

Pengenalan Pola Angka Menggunakan Back Propagation Neural Network

Agus Purwo Handoko ⁴⁾

Abstrak

Jaringan perambatan mundur (Back Propagation Neural Network) dilatih dengan metode belajar terbimbing dan Jaringan diberi sekumpulan pola yang terdiri dari pola masukan dan pola yang diinginkan. serta pelatihan dilakukan berulang-ulang sehingga dihasilkan jaringan yang memberikant anggapan yang benar terhadap semua masukannya.

Karena setiap matrik tersebut ada 35 data yang harus di masukkan, maka diperlukan 35 node input. Karena angka yang dimasukkan dari 0 sampai 9 maka hanya ada 10 kemungkinan output, maka digunakan 10 node output pada Jaringan syaraf. JST harus di trining dulu untuk memperoleh output yang sesuai dengan yang diinginkan, sebagai contoh kita memasukkan angka 0 output pertama harus bernilai 1 atau mendekati 1 sedangkan output yang lain harus bernilai 0. Misalkan dimasukkan pola seperti berikut (angka 1) :00100, 01100, 00100, 00100, 00100, 00100,01110 Output kedua harus bernilai 1 dan yang lain harus bernilai 0. Semua node akan diimplementasi sebagai larik bilangan floating point, karena itu input JST mempunyai 35 elemen floating point dan outputnya sebanyak 10 elemen floating point, sedangkan hidden layer memiliki 50 node.

Proses iterasi training sebanyak 1000 kali cukup bisa memeperoleh hasil yang bisa diterima, namun bisa dilakukan lebih dari 1000 untuk menghasilkan bobot yang lebih baik, maksimal iterasi adalah 20000 kali.

Kata kunci: *Back Propagation Neural Network, Matrik, Iterasi*

I. Pendahuluan

Kecerdasan buatan atau Artificial Intellegence (AI) merupakan cabang dari ilmu computer yang mempelajari otomatisasi tingkah laku cerdas. Berikut ini adalah beberapa cabang AI :

⁴⁾ Staf Pengajar STMIK Sinar Nusantara Surakarta

- **Pencarian**. Program AI seringkali harus mengevaluasi kemungkinan yang jumlahnya banyak sekali, misalnya kemungkinan langkah dalam permainan catur atau penyimpulan dari program untuk membuktikan suatu teori.
- **Pengenalan Pola/ Pattern Recognition**
- **Representasi**, yakni bagaimana merepresentasikan/menuliskan fakta-fakta yang ada ke dalam simbol-simbol atau bahasa logika matematis.
- **Inferensi**.
- Pengetahuan dan penalaran yang masuk akal (*common sense knowledge and reasoning*).
- **Perencanaan**. Program perencanaan bermula dari fakta-fakta umum (terutama fakta mengenai efek dari suatu aksi), fakta tentang situasi yang khusus, dan suatu pernyataan tentang tujuan. Dari sini kemudian dibuat sebuah strategi untuk mencapai tujuan tersebut. Secara umum, biasanya strategi tersebut berupa urutan-urutan aksi.
- **Epistemologi**, yakni studi tentang sumber, sifat, dan keterbatasan pengetahuan yang digunakan untuk pemecahan masalah.

II. Pattern Recognition

Pola adalah entitas yang terdefiniskan dan dapat diidentifikasi melalui cirri-cirinya (features). Cirri – cirri tersebut digunakan untuk membedakan suatu pola dengan pola lainnya. Ciri yang bagus adalah yang mempunyai daya pembeda yang tinggi, sehingga pengelompokan pola berdasarkan cirri yang dimiliki dapat dilakukan dengan keakuratan yang tinggi.

Sebagai contoh :

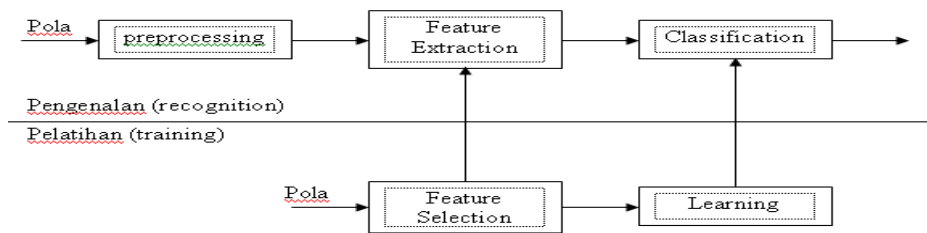
Pola	Ciri
Huruf	tinggi, tebal, titik sudut, lengkungan garis,
Suara	Amplitudo frekuensi, nada, intonasi, warna,
Tanda tangan	panjang, kerumitan, tekanan, dll.
Sidik jari	lengkungan, jumlah garis, dll

Terdapat dua pendekatan yang dilakukan dalam pengenalan pola: pendekatan secara statistik dan pendekatan secara sintatik atau structural.

