

MODIFIKASI ALGORITMA PROPAGASI BALIK UNTUK PENGENALAN DATA IRIS DAN DATA FERET

riwinoto¹, wahyudi², tony³, reinhoran⁴, kusumoputro⁵

1 Politeknik Batam

Jurusan Teknik Informatika

E-mail: riwi@polibatam.ac.id

2. Lembaga Penerbangan dan Antariksa (LAPAN)

Unit Komputasi, Pusat Teknologi Dirgantara Terapan, Jakarta

Email: youdai@lapan.go.id

3. Universitas Tarumanegara, Fakultas Teknologi Informasi, Jakarta

Email: tony.b.fti.untar@gmail.com

4. STMIK Nusa Mandiri, Magister Ilmu Komputer, Jakarta

Email: rbahrns@gmail.com

5. Universitas Indonesia, Fakultas Teknik, Departemen Teknik Elektro

Jakarta

Email: kusumo@eng.ui.ac.id

Abstrak

Variasi dari algoritma propagasi balik konvensional pada pelatihan jaringan saraf tiruan telah diajukan dengan menggunakan fungsi kesalahan kuadratis. Algoritma ini seperti algoritma successive overrelaxation (SOR) pada sistem persamaan linear untuk mencari nilai bobot terakhir yang digunakan untuk memperbaharui bobot pada setiap busur. Dari analisis terhadap percobaan terhadap data feret dan iris sebagai pembanding, jumlah epoch yang diperlukan untuk mencapai konvergensi dengan menggunakan BP SOR 0.9 kali lebih kecil dibandingkan dengan menggunakan BP standar. Hal ini berarti hasil yang didapatkan masih lebih besar dari threshold yang diharapkan yaitu dibawah 0.67 untuk klaim bahwa BP SOR menghasilkan waktu komputasi yang lebih cepat daripada BP. Hasil rate recognition dengan menggunakan BP SOR relatif sama dibandingkan dengan menggunakan BP standar.

Kata kunci : *Propagasi balik, Successive Overrelaxation, Iris, Feret*

1 Pendahuluan

Jaringan saraf tiruan (JST) merupakan struktur pemroses informasi yang terdistribusi dan bekerja secara paralel yang terdiri atas elemen pemroses (yang memiliki memori lokal dan beroperasi dengan informasi lokal) yang diinterkoneksi bersama dengan alur sinyal searah yang disebut koneksi. Setiap elemen pemroses memiliki koneksi keluaran tunggal yang bercabang (*fan out*) ke sejumlah koneksi kolateral yang diinginkan (setiap koneksi membawa sinyal yang sama dari keluaran elemen pemroses tersebut). Keluaran pemroses tersebut dapat merupakan sembarang jenis persamaan matematis yang diinginkan. Seluruh proses yang berlangsung pada setiap elemen pemroses harus benar-benar dilakukan secara lokal yaitu keluaran hanya bergantung pada nilai masukan pada saat itu yang diperoleh melalui koneksi dan nilai yang tersimpan dalam memori lokal (Robert 1988). JST merupakan sistem adaptif yang dapat merubah strukturnya untuk memecahkan masalah berdasarkan informasi eksternal maupun internal yang mengalir melalui jaringan tersebut. JST digunakan untuk memecahkan masalah-masalah

pengenalan (Elsiany 1999), misalnya pengenalan pola, pengenalan suara dan pengenalan ucapan. Tujuan dari JST adalah bagaimana membuat komputer mampu berpikir seperti manusia.

JST mempunyai beberapa metode, seperti propagasi balik (*back propagation/BP*), *Self Organizing Map* (SOM), dan *Learning Vector Quantization* (LVQ).

Paper akan membahas mengenai Propagasi Balik.

Propagasi balik adalah metode jaringan saraf tiruan yang mempunyai target dalam proses pembelajaran (*supervised training*). Propagasi balik terdiri dari dua bagian besar yaitu pembelajaran maju (*feed forward learning*) dan pembelajaran balik (*feed backward learning*). Pada pembelajaran maju, nilai error didapatkan dengan data target dan hasil output dari pembelajaran saat ini (*current learning*). Nilai error ini akan dipropagasikan ke seluruh lapisan-lapisan unit komputasi sebelumnya dalam jaringan untuk dilakukan update terhadap bobot koneksi antara pemroses.

