

**UNIVERSITAS INDONESIA**

**PENGEMBANGAN JARINGAN SARAF TIRUAN *RADIAL BASIS FUNCTION* DENGAN METODE *BACKWARD* DAN METODE *ORTHOGONAL LEAST SQUARE* MENGGUNAKAN KLASIFIKASI SOM**

**SEMINAR**

**Annisaa Primadini**

**1006675171**

**FAKULTAS TEKNIK UNIVERSITAS INDONESIA**

**PROGRAM STUDI TEKNIK ELEKTRO**

**DEPOK**

**JANUARI 2014**



# UNIVERSITAS INDONESIA

**PENGEMBANGAN JARINGAN SARAF TIRUAN *RADIAL BASIS FUNCTION* DENGAN METODE *BACKWARD* DAN METODE *ORTHOGONAL LEAST SQUARE* MENGGUNAKAN KLASIFIKASI SOM**

**SEMINAR**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik**

**Annisaa Primadini**

**1006675171**

**FAKULTAS TEKNIK UNIVERSITAS INDONESIA**

**PROGRAM STUDI TEKNIK ELEKTRO**

**DEPOK**

**JANUARI 2014**

# KATA PENGANTAR

Puji dan syukur ke hadirat Tuhan Yang Maha Esa, karena berkat rahmat dan karunia-Nya lah penulis dapat menyelesaikan seminar ini. Seminar ini ditulis sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik.

Penulis menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan seminar, sangatlah sulit bagi penulis untuk dapat menyelesaikan seminar ini. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. [Prof. Dr.Eng. Drs. Benyamin Kusumoputro M.Eng.](https://academic.ui.edu/main/Info/Lecturer?nip=195711171987031001), selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan penulis dalam penyusunan seminar ini;
2. Orang tua dan keluarga penulis yang telah memberikan bantuan dukungan material dan moral;
3. Lauren dan Muhammad Ashari selaku teman bimbingan;
4. Teman-teman penulis yang tidak dapat disebutkan satu per satu yang telah membantu penulis, baik secara aktif ataupun pasif, dalam menyelesaikan seminar ini.

Akhir kata, penulis berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga seminar ini membawa manfaat bagi pengembangan ilmu.

Depok, Januari 2014

Penulis

# ABSTRAK

Nama : Annisaa Primadini

Program Studi : Teknik Elektro

Judul : Pengembangan Jaringan Saraf Tiruan Radial Basis Function dengan Metode Backward dan Metode Orthogonal Least Square Menggunakan Klasifikasi SOM

Perkembangan teknologi masa kini semakin dituntut untuk dapat mampu menyamai pola pemikiran manusia. Diharapkan terbentuk suatu teknologi cerdas yang mempunyai kecepatan dan akurasi tinggi. Seminar ini melakukan percobaan penerapan jaringan saraf tiruan *Radial Basis Function* menggunakan metode backward dan *Orthogonal Least Square* serta pemanfaatan metode klasifikasi *Self Organizing Map*. Berdasarkan hasil percobaan yang dilakukan, jaringan saraf tiruan *Radial Basis Function* dengan metode klasifikasi SOM sudah menampilkan performa yang baik. Penggunaan Metode Backward pada RBF mempunyai performa yang lebih baik dibandingkan penggunaan Metode OLS. Bagaimanapun juga, jaringan saraf tiruan *Radial Basis Function* masih mempunyai ruang untuk pengembangan lebih lanjut.

