PENDEKATAN *ARTIFICIAL NEURAL NETWORK (ANN)* UNTUK MENGKLASIFIKASIKAN HEWAN VETEBRATA MENGGUNAKAN *KOHONEN SELF ORGANIZING MAP (SOM)*

[1]Tursina

[1]Prodi. Teknik Informatika, Fakultas Teknik, Universitas Tanjungpura
Jl. Akhmad Yani Pontianak 78124 Telp. (0561) 740186
E-mail: tursina15@yahoo.com

Abstract

This research was proposes an approach to identifying vertebrates or invertebrates that premises Kohonen Self Organizing Map (SOM). SOM is one of the topology of the Artificial Neural Network (ANN).The network will be trained to classify five kinds of animals that will represent each class. Modeling requires 6 input neurons to input the characteristics of animals and 5 output neurons (according to the number of classes specified
Key Word : Kohonen Self Organizing Map (SOM), Artificial Neural Network (ANN)

Abstrak

Penelitian ini mengusulkan salah satu pendekatan dalam mengidentifikasi hewan bertulangbelakang atau tidak bertulangbelakang yaitu dengan Kohonen Self Organizing Map(SOM). SOM merupakan salah satu bentuk topologi dari Artificial Neural Network (ANN). Jaringan akan dilatih untuk mengelompokkan 5 macam binatang yang akan mewakili masing-masing kelas. Pemodelan memerlukan 6 neuron input untuk menginputkan ciri-ciri binatang dan 5 neuron output (sesuai dengan jumlah kelas yang ditentukan)

Kata Kunci : Kohonen Self Organizing Map (SOM), Artificial Neural Network (ANN)

I. PENDAHULUAN

Alam semesta mempunyai banyak sekali jenis-jenis hewan, sehingga sangat sulit untuk mengenali dan mempelajarinya. Untuk mengenali hewan dibutuhkan pengklasifikasian hewan tersebut. Klasifikasi hewan adalah pengelompokan berdasarkan sesamaan bentuk dan fungsi pada tubuh hewan. Klasifikasi hewan bersifat dinamis. Hal ini desebabkan beberapa kemungkinan seperti adanya perkembangan pengetahuan tentang hewan, penggunaan karakter yang berbeda dalam klasifikasi. Klasifikasi hewan berdasarkan atas perbedaan dan persamaan karakter tertentu pada hewan yang bersangkutan. Dalam penenlitian ini dilakukan pengklasifikasian hewan yang bertulang belakang atau tidak bertulangbelakang dengan *menggunakan*

Kohonen Self Organizing Map(SOM) yang merupakan salah satu bentuk topologi dari *Artificial Neural Network* (ANN).

Kohonen Self Organizing Map (SOM) merupakan salah satu bentuk topologi dari *Artificial Neural Network* (ANN) yang diperkenalkan oleh Teuvo Kohonen pada tahun pada tahun 1982[1]. Kohonen SOM telah banyak dimanfaatkan untuk pengenalan pola baik berupa citra maupun suara. Jaringan ini mampu mereduksi dimensi *input* pola ke jumlah yang lebih sedikit sehingga pemrosesan komputer menjadi lebih hemat[2].

Kohonen SOM merupakan *unsupervised* ANN yaitu ANN yang dalam proses pelatihannya tidak memerlukan pengawasan berupa target *output*. Metode pembelajaran dilakukan dengan meng-*update* bobot berdasarkan sinyal yang diberikan ke *neuron input* [3].

Penelitian ini diharapkan dapat memudahkan pengenalan hewan yang bertulang belakang dengan hewan yang tidak bertulang belakang**.**

II. METODE PENELITIAN

2.1 Analisis dan Perancangan Sistem

2.1.1 Pengelompokan hewan

Kohonen SOM akan digunakan untuk mengelompokkan hewan bertulang belakang. Setelah Kohonen SOM selesai dilatih, maka dengan memasukkan ciri dari seekor hewan maka Kohonen SOM akan dapat memberitahukan mahkluk tersebut termasuk kelompok yang mana.

Ada 6 pertanyaan yang digunakan untuk identifikasi yaitu:

1. Apakah hewan tsb beranak?
2. Apakah hewan tsb hidup di darat?
3. Apakah hewan tsb hidup di air?
4. Apakah hewan tsb bersisik?
5. Apakah hewan tsb bersayap?
6. Apakah hewan tsb berkaki?