Kelemahan dari propagasi balik (BP) adalah komputasi waktu yang besar terutama jika data yang harus dipelajari cukup besar. Untuk itu, teknik ini perlu dimodifikasi sehingga dapat mempercepat waktu komputasi.

Renato De (1998), mengusulkan algoritma *Successive Overrelaxation (SOR) Back Propagation* dengan melakukan modifikasi komputasi perhitungan error pada bobot koneksi ke lapisan keluaran sebanyak dua kali. Pada propagasi balik standar komputasi dilakukan satu kali. Dengan modifikasi tersebut, Propagasi Balik *Successive Overrelaxation* lebih cepat mencapai konvergensi dalam fase pembelajaran.

2. Metode Penelitian / Eksperimental

Propagasi Balik Standar

Aturan pembelajaran yang berlaku pada algoritma propagasi balik standar, adalah sebagai berikut ²:

1. Inisialisasi bobot awal dengan menggunakan cara inisialisasi Nguyen-Widrow.
2. Untuk tiap pola pelatihan, lakukan langkah 3 dan 4.
3. Lakukan langkah komputasi maju :

- a. Setiap neuron di lapisan masukan menerima sinyal x_i dan meneruskannya ke lapisan tersembunyi.
- b. Setiap neuron di lapisan tersembunyi menjumlahkan sinyal masukan yang dikalikan dengan bobot bersesuaian :

$$z_{in_j} = v_{ij} + \sum_{i=1}^n x_i v_{ij}$$

(1)

Lalu menghitung sinyal keluaran dengan fungsi aktivasi :

$$z_j = f(z_{in_j})$$

(2)

dan meneruskan sinyal ini pada semua neuron di lapisan keluaran.

- c. Setiap neuron di lapisan keluaran menjumlahkan sinyal masukan yang dikalikan dengan bobot yang bersesuaian :

$$z_{in_k} = w_{jk} + \sum_{j=1}^k z_j w_{jk}$$

(3)

lalu menghitung sinyal keluaran dengan fungsi aktivasi :

$$y_k = f(z_{in_k})$$

(4)

4. Lakukan komputasi mundur :

- a. Komputasi di lapisan keluaran :

- Setiap neuron di lapisan keluaran diberikan target masukan tertentu.
- Hitung sinyal error δ_k untuk setiap neuron k di lapisan ini, dengan menggunakan fungsi kesalahan kuadratis :

$$\delta_k = (t_k - y_k) f'(z_{in_k})$$

(5)

- Hitung modifikasi bobot untuk setiap neuron di lapisan tersembunyi :

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

(6)

- Hitung modifikasi bias untuk setiap neuron di lapisan keluaran :

$$\Delta w_{zk} = \alpha \delta_k$$

(7)

b. Komputasi di lapisan tersembunyi

- Hitung sinyal error δ_j untuk setiap neuron j di lapisan tersembunyi :

$$\delta_j = \left(\sum_{k=1}^m \delta_k w_{jk} \right) f'(z_{in_j})$$

(8)

- Hitung modifikasi bobot untuk hubungan antar neuron bersangkutan dengan semua neuron di lapisan masukan :

$$\Delta v_{ij} = \alpha \delta_j x_i$$

(9)

- Hitung modifikasi bias untuk setiap neuron di lapisan tersembunyi :

$$\Delta v_{zj} = \alpha \delta_j$$

(10)

- Ubah bobot dan bias :

$$w'_{jk} = w_{jk} + \Delta w_{jk}$$

(11)

$$w'_{ij} = w_{ij} + \Delta v_{ij}$$

(12)

5. Uji kriteria kesalahan, bila kesalahan lebih besar dari batas yang telah ditetapkan, ulangi langkah 2 dan 4. Bila tidak, pelatihan selesai.