Kata Kunci :

Jaringan Saraf Tiruan, *Radial Basis Function, Self Organizing Map, Pattern Recognition*

# DAFTAR ISI

[HALAMAN JUDUL ii](#_Toc377121972)

[KATA PENGANTAR ii](#_Toc377121973)

[ABSTRAK iii](#_Toc377121974)

[DAFTAR ISI iv](#_Toc377121975)

[DAFTAR TABEL vi](#_Toc377121976)

[DAFTAR GAMBAR vii](#_Toc377121977)

[BAB 1 PENDAHULUAN 1](#_Toc377121978)

[1. 1 Latar Belakang 1](#_Toc377121979)

[1. 2 Tujuan 2](#_Toc377121980)

[1. 3 Batasan Masalah 2](#_Toc377121981)

[1. 4 Metodologi Penulisan 2](#_Toc377121982)

[1. 5 Sistematika Penulisan 3](#_Toc377121983)

[BAB 2 JARINGAN SARAF TIRUAN *RADIAL BASIS FUNCTION* 4](#_Toc377121984)

[2. 1 Jaringan Saraf Tiruan 4](#_Toc377121985)

[2.1.1. Pengertian Jaringan Saraf Tiruan 4](#_Toc377121986)

[2.1.2. Arsitektur Jaringan Saraf Tiruan 5](#_Toc377121987)

[*2. 2* *Radial Basis Function* 9](#_Toc377121988)

[2. 3 *Self-Organizing Map* 11](#_Toc377121989)

[BAB 3 JARINGAN SARAF TIRUAN *RADIAL BASIS FUNCTION* DENGAN METODA BACKWARD 13](#_Toc377121990)

[3. 1 Metode Backward pada Jaringan Saraf Tiruan *Radial Basis Function* 13](#_Toc377121991)

[3. 2 Algoritma dan Program Percobaan Jaringan Saraf Tiruan *Radial Basis Function* Metode Backward 16](#_Toc377121992)

[3. 3 Hasil Percobaan Jaringan Saraf Tiruan *Radial Basis Function* Metode Backward 21](#_Toc377121993)

[3. 4 Analisis Percobaan Jaringan Saraf Tiruan *Radial Basis Function* Metode Backward 26](#_Toc377121994)

[BAB 4 JARINGAN SARAF TIRUAN *RADIAL BASIS FUNCTION* DENGAN METODA OLS 27](#_Toc377121995)

[4. 1 Metode OLS pada Jaringan Saraf Tiruan *Radial Basis Function* 27](#_Toc377121996)

[4. 2 Algoritma dan Program Percobaan Jaringan Saraf Tiruan *Radial Basis Function* Metode OLS 30](#_Toc377121997)

[4. 3 Hasil Percobaan Jaringan Saraf Tiruan *Radial Basis Function* Metode OLS 36](#_Toc377121998)

[4. 4 Analisis Percobaan Jaringan Saraf Tiruan *Radial Basis Function* Metode OLS 40](#_Toc377121999)

[BAB 5 KOMPARASI JARINGAN SARAF TIRUAN ALGORITMA *RADIAL BASIS FUNCTION* DENGAN ALGORITMA *BACKPROPAGATION* 42](#_Toc377122000)

[5. 1 Komparasi Hasil Percobaan Jaringan Saraf Tiruan Algoritma *Radial Basis Function* Metode Backforward, Hasil Percobaan Jaringan Saraf Tiruan Algoritma *Radial Basis Function* Metode OLS, dan Hasil Percobaan Jaringan Saraf Tiruan Algoritma *Backpropagation* 42](#_Toc377122001)

[5. 2 Analisis Komparasi Hasil Percobaan Jaringan Saraf Tiruan Algoritma *Radial Basis Function* Metode Backforward, Hasil Percobaan Jaringan Saraf Tiruan Algoritma *Radial Basis Function* Metode OLS, dan Hasil Percobaan Jaringan Saraf Tiruan Algoritma *Backpropagation* 46](#_Toc377122002)

[KESIMPULAN 48](#_Toc377122003)

[DAFTAR PUSTAKA 49](#_Toc377122004)

# DAFTAR TABEL

[Tabel 3.3. 1 Percobaan RBF Backward dengan Alpha = 0.5 22](#_Toc377122005)

[Tabel 3.3. 2 Percobaan RBF Backward dengan Alpha = 0.6 22](#_Toc377122006)

[Tabel 3.3. 3 Percobaan RBF Backward dengan Alpha = 0.7 23](#_Toc377122007)

[Tabel 3.3. 4 Perbandingan Pengujian RBF Backward Klasifikasi SOM Dengan Klasifikasi Cmeans Untuk Alpha = 0.5 24](#_Toc377122008)

