0.9005	-0.6584	-0.2135	0.8685	0.9338	-0.1397	-0.6888
-0.8432	0.9886	0.1831	-0.4711	0.3299	0.7806	-0.6178
0.2816	-0.1204	-0.7605	-0.6794	0.7408	0.4698	-0.1551

-0.6182 -0.3199 -0.9237 0.7457 -0.9801 0.3746 0.7120

Net.b{2}

ans =

-0.0195

0.6319

-0.0785

-0.0853

-0.0986

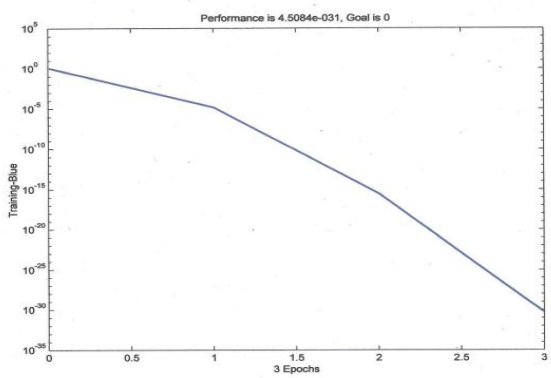
-0.1756

TRAINLM, Epoch 0/100, MSE 0.986253/0, Gradient 678.591/1e-010

TRAINLM, Epoch 3/100, MSE 1.05503e-031/0, Gradient 2.10528e-013/1e-010

TRAINLM, Minimum gradient reached, performance goal was not met.

Training dihentikan pada epoch ke 3/100 meskipun unjuk kerja yang diinginkan (mse = 0) belum tercapai. Pada epoch ke 3 ini mse =1,05503e-031. Selain keterangan tentang perubahan error, matlab juga menampilkan grafik perubahan seperti yang tampak pada gambar 5.



Gambar 5. Grafik perubahan error

VI. Kesimpulan

Jaringan saraf tiruan diuji dengan menggunakan set pelatihan yang digunakan untuk menguji kemampuan memorisasi jaringan dan menguji kemampuan generalisasi jaringan. Dari pengalaman selama pelatihan diharapkan jaringan mampu menggeneralisasikan kasus yang ia hadapi dan kemudian menarik kesimpulan yang cenderung keoutput tertentu. Hasil percobaan memperlihatkan bahwa jaringan saraf tiruan yang dirancang ternyata dapat mengenali dengan baik karakter ‘A’,’B’ dan ‘C’ baik yang mengandung noise ataupun tidak.

VII. Daftar Pustaka

1. Hanselman,D dan Bruce Littlefield., 2000: *Matlab Bahasa Komputasi Teknis*, Penerbit Andi, Yogyakarta.

2. Kusumadewi, S., 2004: *Membangun Jaringan Saraf Tiruan (Menggunakan Matlab & Exel Link)*, Penerbit Graha Ilmu, Yogyakarta.
3. Kusumadewi, S., 2003: *Artificial Intelligence (Teknik dan Aplikasinya)*, Penerbit Graha Ilmu, Yogyakarta.
4. Puspitaningrum, D., 2006: *Pengantar Jaringan Saraf Tiruan*, Penerbit Andi, Yogyakarta.
5. Siang, J., 2004 : *Jaringan Saraf Tiruan & Pemrogramannya Menggunakan Matlab*, Penerbit Andi, Yogyakarta.
6. Turban, E., 1995: *Decision Support and Expert System*, Prentice Hall, NJ.