Propagasi Balik SOR

Aturan pembelajaran yang berlaku pada algoritma propagasi balik (BP) SOR adalah sebagai berikut (Elsiany 1999):

1. Inisialisasi bobot awal dengan menggunakan cara inisialisasi Nguyen-Widrow.
2. Untuk tiap pola pelatihan, lakukan langkah 3 dan 4.
3. Lakukan langkah komputasi maju :
 - a. Setiap neuron di lapisan masukan menerima sinyal xi dan meneruskannya ke lapisan tersembunyi.
 - b. Setiap neuron di lapisan tersembunyi menjumlahkan sinyal masukan yang dikalikan dengan bobot bersesuaian :

$$z_{in_j} = v_{jk} + \sum_{i=1}^n x_i v_{ij}$$

(13)

Lalu menghitung sinyal keluaran dengan fungsi aktivasi :

$$o_j = f(z_{in_j})$$

(14)

dan meneruskan sinyal ini pada semua

neuron di lapisan keluaran.

- c. Setiap neuron di lapisan keluaran menjumlahkan sinyal masukan yang dikalikan dengan bobot yang bersesuaian :

$$y_{in_k} = w_{0k} + \sum_{j=1}^k z_j w_{jk}$$

(15)

lalu menghitung sinyal keluaran dengan fungsi aktivasi :

$$y_j = f(y_{in_k})$$

(16)

4. Lakukan komputasi mundur :

- a. Komputasi di lapisan keluaran :

- Setiap neuron di lapisan keluaran diberikan target masukan tertentu.
- Hitung sinyal error δ_k untuk setiap neuron k di lapisan ini, dengan menggunakan fungsi kesalahan kuadratis :

$$\delta_k = (t_k - y_k) f'(y_{in_k})$$

(17)

- Hitung modifikasi bobot untuk setiap neuron di lapisan tersembunyi :

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

(18)

- Hitung modifikasi bias untuk setiap neuron di lapisan keluaran :

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

(19)

- Ubah bobot menjadi :

$$w_{jk}^{(new)} = w_{jk}^{(old)} + \Delta w_{jk}$$

(20)

- Ubah bias menjadi :

$$w_{0k}^{(new)} = w_{0k}^{(old)} + \Delta w_{0k}$$

(21)

- Ubah nilai net pada neuron keluaran dengan menggunakan bobot terbaru di atas :

$$y_{in_k} = y_{in_k} + \Delta w_{jk} f(z_{in_j})$$

(22)

- b. Komputasi di lapisan tersembunyi :

- Hitung sinyal error δ_k yang baru untuk setiap neuron k di lapisan ini dengan menggunakan nilai net terbaru pada neuron di lapisan keluaran :

$$\delta_k = (t_k - f(y_{in_k}))$$

(23)

- Hitung sinyal error δ_j untuk setiap neuron j di lapisan tersembunyi :

$$\delta_j = \left(\sum_{k=1}^m \delta_k w_{jk}^{(new)} \right) f'(z_{in_j})$$

(24)

- Hitung modifikasi bobot untuk hubungan antar neuron bersangkutan dengan semua neuron di lapisan masukan :

$$\Delta v_{ij} = \alpha \delta_j x_i$$

(25)

- Hitung modifikasi bias untuk setiap neuron di lapisan tersembunyi :

$$\Delta v_{c_j} = \alpha \delta_j$$

(26)

- Ubah bobot dan bias :

$$v_{ij}^{(new)} = v_{ij}^{(old)} + \Delta v_{ij}$$

(27)

$$v_{c_j}^{(new)} = v_{c_j}^{(old)} + \Delta v_{c_j}$$

(28)

5. Uji kriteria kesalahan, bila kesalahan lebih besar dari batas yang telah ditetapkan, ulangi langkah 2 dan 4. Bila tidak, pelatihan selesai.

Eksperimental setup : menggunakan MATLAB

1. Percobaan awal menggunakan data iris dari Iris Flower Data Set. Data tersebut dibagi 2 : 100 data untuk data training dan 50 untuk data testing. Format data target :

1 0 0 : setosa

0 1 0 : versicolor

0 0 1 : virginica

Algoritma pembelajaran propagasi balik dan propagasi balik SOR diterapkan pada data tersebut dengan beberapa parameter: laju pembelajaran (α) 0.3, momentum 0.2, dan neuron tersembunyi (hidden) sebanyak 6.