[Tabel 4.3. 1 Percobaan RBF OLS Tanpa Fungsi Aktivasi Dan Tanpa Bias 37](#_Toc377122009)

[Tabel 4.3. 2 Percobaan RBF OLS Tanpa Fungsi Aktivasi dengan Bias 38](#_Toc377122010)

[Tabel 4.3. 3 Percobaan RBF OLS Menggunakan Fungsi Aktivasi 38](#_Toc377122011)

[Tabel 4.3. 4 Perbandingan RBF OLS Klasisfikasi SOM Dengan Klasifikasi C-Means Untuk Kondisi Nilai Alpha 0.5, Tidak Menggunakan Fungsi Aktivasi Dan Menggunakan Bobot Bias 39](#_Toc377122012)

# DAFTAR GAMBAR

[Gambar 2. 1 Anatomi Neuron (B. Kosko, 1992) 5](#_Toc377122013)

[Gambar 2. 2 Model Matematis dari Sebuah Neuron (Neural Network Control of Robot Manipulators and Nonlinear Systems, 1999) 6](#_Toc377122014)

[Gambar 2. 3 Fungsi Aktifasi Sigmoid (Neural Network Control of Robot Manipulators and Nonlinear Systems, 1999) 6](#_Toc377122015)

[Gambar 2. 4 Jaringan Saraf Tiruan *Multilayered* 8](#_Toc377122016)

[Gambar 2. 5 Model Jaringan Saraf Tiruan *Radial Basis Function* 10](#_Toc377122017)

[Gambar 2. 6 Arsitektur Jaringan SOM 12](#_Toc377122018)

[Gambar 3.3 1 Grafik Perbandingan Recognition Rate Klasifikasi SOM Dengan C-Means Untuk Setiap Iterasi 24](file:///F:\Lecture\Semester%207\Seminar\Seminar%20-%20Final\Laporan%20Seminar%20Rev1%20Upoladed.docx#_Toc377122019)

[Gambar 3.3 2 Grafik Perbandingan Epoch RBF Klasifikasi SOM Dengan C-Means Untuk Setiap Iterasi 25](file:///F:\Lecture\Semester%207\Seminar\Seminar%20-%20Final\Laporan%20Seminar%20Rev1%20Upoladed.docx#_Toc377122020)

[Gambar 3.3 3 Grafik Perbandingan Cost Time RBF Klasifikasi SOM Dengan C-Means Untuk Setiap Iterasi 25](#_Toc377122021)

[Gambar 4.3 1 Grafik Perbandingan Recognition Rate RBF NN Klasifikasi SOM Dengan Cmeans 40](#_Toc377122022)

[Gambar 4.3 2 Grafik Perbandingan Cost Time Rbf Nn Klasifikasi Som Dengan Cmeans 40](#_Toc377122023)

[Gambar 5.1. 1 Grafik Perbandingan Waktu Komputasi Jaringan Saraf Tiruan 43](#_Toc377122024)

[Gambar 5.1. 2 Grafik Perbandingan Recognition Rate Jaringan Saraf Tiruan 44](#_Toc377122025)

[Gambar 5.1. 3 Grafik Perbandingan Jumlah Epoch Jaringan Saraf Tiruan 45](#_Toc377122026)

[Gambar 5.1. 4 Grafik Perbandingan Error Jaringan Saraf Tiruan 46](#_Toc377122027)

1. PENDAHULUAN
2. Latar Belakang

Dewasa ini, perkembangan teknologi telah menjadi bagian yang tidak terpisahkan dari kehidupan manusia. Teknologi sendiri memudahkan manusia dalam melakukan seluruh aktivitas, di mana pada suatu titik manusia hampir tidak bisa melakukan appapun tanpa adanya bantuan teknologi. Dengan ketergantungannya manusia pada teknologi yang berkembang dan terintegrasinya teknologi pada kehidupan manusia, manusia membutuhkan suatu teknologi yang cepat dan akurat serta dapat meniru pola pikir manusia sehingga teknologi tersebut tepat guna dalam mendampingi hidup manusia.