Jawaban dari 6 pertanyaan diatas digunakan sebagai ciri-ciri suatu hewan dan selanjutnya berdasarkan ciri-ciri itulah dilakukan pengelompokkan menjadi 5 kelompok yaitu:

1. Kelompok Mamalia
2. Kelompok Reptilia
3. Kelompok Unggas
4. Kelompok Amfibi
5. Kolompok Ikan

2.2.2 Pemodelan

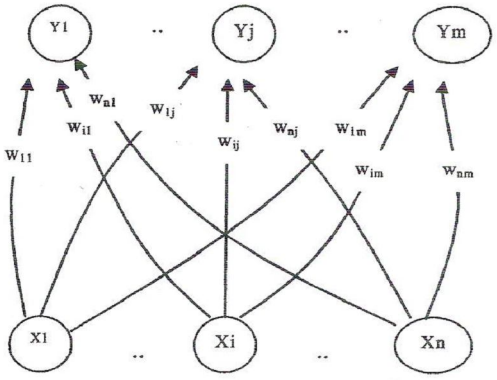
Ciri-ciri mahkluk hidup tersebut direpresentasikan dengan fungsi binari sebagai berikut:

- Setiap ciri yang benar akan direpresentasikan dengan 1
- Setiap ciri yang salah akan direpresentasikan dengan 0

Sebagai contoh : Arwana akan memiliki ciri 001100 karena Arwana tidak

beranak, tidak hidup di darat, hidup di air, bersisik, tidak bersayap dan tidak berkaki.

Pemodelan yang digunakan adalah sistem dengan 6 *neuron input* dan 5 *neuron output*. Memerlukan 6 *neuron input* untuk menginputkan ciri-ciri mahkluk hidup dan memerlukan 5 *neuron output* sesuai dengan jumlah kelompok yang diharapkan akan terbentuk. Arsitektur jaringan dapat dilihat pada Gambar 1. (ket. m= 5; n= 6)



Gambar 1. Arsitektur jaringan

Jaringan akan dilatih untuk mengelompokkan 5 macam binatang yang akan mewakili masing-masing kelas.

Tabel 1. Representasi ciri-ciri hewan

Nama Hewan	Ciri – ciri					
	Beranak	Hidup di Darat	Hidup di Air	Bersisik	Bersayap	Berkaki
Arwana	0	0	1	1	0	0
Buaya	0	1	1	0	0	1
Merpati	0	1	0	0	1	1
Cicak	0	1	0	1	0	1
Kuda Zebra	1	1	0	0	0	1

Tabel 1 merupakan representasi dari data ciri-ciri hewan yang akan dikelompokkan. Proses pelatihan akan dilakukan sebanyak 20 kali iterasi. Nilai awal dari bobot koneksi menggunakan bilangan real acak antara 0 dan 1. Urutan vektor *input* setiap iterasi adalah arwana – buaya – merpati – cicak – zebra.

Nilai $\gamma(0)$ = 2, T = 20, $\alpha(t)$ = 4, dengan pertimbangan posisi kelompok yang terbentuk tidak memiliki hubungan sama sekali (mamalia tidak berhubungan dengan ikan, maupun kelas yang lain)

2.2.3 Pelatihan Jaringan

Bagian pertama dari pelatihan adalah memasukkan *data training* seperti yang terlihat pada Tabel 3. Bagian kedua adalah menentukan bobot awal w_{ij}. Nilai yang digunakan adalah bilangan real acak dari 0 sampai 1. Jumlah bobot yang di bangkitkan sebanyak cacah *neuron input* x cacah *neuron output* = 6 x 5 = 30. Pembangkitan bobot dapat dilihat pada Tabel 2.

Tabel 2. Pembangkit bobot.

W	1	2	3	4	5
1	0,260000	0,940000	0,900000	0,430000	0,550000
2	0,090000	0,020000	0,750000	0,040000	0,390000
3	0,200000	0,540000	0,860000	0,410000	0,180000
4	0,750000	0,180000	0,450000	0,600000	0,150000
5	0,660000	0,060000	0,580000	0,450000	0,820000
6	0,220000	0,310000	0,380000	0,680000	0,550000

Misalnya:

W_{13} = **0,200000**

Dalam iterasi pertama melibatkan data *training* Arwana yaitu:

$X_1 = 0$

$X_2 = 0$

$X_3 = 1$

$X_4 = 1$

$X_5 = 0$

$X_6 = 0$

Data tersebut akan dibandingkan dengan kelima *output* yang ada, dan ditentukan yang paling dekat sebagai *best matching* dengan rumus:

$$d_j = \sum_{i=1}^{n} (W_{ij} - X_i)^2$$

Bobot yang diubah adalah bobot yang berhubungan dengan output ke 5 maka yang diubah adalah $W_{11}, W_{12}, W_{13}, W_{14}, W_{15}$

Contoh:

Jarak data arwana ke setiap *output* misalnya diperoleh:

$d_1 = 2.1113$ $d_2 = 2.1643$ $d_3 = 2.5553$ $d_4 = 1.1442$ $d_5 = 1.1013$

Dapat dilihat jarak yang terdekat adalah 1.1013 yaitu *output* ke 5. Maka *best matching unit* adalah *output neuron* ke 5. Selanjutnya nilai bobot yang berhubungan dengan neuron tsb akan di-*update*. Nilai perubahan didapat dengan rumus. [4]

$$W_{ij}(t+1) = W_{ij}(t) + \gamma(t)\, h_{ik}(t)\, (x_i(t) - W_{ij}(t))$$

Misalnya didapat nilai perubahan bobot (delta/del):

W_{11} = 0,550000 + (- 0,519896) = 0,030104

dst sampai W_{15} . Contoh *Update* data Arwana dapat dilihat pada Tabel 3.

Tabel 3. *Update* data Arwana

w(i,j)	1	2	3	4	5
1	0,2600	0,9400	0,9000	0,4300	0,0301
2	0,0900	0,0200	0,7500	0,0400	0,2209
3	0,2000	0,5400	0,8600	0,4100	1,6827
4	0,7500	0,1800	0,4500	0,6000	0,6499
5	0,6600	0,0600	0,5800	0,4500	-0,4197
6	0,2200	0,3100	0,3800	0,6800	0,2073

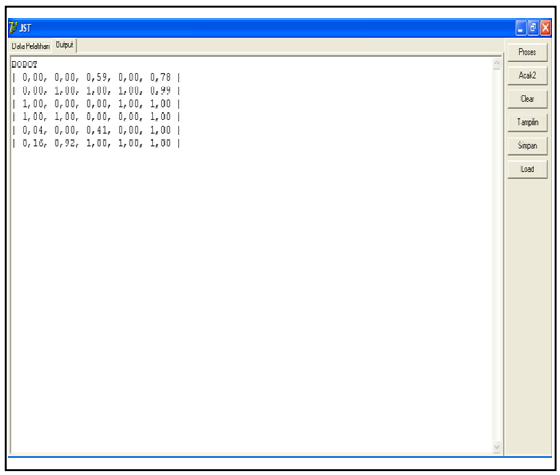
Proses selanjutnya berulang untuk data : buaya, merpati, cicak, zebra.

Setelah data zebra masuk, proses iterasi pertama selesai.

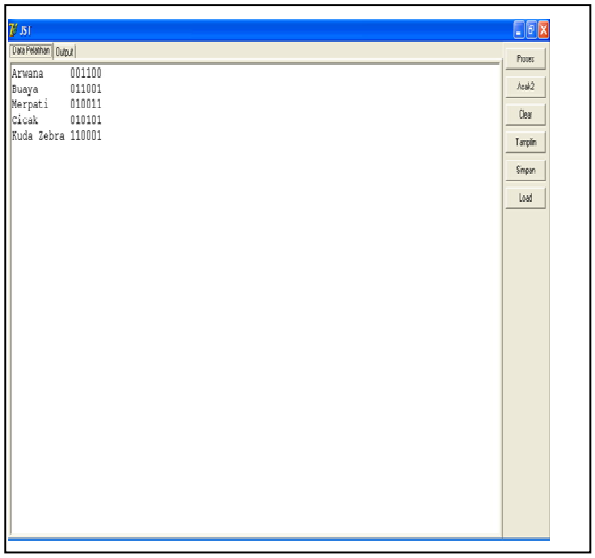
Proses diulang sampai selesai 20 iterasi.

III. HASIL DAN PEMBAHASAN

Untuk memberikan gambaran yang lebih jelas mengenai contoh pelatihan pada jaringan Kohonen SOM maka dilakukan implementasi ke dalam sebuah program. *Input* Data Training dapat dilihat pada Gambar 2. sedangkan *generate* bobot awal terlihat pada Gambar 3.



Gambar 2. *Input* Data *Training*



Gambar 3. *Generate* Bobot Awal

Bagian utama dari program adalah sebagai berikut:

```
for t := 1 to IterasiTotal do
begin
  iterasi := t;
  for i := 0 to Memo1.Lines.Count - 1 do
ProsesInput(i);
end;
```

IterasiTotal merupakan konstanta yang sudah ditentukan dengan nilai tertentu. Di dalam program ini ada tiga

konstanta yang ditentukan dari awal. Langkah-langkah dalan pelatihan sebagai berikut:

- Meng-*input*-kan data pelatihan

Disetiap iterasi, dilakukan proses pembelajaran untuk setiap *input* sebanyak n baris yang ada dalam data pelatihan. Berikut ini adalah data *input* untuk pelatihan:

1. Arwana : 001100
2. Buaya : 011001
3. Merpati : 010011
4. Cicak : 010101
5. Zebra : 110001

- Proses *Parsing*.