2. Percobaan selanjutnya menggunakan data feret (*face recognition technology*) dari laboratorium computational logic Fakultas ilmu komputer UI. Percobaan dilakukan dalam 3 skenario :
 - a. 70/30 : 70 data training dan 30 data testing
 - b. 50/50 : 50 data training dan 50 data testing
 - c. 30/70 : 30 data training dan 70 data testing

Format data target :

1000000000 : orang ke-1

0100000000 : orang ke-2

0010000000 : orang ke-3

0001000000 : orang ke-4

0000100000 : orang ke-5

0000010000 : orang ke-6

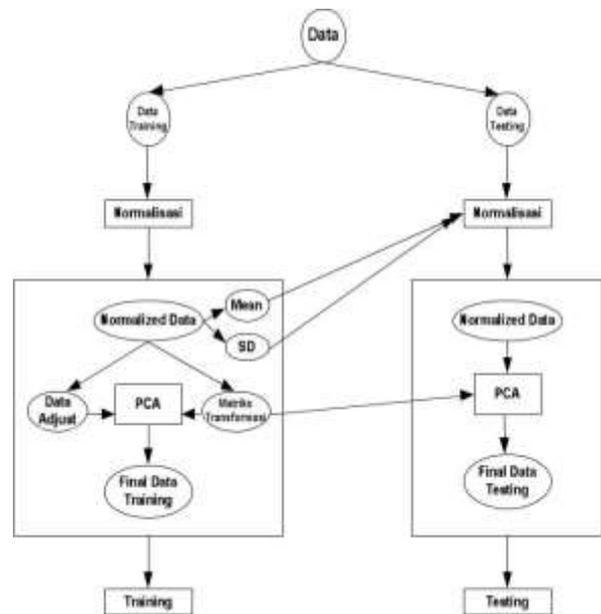
0000001000 : orang ke-7

0000000100 : orang ke-8

0000000010 : orang ke-9

0000000001 : orang ke-10

Algoritma pembelajaran propagasi balik dan propagasi balik SOR diterapkan pada data tersebut dengan parameter : laju pembelajaran (α) 0.3, momentum 0.2, dan neuron tersembunyi 40, 50, dan 60.



Gambar 1 Blok diagram percobaan

Data Iris

Diperkenalkan oleh Sir Ronald Aylmer Fisher pada tahun 1936. Bunga iris memiliki 3 spesies, antara lain setosa, versicolor, dan virginica.



Gambar 2 Setosa, versicolor, dan virginica

Data iris memiliki 4 feature : sepal length, sepal width, petal length, dan petal width.



Gambar 3 Feature iris

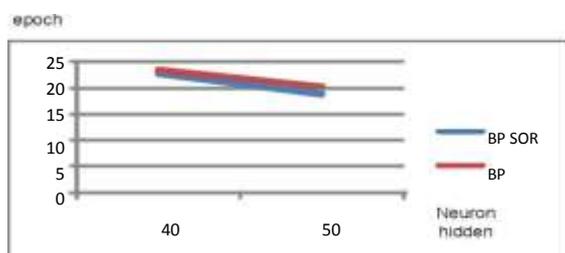
Data FERET yang digunakan adalah data wajah 10 orang dengan 10 keadaan yang berbeda-beda. Data berbentuk file bmp dengan ukuran 30x40 piksel.

3. Hasil Penelitian

3.1 Hasil Proses Pembelajaran dalam epoch

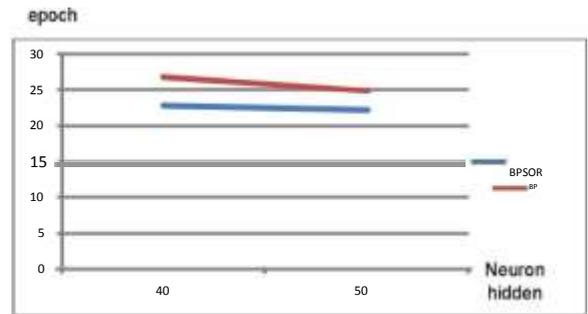
Berikut adalah jumlah epoch hasil proses pembelajaran menggunakan BP standar dan BP SOR dengan alpha 0.5, momentum 0.1.