Untuk membentuk suatu teknologi yang dapat mengikuti pola pikir manusia, maka dibentuk suatu sistem yang mempunyai pola yang sama dengan kecerdasan manusia, yaitu dengan mengikuti cara kerja otak dan saraf manusia. Salah satu sistem yang dikembangkan dalam mengikuti kategori ini adalah Jaringan Saraf Tiruan. Sistem Jaringan Saraf Tiruan mengadaptasi sistem kerja saraf manusia dari penerimaan rangsangan hingga pengambilan keputusan untuk pemberian respon.

Dalam mengoptimalkan cara kerja sistem jaringan saraf tiruan telah dikembangkan berbagai algoritma untuk mendapatkan hasil yang cepat dan akurat. Pada seminar ini akan dibahas mengenai pengembangan algoritma *Radial Basis Function* yang memanfaatkan nilai tengah dan rentang data hasil pengelompokan. Kelebihan algoritma ini adalah bentuk algoritma yang lebih sederhana dibandingkan algoritma *Backpropagation*. Sehingga, diharapkan algoritma *Radial Basis Function* pada jaringan saraf tiruan mempunyai hasil komputasi dan keakuratan yang lebih baik.

1. Tujuan

Penelitian ini disusun berdasakan riset dan percobaan yang dilakukan dalam mengembangkan algoritma *Radial Basis Function* yang diaplikasikan pada pembelajaran jaringan saraf tiruan. Tujuan dari penelitian ini adalah sebagai berikut :

1. Mengembangkan algoritma *Radial Basis Function* untuk diterapkan dalam pelatihan Jaringan Saraf Tiruan
2. Menganalisa kinerja Jaringan Saraf Tiruan dengan algoritma *Radial Basis Function*
3. Mendapatkan waktu komputasi dan akurasi yang lebih baik dibandingkan dengan pelatihan menggunakan Jaringan Saraf Tiruan *Backpropagation*
4. Batasan Masalah

Fokus penelitian ini adalah pada waktu komputasi dan akurasi hasil pembelajaran jaringan saraf tiruan menggunakan algoritma *Radial Basis Function*. Dalam upaya tersebut, dilakukan beberapa pengembangan perhitungan matematis dari koreksi nilai tengah dan rentang data. Selain itu diterapkan pula metode *Self-Organizing Map* dalam proses klasifikasi pada pembelajaran Jaringan Saraf Tiruan *Radial Basis Function*. Setelah mendapatkan model matematis, model akan disimulasikan pada jaringan saraf tiruan dengan menggunakan MATLAB R2013a

1. Metodologi Penulisan

Metodologi yang digunakan selama melakukan penelitian dan penulisan laporan seminar ini adalah :

1. Studi Literatur

Penulis membaca buku, e-book, jurnal, seminar dan literatur lain yang berkaitan dengan penelitian ini

1. Konsultasi dengan dosen pembimbing

Penulis melakukan pertemuan dengan dosen pembimbing mengenai penelitian yang berlangsung.

1. Sistematika Penulisan

Laporan ini terbagi ke dalam 6 Bab, yaitu :

1. BAB 1 PENDAHULUAN

Bab ini membahas tentang latar belakang penelitian , tujuan penelitian, batasan masalah, metodologi penelitian dan sistematika penulisan laporan

1. BAB 2 JARINGAN SARAF TIRUAN *RADIAL BASIS FUNCTION*

Bab ini membahas tentang dasar teori Jaringan Saraf Tiruan, algoritma *Radial Basis Function* dan *Self-Organizing Map* yang digunakan untuk klasifikasi data.

1. BAB 3 JARINGAN SARAF TIRUAN *RADIAL BASIS FUNCTION* DENGAN METODE BACKWARD

Bab ini membahas tentang percobaan menggunakan algortima *Radial Basis Function* dengan metode Backward sesuai dengan batasan percobaan. Selain itu, pada bab ini juga akan membahas hasil dan analisa percobaan.