Pengenalan pola secara Statistik.

Pendekatan ini menggunakan teori-teori ilmu peluang. Cirri-ciri yang dimiliki oleh suatu pola ditentukan distribusi statistiknya. Pola yang

berbeda memiliki distribusi yang berbeda pula. Sistem pengenalan pola dengan pendekatan statistic ditunjukkan oleh diagram :



Gambar 1. Sistem pengenalan pola dengan pendekatan statistik

Ada dua fase dalam sistem pengenalan pola, yaitu fase pelatihan dan fase pengenalan. Pada fase pelatihan dipelajari untuk menentukan cirri yang akan digunakan dalam proses pengenalan serta prosedur klasifikasinya. Pada fase pengenalan, citra diambil cirinya kemudian ditentukan kelas kelompoknya.

1. Preprocessing.

Proses awal yang dilakukan untuk memperbaiki kualitas image dengan menggunakan terknik teknik pengolahan berdasarkan ciri dari pola tersebut.

2. Feature Extraction.

Proses pengambilan cirri-ciri yang terdapat pada obyek di dalam image. Pada proses ini obyek di dalam image mungkin perlu dideteksi seluruh tepinya, lalu menghitung property-property obyek yang berkaitan sebagai cirri. Beberapa proses ekstraksi ciri mungkin perlu mengubah image masukan sebagai citra biner, melakukan penapisan pola, dan sebagainya.

3. Clasification.

Proses mengelompokkan obyek ke dalam kelas yang sesuai.

4. Feture Extraction.

Proses memilih ciri pada suatu obyek agar diperoleh ciri yang optimum, yaitu ciri yang dapat digunakan untuk membedakan suatu obyek dengan obyek lainnya.

5. Learning.

Proses belajar membuat aturan klasifikasi sehingga jumlah kelas yang tumpang tindih dibuat sekecil mungkin.

III. Jaringan syaraf tiruan

Jaringan syaraf tiruan didefinisikan sebagai suatu system pemrosesan informasi yang mempunyai karakteristik menyerupai jaringan syaraf manusia

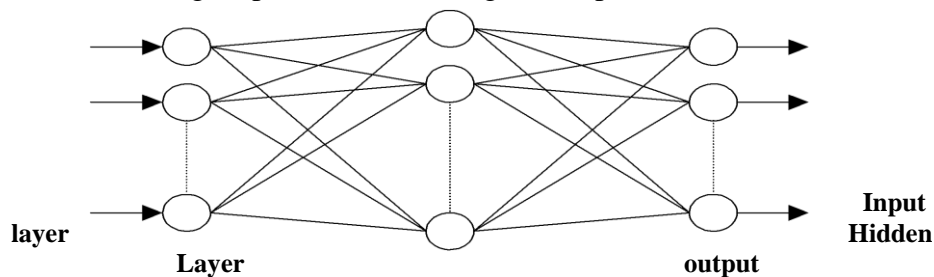
Jaringan saraf tiruan memiliki sejumlah kelebihan dibanding dengan metode perhirungan lainnya, yaitu :

1. kemampuan mangkuisisi pengetahuan walaupun dalam kondisi ada gangguan dan ketidakpastian. Hal ini karena jaringan saraf tiruan mampu melakukan generalissi, abstraksi, dan ekstraksi terhadap properti statitik dari data.
2. kemampuan merepresentasikan pengetahuan secara fleksibel. Jaringan saraf tiruan dapat menciptakan sendiri representasi melalui pengaturan diri sendiri atau kemampuan belajar.
3. keampuan untuk memberikan toleransi atas uatu distorsi (eror/fault), dimana gangguan kecil pada data dapat dianggap hanya sebagai noise belaka.
4. kemampuan memproses pengetahuan secara efisien karena memakai sistem pararel, sehingga waktu yang diperlukan untuk mengoperasikannya menjadi lebih singkat.

Walaupun memiliki kelebihan, jaringan saraf tiruan juga mempunyai sejumlah keterbatasan, antara lain kekurangmmpuannya dalam melakukan operasi-operasi numerik dengan presisi tinggi, operasi algoritma aritmatik, operasi logika, dan operasi simbolis serta lamanya proses pelatihan yang terkadang membutuhkan waktu berhari-hari untuk jumlah data yang besar.