Data training/Data testing=70/30



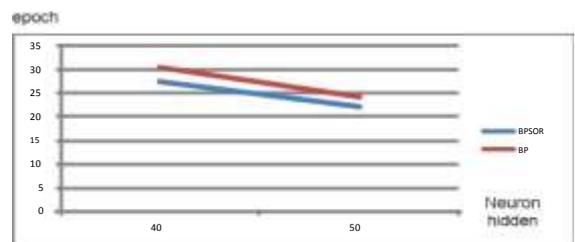
GrGrafik 1: Jumlah epoch pembelajaran berdasarkan jumlah neuron hidden pada perbandingan data training: testing 70:30

Data training/Data testing=50/50



GrGrafik 2: Jumlah epoch pembelajaran berdasarkan jumlah neuron hidden pada perbandingan data training: testing 50:50

Data training/Data testing=30/70



Grafik 3: Jumlah epoch pembelajaran berdasarkan jumlah neuron hidden pada perbandingan data training: testing 30:70

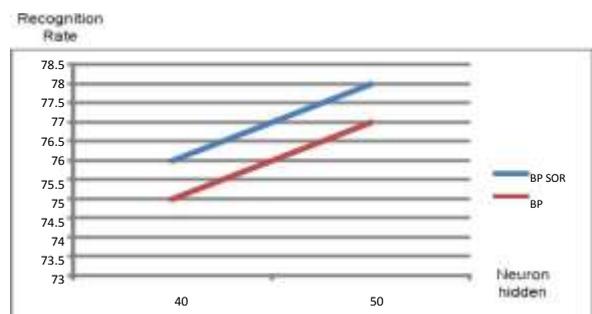
Sebagai pembandingan ditampilkan hasil epoch untuk pembelajaran data iris menggunakan BP Standard dan BP SOR dengan menggunakan dengan alpha 0.5, momentum 0.1, neuron hidden 6 dan data training/testing 100/50.

Tabel 1 Jumlah Epoch Proses Pembelajaran

EPOCH											
P1		P2		P3		P4		P5		SOR	BP
SOR	BP	SOR	BP	SOR	BP	SOR	BP	SOR	BP	Average	Average
2776	4215	1229	3532	5185	3574	5701	4657	2236		2876	2988

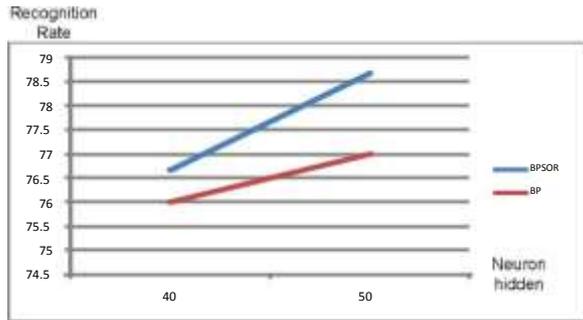
3.2 Hasil Recognition Rate

Data training/Data testing=70/30

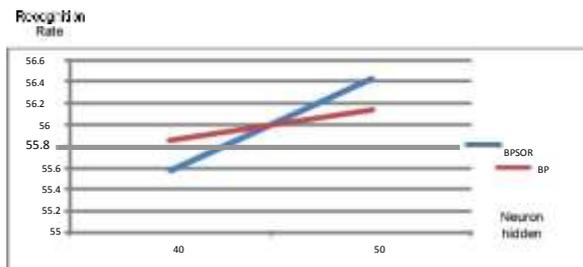


GrGrafik 4: Tingkat Recognition rate(%) berdasarkan jumlah

neuron hidden pada perbandingan data training: testing 30:70
Data training/Data testing=50/50



GrGrafik 5: Tingkat Recognition rate(%) berdasarkan jumlah neuron hidden pada perbandingan data training: testing 50:50
Data training/Data testing=30/70



Gr Grafik 6: Tingkat Recognition rate(%) berdasarkan jumlah neuron hidden pada perbandingan data training: testing 30:70

Sebagai pembanding berikut adalah hasil *recognition rate* untuk Data Iris BP Standar dan BP SOR dengan menggunakan dengan alpha 0.5, momentum 0.1, neuron hidden 6.

Tabel 2 Perbandingan rate recognition BP SOR/BP pada 5 percobaan

RECOGNITION RATE (%)											
P1		P2		P3		P4		P5		SOR	BP
SOR	BP	SOR	BP	SOR	BP	SOR	BP	SOR	BP	Average	Average
98	82	96	98	98	92	96	98	96	98	96.8	93.6

Catatan:

P1 : percobaan 1, P2 : percobaan 2 dan seterusnya.