1. BAB 4 JARINGAN SARAF TIRUAN *RADIAL BASIS FUNCTION* DENGAN METODE OLS

Bab ini membahas tentang percobaan menggunakan algortima *Radial Basis Function* dengan metode OLS sesuai dengan batasan percobaan. Selain itu, pada bab ini juga akan membahas hasil dan analisa percobaan.

1. BAB 5 KOMPARASI JARINGAN SARAF TIRUAN ALGORITMA *RADIAL BASIS FUNCTION* DENGAN ALGORITMA *BACKPROPAGATION*

Bab ini membahas tentang komparasi dan analisis antara jaringan saraf tiruan algoritma *Radial Basis Function* dengan algoritma *Backpropagation*

1. KESIMPULAN

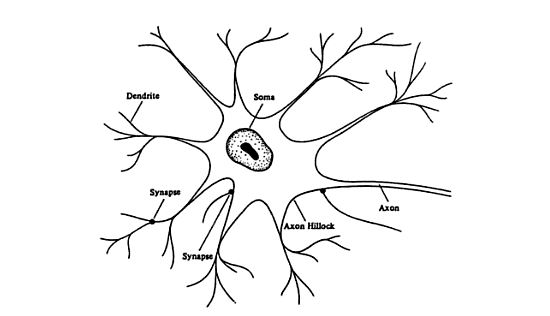
Bab ini membahas tentang kesimpulan dari hasil dan analisa dari penelitian yang telah dilakukan berhubungan dengan tujuan penelitian ini.

1. JARINGAN SARAF TIRUAN *RADIAL BASIS FUNCTION*

Pada bab ini dipaparkan teori mengenai jaringan saraf tiruan dan salah satu metodenya yaitu *Radial Basis Function* (RBF) serta salah satu metode *clustering* pada jaringan saraf tiruan yaitu *Self-Organizing Map* (SOM).

1. Jaringan Saraf Tiruan
   1. Pengertian Jaringan Saraf Tiruan

Jaringan saraf tiruan adalah suatu metode komputasi yang dibentuk berdasarkan proses pengolahan informasi biologis yang terjadi pada sistem saraf manusia. Perancangan struktur jaringan tersebut didasarkan pada unit terkecil dari sistem saraf manusia, yaitu neuron, seperti yang ditunjukkan oleh Gambar 2.1. Pada jaringan saraf manusia, sinyal yang ditangkap oleh *reseptor* akan disebarkan dalam bentuk beda potensial pada sel. *Dendrites* akan membawa sinyal informasi dari neuron lainnya menuju *soma* atau inti sel saraf, dengan mengalikan setiap sinyal dengan koefisien bobot pemindahan informasi (*transfer weighting coefficient*). Setelah signal komposit melewati suatu nilai *threshold* pada sel saraf (neuron) tersebut, sebuah sinyal potensial aksi akan disalurkan ke neuron lainnya melewati sinapsis.

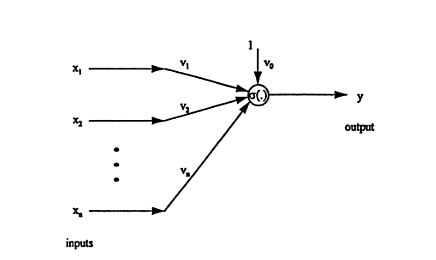


Gambar 2. Anatomi Neuron (B. Kosko, 1992)

* 1. Arsitektur Jaringan Saraf Tiruan

Dari proses pengolahan informasi pada sel saraf manusia yang telah dipaparkan sebelumnya, dibentuk suatu model matematis yang digunakan dalam komputasi jaringan saraf tiruan. Gambar 2.2 menunjukkan model matematis neuron, yang terdiri atas bobot *dendrite* , nilai *threshold* atau bias , penjumlahan dari sinyal yang sudah dipengaruhi oleh bobot neuron, dan penjumlahan fungsi nonlinear . Data input neuron adalah suatu sinyal sebanyak n pada waktu t dan output y(t) skalar yang di dapatkan dari perhitungan