IV. Backpropagation Neural Network

Jaringan backpropagation terdiri atas tiga lapisan atau lebih unit pengolahan. Gambar 2 menunjukkan jaringan backpropagation dengan tiga lapisan pengolahan, bagian kiri sebagai masukan, bagian tengah disebut sebagai lapisan tersembunyi dan bagian kanan disebut lapisan keluaran. Ketiga lapisan ini terhubung secara penuh.



Gambar 2. Struktur Jaringan Syaraf

Untuk menghitung nilai penjumlahan berbobot digunakan rumus :

$$S_j = \sum_{i=0}^n a_i \cdot w_{ji}$$

Dengan :

a_i = masukan yang berasal unit i

w_{ji} = bobot sambungan dari unit i ke unit j

setelah nilai S_j dihitung, fungsi sigmoid diterapkan pada S_j untuk membentuk $f(S_j)$. Fungsi sigmoid ini mempunyai persamaan :

$$f(S_j) = \frac{1}{1 + e^{-S_j}}$$

Hasil perhitungan $f(S_j)$ ini merupakan nilai aktivasi pada unit pemroses j. Nilai ini dikirimkan ke seluruh keluaran unit j. Setelah perambatan maju dikerjakan maka jaringan siap melakukan perambatan mundur.

Jika j adalah salah satu unit pada lapisan keluaran maka galat lapisan keluaran dapat dihitung dengan rumus :

$$\delta_j = (t_j - a_j) f'(S_j)$$

Dengan :

t_j = keluaran yang diinginkan dari unit.

a_j = keluaran dari unit j

$f'(S_j)$ = turunan dari fungsi sigmoid

S_j = hasil penjumlahan berbobot.

Jika j adalah suatu lapisan tersembunyi, maka galat lapisan tersembunyi dapat dihitung dengan menggunakan rumus :

$$\delta_j = \sum_k \delta_k \cdot w_{kj} \cdot f'(S_j)$$

$$\Delta w_{ji} = \eta \cdot \delta_j \cdot a_i$$

Dengan :

Δw_{ji} = perubahan bobot dari unit i ke unit j

η = laju belajar/konvergensi

δ_j = galat lapisan tersembunyi

A_i = masukan yang berasal dari unit i

Variabel η menyatakan suatu konstanta belajar yang berharga antara 0,25 – 0,75. Nilai ini menunjukkan kecepatan belajar dari jaringan. Nilai yang terlalu tinggi dapat menyebabkan jaringan menjadi tidak stabil

sedangkan nilai yang terlalu kecil dapat menjadikan waktu belajar yang lama. Oleh karena itu pemilihan nilai η harus seoptimal mungkin agar didapatkan proses belajar yang cepat.

Algoritma selengkapnya jaringan backpropagation adalah sebagai berikut :

- ❖ Langkah 0 : Inisialisasi bobot-bobot (tetapkan dalam nilai acak kecil).
- ❖ Langkah 1 : Bila syarat berhenti adalah salah, kerjakan langkah 2 sampai 9.
- ❖ Langkah 2 : Untuk setiap pasangan pelatihan, kerjakan langkah 3 – 8.

Umpan maju

- ❖ Langkah 3 : Tiap unit masukan (x_i , $i = 1, \dots, n$) menerima isyarat masukan x_i dan diteruskan ke unit-unit tersembunyi.
- ❖ Langkah 4 : Tiap unit tersembunyi (z_j , $j = 1, \dots, p$) menjumlahkan isyarat masukan terbobot,

$$z_in_{jk} = v_{0j} + \sum x_i v_{ij}$$

Dengan menerapkan fungsi aktivasi hitung :

$$Z_j = f(z_in_j)$$

Dan kirim isyarat ini ke unit-unit keluaran.