4.Pembahasan

4.1 Pembahasan Peningkatan jumlah epoch

Dari hasil perbandingan proses pembelajaran terlihat bahwa jumlah epoch yang diperlukan dalam mencapai konvergensi dengan menggunakan metode BP SOR lebih sedikit dari pada metode BP pada semua kasus.

Berikut adalah perbandingan rata-rata epoch menggunakan BP SOR dengan BP.

Tabel 3 Perbandingan jumlah epoch BP SOR terhadap BP

Epoch Average BP SOR/BP	
Feret	Iris
0.943910052	0.96251673

Peningkatan kecepatan komputasi diharapkan dengan perbandingan jumlah epoch BP SOR dengan BP yang semakin kecil. Karena dalam 1 epoch, terjadi 2 kali komputasi nilai error di BP dan 3 kali komputasi di BP SOR maka peningkatan kecepatan komputasi BP SOR dapat diklaim jika perbandingan jumlah epoch menggunakan BP SOR dengan BP di bawah 2/3 atau 0.67.

Dari percobaan menggunakan data Feret dan Iris terlihat bahwa perbandingan epoch menggunakan BP SOR dengan BP adalah 0.9. Ini berarti, penurunan jumlah epoch menggunakan BP SOR tidak cukup signifikan mempercepat komputasi pembelajaran.

Dari pengamatan terhadap data feret dan iris, terlihat bahwa pengurangan jumlah epoch pada data feret sedikit lebih baik dari data Iris. Hal ini menunjukkan data feret lebih *separable* dibandingkan data iris. Jadi penggunaan BP SOR, akan menghasilkan hasil pengurangan yang signifikan jika:

1. Data untuk setiap kategori target lebih *separable*
2. Dalam setiap kategori mempunyai variasi yang banyak

4.2 Pembahasan Recognition Rate

Dari data *recognition rate* seluruh kasus, diperoleh rata-rata *recognition rate* sebagai berikut:

Tabel 4 Perbandingan rate recognition BP SOR/BP

RR Average BP SOR/BP	
Feret	Iris
1.007527455	1.03418803

Baik dari data feret maupun iris, terlihat pada semua kasus *recognition rate* menggunakan metode BP SOR meningkat tipis dibandingkan dengan BP SOR. Pada Data feret, peningkatan hanya 0.7 % dan pada data iris hanya 3.4%. Dari nilai tersebut dapat disimpulkan

bahwa penggunaan BP SOR tidak dapat meningkatkan *recognition rate* secara signifikan dibandingkan penggunaan BP.

Selain perbandingan *recognition rate* antara metode BP SOR dan BP, nilai *recognition rate* pada data feret dan iris dan nilai *recognition rate* antara kasus pada perbandingan data training dengan testing cukup menarik untuk diamati.

Nilai *recognition rate* pada data feret rata-rata menunjukkan berkisar antara kisaran 70-80% untuk training/testing 70/30 dan 50/50. Sedangkan pada data iris pada *recognition rate* mencapai 96%. Dari data tersebut terlihat bahwa *recognition rate* pada data feret baik menggunakan BP SOR maupun BO menurun sampai ke 70% artinya terdapat penurunan hingga 20%.

Penurunan tersebut disebabkan jumlah sampel pembelajaran pada data iris 5 kali lebih banyak dibandingkan pada data feret. Dengan jumlah sampel yang banyak maka software BP SOR maupun BP dapat belajar dengan lebih baik dan dapat mengenali data testing dengan lebih akurat.

Pada data feret dengan variasi perbandingan data training dan testing yaitu 70/30 dan 50/50, *recognition rate* yang dicapai lebih dari 70%. Sedangkan pada perbandingan data training dengan testing 30/70 *recognition rate* turun pada kisaran 50-60%. Ini berarti terdapat penurunan *recognition rate* sekitar 20%.

Penurunan ini disebabkan penurunan jumlah sampel yang digunakan. Pada perbandingan data training /testing 50/50 *recognition ratenya* relatif sama dengan 70/30. Meskipun jumlah sampel pada 50:50 berkurang namun jumlah sampel tersebut masih cukup mampu dipelajari oleh sistem untuk bisa mengenali data testing seakurat pengenalan dengan data training/testing 70/30. Sedangkan pada kasus perbandingan data training/testing 30/70, jumlah sampel sudah tidak mampu lagi mengenali data testing seakurat data training 70/30.