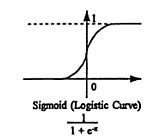
(2.1)



Gambar 2. Model Matematis dari Sebuah Neuron (Neural Network Control of Robot Manipulators and Nonlinear Systems, 1999)

Fungsi penjumlahan disebut dengan fungsi aktivasi. Terdapat beberapa pilihan fungsi aktivasi disesuakan dengan tingkah laku (*behaviour*) dari neuron atau jaringan tersebut. Fungsi aktivasi yang biasa digunakan adalah fungsi sigmoid, yang didefinisikan sebagai

(2.2)



Gambar 2. Fungsi Aktifasi Sigmoid (Neural Network Control of Robot Manipulators and Nonlinear Systems, 1999)

Pada struktur jaringan saraf tiruan, jumlah lapisan neuron dapat ditentukan sesuai dengan karakteristik sinyal input dan output serta algoritma jaringan saraf tiruan yang digunakan. Secara umum, jaringan saraf tiruan yang sering digunakan adalah multilayered neural network yang dapat dibagi menjadi tiga bagian, yaitu:

1. Lapisan masukan (*input layer*)

Lapisan ini bertugas menerima sinyal input dan meneruskannya ke hidden layer. Lapisan ini bertugas seperti *dendrites* jika dibandingkan dengan model asli jaringan saraf manusia.

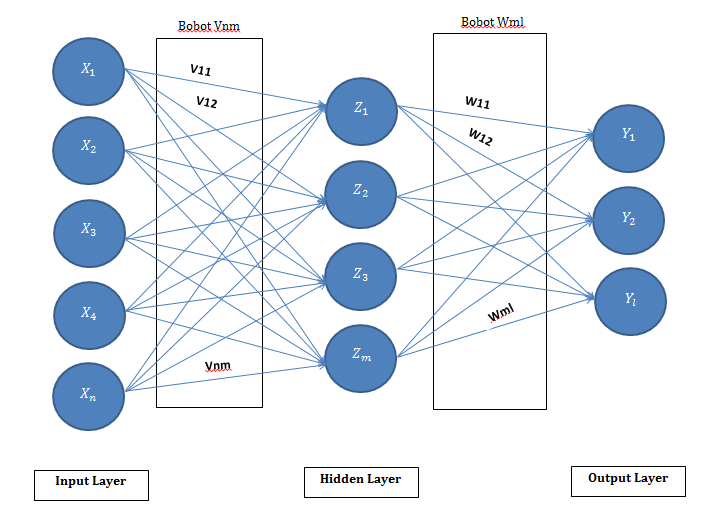
1. Lapisan tersembunyi (*hidden layer*)

Pada lapisan ini akan dilakukan komputasi menggunakan fungsi aktivasi. Hasil perhitungan dari lapisan ini akan diteruskan kembali ke lapisan berikutnya, yaitu output layer, sebagai sinyal input pada output layer.

1. Lapisan keluaran (*output layer*)

Lapisan ini juga akan melakukan komputasi dengan menggunakan fungsi aktivasi, dengan sinyal output hidden layer sebagai sinyal inputnya. Hasil perhitungan dari lapisan ini akan menghasilkan output yang akan dibandingkan dengan nilai target yang ingin dicapai sistem untuk memperoleh nilai *error*.

Struktur jaringan saraf tiruan multi-layer dapat dilihat pada Gambar 2.4 berikut



Gambar 2. Jaringan Saraf Tiruan *Multilayered*

Pada jaringan saraf tiruan terdapat dua tipe pembelajaran, yaitu :

1. Pembelajaran dengan pengarahan (*supervised*)

Metode pembelajaran *supervised* digunakan pada kumpulan data yang sudah diketahui target keluarannya, Pembelajaran dilakukan dengan menyajikan secara berturut pola-pola latihan masukan bersama dengan vektor target-keluaran yang berkaitan. Nilai bobot akan berubah sesuai dengan algoritma yang digunakan.