- ❖ Langkah 5 : Tiap unit keluaran (y_k , $k = 1, \dots, m$) menjumlahkan isyarat masukan

berbobot,

$$y_in_k = w_{0k} + \sum_{j=1}^p z_j w_{jk}$$

Dengan menerapkan fungsi aktivasi hitung,

$$y_j = f(y_in_k)$$

Perambatan balik galat :

- ❖ Langkah 6 : Tiap unit keluaran (y_k , $k = 1, \dots, m$) menerima pola sasaran berkaitan

dengan pola pelatihan masukannya. Hitung galat informasi:

$$\delta_k = (t_k - y_k) f'(y_in_k)$$

Hitung koreksi bobot dan prasikapnya:

$$\Delta w_{jk} = \alpha \delta_k x_j$$

$$\Delta w_{0k} = \alpha \delta_k$$

- ❖ Langkah 7 : tiap unit tersembunyi ($z_j, j = 1, \dots, p$) menjumlahkan delta masukannya (dari unit-unit di lisan atasnya).

$$\delta_{in_j} = w_{0k} + \sum_{k=1}^m \delta_j w_{jk}$$

Hitung galat informasinya:

$$\delta_j = \delta_{in_j} f'(x_{in_j})$$

Perbaharui bobot dan prasikap:

- ❖ Langkah 8 : tiap keluaran ($y_k, k=1, \dots, m$) memperbaharui bobot-bobot dan

prasikapnya ($j=0, 1, \dots, p$)

$$w_{jk} \text{ (baru)} = w_{jk} \text{ (lama)} + \Delta w_{jk}$$

tiap unit tersembunyi ($z_j, j=1, \dots, n$);

$$v_{ij} \text{ (baru)} = v_{ij} \text{ (lama)} + \Delta v_{ij}$$

- ❖ Langkah 9 : Uji syarat berhenti.

Prosedur pembaruan bobot-bobot dapat dimodifikasikan dengan menggunakan momentum k dalam rumus pembaruan bobot, biasanya konvergensi akan lebih cepat dicapai. Dalam pembaruan bobot menggunakan momentum, nilai bobot pada iterasi ke $(t+1)$ ditentukan oleh bobot pada iterasi ke- t dan ke- $(t-1)$.

V. Pembahasan Aplikasi *Pattern Recognition* untuk Pengenalan Pola Angka menggunakan *Backpropagation NN*

Dalam aplikasi ini, JST digunakan untuk mengenali pola angka 0 sampai 9 yang diwakili oleh sebuah matrik berukuran 5×7 . Matrik tersebut berupa array bilangan 0 dan 1 sejumlah 35, 1 menyatakan ada dan 0 tidak ada.

Contoh angka 0, diwakili oleh :

```
01110
10001
10001
10001
10001
10001
10001
10001
01110
```

Larik bilangan adalah: 0,1,1,1,0,1,0,0,1,.....

Karena setiap matrik tersebut ada 35 data yang harus di masukkan, maka diperlukan 35 *node* input. Karena angka yang dimasukkan dari 0 sampai 9 maka hanya ada 10 kemungkinan output, maka digunakan 10 *node* output pada Jaringan syaraf. JST harus di trining dulu untuk memperoleh output yang sesuai dengan yang diinginkan, sebagai contoh kita memasukkan angka 0 output pertama harus bernilai 1 atau mendekati 1 sedangkan output yang lain harus bernilai 0. Misalkan dimasukkan pola seperti berikut (angka 1) :

```
00100
01100
00100
00100
00100
00100
00100
01110
```

Output kedua harus bernilai 1 dan yang lain harus bernilai 0.

Semua *node* akan diimplementasi sebagai larik bilangan *floating point*, karena itu input JST mempunyai 35 elemen *floating point* dan outputnya sebanyak 10 elemen *floating point*, sedangkan hidden layer memiliki 50 *node*.

Proses Training JST

JST yang dibuat memiliki 35 input, 50 hidden layer dan 10 output. Proses training dilakukan melalui 4 fase, seperti berikut ini :

Fase 1 :

Untuk setiap *node* pada layer tersembunyi, menghitung bobot dari semua *node* input yang ada. Aplikasikan fungsi aktivasi (fungsi sigmoid) untuk menghasilkan output dari setiap hidden layer.

Fase 2 :

Hal yang sama dilakukan seperti pada fase1 untuk *node* output. Yaitu mencari bobot tiap output yang terhubung ke hidden layer dan aplikasikan fungsi aktivasi. Untuk setiap output dicari nilai selisih / delta (δ).