Setiap melakukan proses terhadap satu baris data *input*, program melakukan *parsing* dengan memisahkan nama dan data *input* berupa angka enam digit yang berisi 0 atau 1. Kemudian data *input* tersebut diubah ke dalam array yang berisi enam elemen yaitu nilai integer. Dalam program akan dipisahkan antara nama hewan dengan data *input* (yang berupa karakteristik hewan tersebut). Nama dimasukkan ke dalam variabel *nama* sementara data *input* dimasukkan ke dalam variabel *input*. Proses berikutnya adalah memasukkan data di variabel *input* ke dalam Array. Array *input* ada di variabel global *_I* sehingga data string *s* diubah menjadi angka-angka dan dimasukkan ke dalam *_I*. Jika karakter untuk indek ke i di dalam string *s* adalah '1' maka variabel *_I[i-1]* = 1 selainnya itu variabel *_I[i-1]* dianggap 0.

- Proses mencari nilai dari node *output* (lima node dari indek 0 hingga indek 4).

Kode program untuk mendapatkan *output* adalah sebagai berikut:

```
for j := 0 to 4 do
  _O[j] := getOutput(j);
```

Seluruh hasil dari output dengan menggunakan fungsi getOutput(j) dimana j adalah indek node output disimpan ke dalam data array *output* yaitu variabel _O sehingga _O[j] = nilai output node ke j. getOutput sendiri adalah fungsi yang dibuat untuk mendapatkan nilai output dengan parameter j yang berupa indek dari node output.

- Proses mencari nilai indek *output* yang nilainya paling kecil.

Berikut ini adalah kode program untuk mencari indek *output* yang memiliki nilai paling kecil.

```
o := 0;
for j := 1 to 4 do if (_O[j] < _O[o]) then
o := j;
```

```
indexMinimal := o;
```

- Proses perbaikan bobot.

Proses ini hanya dilakukan terhadap bobot-bobot yang terkait langsung dengan output _O[o] atau node *output* yang memiliki nilai paling kecil.

Proses akan berlangsung untuk setiap data *input* sebanyak **iterasiTotal**. Pada kasus data input ini, hasil akhirnya berupa data bobot yang tidak berubah lagi yaitu bisa ditunjukkan dengan Tabel 4.

Tabel.4 Data Bobot

```
    0     1     2     3     4
0 | 0,00, 0,00, 0,59, 0,00, 0,78 |
1 | 0,00, 1,00, 1,00, 1,00, 0,99 |
2 | 1,00, 0,00, 0,00, 1,00, 1,00 |
3 | 1,00, 1,00, 0,00, 0,00, 1,00 |
4 | 0,02, 0,00, 0,41, 0,00, 1,00 |
5 | 0,15, 0,94, 1,00, 1,00, 1,00 |
```

Pada Tabel 4 bagian vertikal mewakili *input* sementara horizontal mewakili *output*. Sementara klasifikasi yang diperoleh terhadap *input* yang ada dapat dilihat pada Tabel 5.

Tabel 5 Klasisfikasi sementara

```
Arwana => 0
Buaya => 3
Merpati => 2
Cicak => 1
Kuda Zebra => 2Gambar 4.3 Hasil
Klasifikasi Setelah 20 Kali Iterasi
```

II. Kesimpulan

- Kohonen SOM dapat digunakan untuk klasifikasi suatu objek yang tidak memiliki aturan klasifikasi yang pasti.
- Ada ketidak sesuaian dengan kondisi *real* yaitu a merpati berada pada satu kelompok dengan zebra. Hal ini menunjukkan bahwa Kohonen SOM tidak tepat digunakan untuk klasifikasi objek yang memiliki aturan pengelompokkan.
- Kohonen SOM mengelompokkan data-data yang ada kedalam kelompok-kelompok, bukan mengelompokkan sesuai dengan kelas yang ditentukan sebelumnya, walaupun maksimum kelompok tetap ditentukan

6. Referensi

[1] Demuth, Beale, *Neural Network Toolbos For Use with Matlab,* The Mathworks Inc., Massachusetts, 1995

[2] Kusumadewi, S., *Artificial Intelligence* (*Teknik dan Aplikasinya*), Graha Ilmu, Yogyakarta, 2002

[3] Hanselman, Littlefield, *MATLAB: Bahasa Komputasi Teknis*, Edisi Pertama, Andi and Pearson Education Asia Pte. Ltd., Yogyakarta, 2001

[4]Siang, Jong Jek. *Jaringan Syaraf Tiruan & Matlab*. Yogyakarta: Andi Offset, 2005