5. Kesimpulan

Dari pembahasan mengenai jumlah epoch dan hasil pengenalan data testing menunjukkan bahwa penggunaan BPSOR pada data iris dan data feret tidak

berhasil meningkatkan peromansi komputasi karena pengurangan epoch dan peningkatan *recognition rate* tidak signifikan. Jadi pada pengenalan data feret dan iris, metode BPSOR tidak perlu digunakan untuk menggantikan metode BP. Metode BPSOR dapat digunakan untuk meningkatkan kecepatan komputasi dengan jika data yang tersedia memiliki seperatabilitas yang tinggi antar kategori yang berbeda dan untuk setiap kategori memiliki variasi image yang banyak.

Daftar Referensi

- [1] Heck-Nielsen, Robert (1988), Neurocomputing picking the human brain, IEEE Spectrum.
- [2] Salomy, Elsianny, Algoritma Successive Overrelaxation (SOR) Backpropagation untuk Perbaikan Metode Pelatihan Pada Sistem Pengenal Huruf Tulisan Tangan, Fakultas Ilmu Komputer, Universitas Indonesia, 1999.
- [3] Iris Flower Data Set , http://en.wikipedia.org/wiki/Iris_flower_data_set
- [4] Leone, Renato De., Rosario Capparuccia, Emanuela Mereli, A Successive Overrelaxatiob Backpropagation Algorithm for Neural Network Training, IEEE Transaction on Neural Network, Vol. 9, No.3, 1998

Biografi



Riwinoto dilahirkan di Tegal, Provinsi Jawa Tengah, Indonesia pada 6 Agustus 1979. Lulus pendidikan S1 dari teknik informatika ITB tahun 2003 dan S2 dari ilmu komputer UI tahun 2010. Beliau bekerja sebagai Dosen Jurusan Teknik Informatika Politeknik Batam sejak tahun 2003-Sekarang.

Fokus penelitian yang dilakukan pada intelegensia buatan dan biomedik informatika dengan menggunakan pengolahan gambar, dan sinyal.



Wahyudi, Lahir di Jakarta, Indonesia, 28 Desember 1974. Memperoleh gelar sarjana pada bidang Sistem Komputer pada tahun 2001 dari STMIK Budi Luhur, Indonesia; gelar Magister Ilmu Komputer pada tahun 2010 dari Universitas Indonesia.

Saat ini beliau adalah bertugas di Unit Komputasi, Pusat Teknologi Dirgantara Terapan, Lembaga Penerbangan dan Antariksa(LAPAN) sebagai Perekayasa.



Tony Lahir di Medan, Indonesia, 10 Agustus 1982. Memperoleh gelar sarjana pada bidang Sistem Komputer pada tahun 2005 dari Fakultas Teknologi Informasi (FTI) Universitas Tarumanagara (Untar). Sejak tahun 2008 sampai sekarang sedang mengambil Magister Ilmu Komputer di Fakultas Ilmu Komputer (Fasilkom) Universitas Indonesia (UI).. .

Saat ini tercatat sebagai pengajar di Fakultas Teknologi Informasi, Universitas Tarumanagara sejak tahun 2006



Bahrun Said Reinhoran, Lahir di Kebumen, Indonesia, 12 September 1980. Memperoleh gelar sarjana pada bidang Teknik Elektro pada tahun 2003 dari Universitas Indonesia, gelar M.Kom pada bidang Ilmu Komputer pada tahun 2010 dari Universitas Indonesia. Saat ini merupakan staff pengajar pada STMIK Nusa Mandiri



Benyamin Kusumoputro, Lahir di Bandung, Indonesia, 17 November 1957. Memperoleh gelar sarjana pada bidang Fisika, MIPA pada tahun 1981 dari Institut Teknologi Bandung, Indonesia, gelar M.Eng. bidang Optoelectronics and Laser Application pada tahun 1984 dari Tokyo Institute of Technology, Jepang, dan gelar Dr. Eng. bidang Electrical and Electronics Engineering, dari Tokyo Institute of Technology, Jepang pada tahun 1993.

Saat ini beliau adalah profesor dan pengajar di Fakultas Ilmu Komputer, Universitas Indonesia, Indonesia. Beliau juga menjabat sebagai ketua Laboratorium Kecerdasan Buatan