Delta (δ) = Output*(1-Output)*(Target-output)

Fase 3 (Back Propagation):

Jumlah dari semua hasil kali tersebut disebut ‘delta (δ) total’ untuk hidden layer. Kemudian dihitung $M\delta$ (delta untuk layer tengah/hidden), dengan rumus :

$$M\delta = \text{output_layer tengah} * (1 - \text{output_layer tengah}) * \delta_{\text{total}}$$

Fase 4:

Fase terakhir adalah menyesuaikan bobot antara layer input dan layer tengah dengan menambahkan: **(Bobot tiap node + Mδ*input).**

Fungsi Sigmoid :

- a. Fungsi Logsigmoid (binary sigmoid) yang didefinisikan sebagai

$$F(x) = \frac{1}{1 + e^{-1*value}}$$

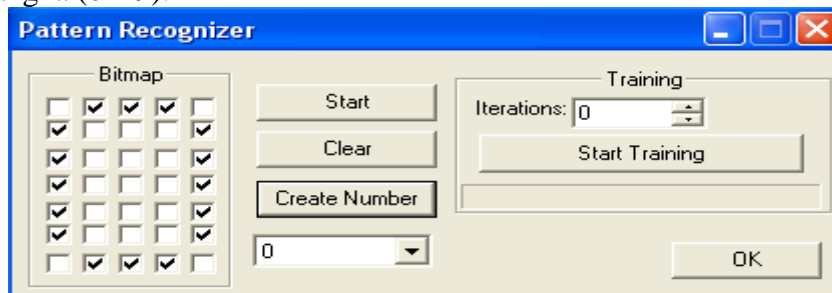
- b. Fungsi Tansigmoid (bipolar sigmoid) yang didefinisikan sebagai

$$F(x) = \frac{1}{1 + e^{-1*value}} - 1$$

Proses iterasi training sebanyak 1000 kali cukup bisa memperoleh hasil yang bisa diterima, namun bisa dilakukan lebih dari 1000 untuk menghasilkan bobot yang lebih baik, maksimal iterasi adalah 20000 kali.

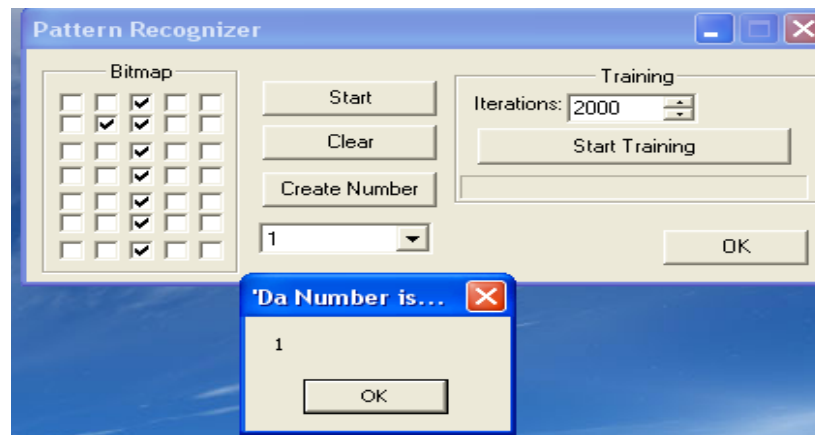
Hasil UjiCoba Program :

Tampilan window pattern recognition adalah seperti berikut ini, dengan memasukkan tanda cek (√) pada matrik/ kotak 5 x 7 yang tersedia. Pola yang terbentuk pada matrik akan dikenali oleh program sebagai angka (0 – 9).



Gambar 3. Tampilan window awal

Sebelum program mengenali pola, harus dilakukan training terlebih dahulu dengan menekan tombol ‘start training’ dan sebelumnya memasukkan banyaknya proses iterasi. Untuk menggunakan program tanda cek dimasukkan ke matrik, yang membentuk pola angka tertentu kemudian tekan tombol ‘start’, program akan melakukan proses *recognition*. Berikut ini contoh hasil pengenalan pola angka 1 :



Gambar 3. Hasil pengenalan pola

VI. KESIMPULAN

Kesimpulan yang dapat diambil dari percobaan adalah dengan menggunakan input layer 35 dan 50 hidden layer dan 10 output pada NN yang dibuat, cukup mampu untuk melakukan pengenalan pola angka yang diwakili oleh matrik 5 x 7.

F. Referensi

- Harada. S., Itoh. Y, dan Nakatani. H.,1997, "Interactive image retrieval by natural language", *Journal of Optical Engineering*, vol.36, no.12.
- Hartati, S.,1990., "Orientation Adaptive Quadrees Representation", *Tesis Master, Fakultas Ilmu Komputer, University of New Brunswick, Kanada*.
- Hermawan, A., 2004, *Jaringan Syaraf Tiruan: Teori dan Aplikasi*, Penerbit Andi, Yogyakarta
- Pandya, AS, 1995, "*Pattern Recognition with Neural Networks in C++*, CRC Press.
- Siang, J.J., 2005, *Jaringan Syaraf Tiruan & Pemrogramannya Menggunakan Matlab*, Penerbit Andi, Yogyakarta.