1. Pembelajaran tanpa pengarahan (*Unsupervised*)

Metode pembelajaran *Unsupervised* digunakan pada kumpulan data yang belum diketahui target keluarannya, sehingga hasil selama pembelajaran tidak dapat diketahui. Dalam proses pembelajaran, jaringan saraf tiruan akan membentuk beberapa unit kelompok dan mengelompokkan data input, lalu menentukan karakteristik dari masing-masing kelompok sehingga dapat menentukan unit tertentu untuk data input baru.

1. *Radial Basis Function*

*Radial Basis Function* adalah suatu metode jaringan saraf tiruan yang mempunyai struktur jaringan yang sederhana sehingga dapat menghindari waku komputasi yang panjang. Kelebihan metode RBF ini adalah mudah diimplementasikan dan tidak bergantung kepada jumlah dimensi dan kompleksitas geometris data.

Pada awalnya, metode *Radial Basis Function* pada jaringan saraf tiruan digunakan pada interpolasi multivariabel, dan untuk model RBF ini diperlukan hidden neuron dengan jumlah yang samap dengan banyaknya neuron input. Selanjutnya, pengembangan metode *Radial Basis Function* yang diaplikasikan sebagai unit perhitungan pada hidden layer dikembangkan oleh Broomhead dan Lowe, sehingga terbentuk metode *Radial Basis Function* yang lebih efisien.

Jaringan saraf tiruan *Radial Basis Function* menggunakan fungsi radial dalam perhitungannya, dimana pada fungsi radial ini nilai keluarannya akan simetris pada nilai tengah . Sehingga fungsi radial ini dapat dinayatakan dengan dalam fungsi jarak, yaitu

(2.3)

Berbagai macam fungsi radial dapat digunakan dalam algoritma jaringan saraf tiruan *Radial Basis Function*, antara lain :

1. Fungsi *thin-plate-spline*

(2.4)

1. Fungsi multikuadratik

(2.5)

Dimana adalah parameter *spread* atau variansi input.

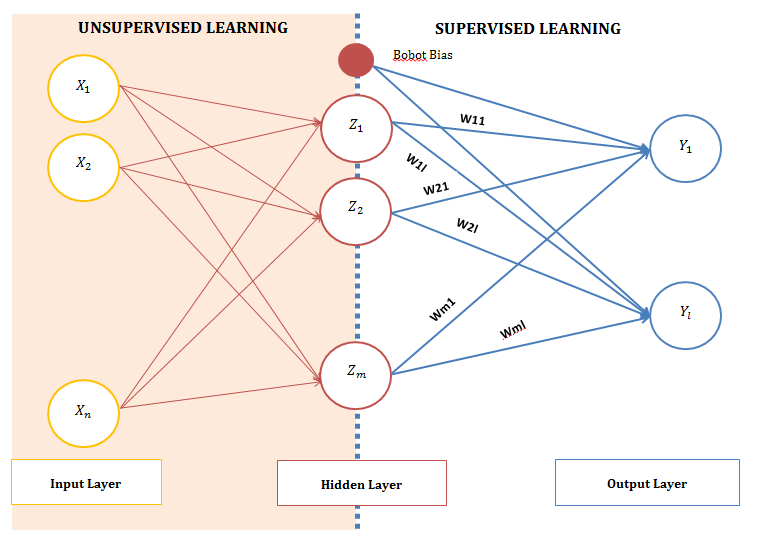
1. Fungsi invers multikuadratik

(2.6)

1. Fungsi Gaussian

(2.7)

Jaringan Saraf Tiruan *Radial Basis Function* memiliki tiga lapisan neuron, yang masing-masing mempunyai fungsi yang berbeda, dimodelkan sesuai dengan jaringan saraf tiruan multi-layer. Lapisan neuron pada Jaringan Saraf Tiruan *Radial Basis Function* diperlihatkan pada Gambar 2. 5. Lapisan tersebut adalah Input Layer, Hidden Layer dan Output Layer dengan masing-masing fungsi yang sudah dipaparkan pada bagian 2.1.2.



Gambar 2. Model Jaringan Saraf Tiruan *Radial Basis Function*

Pada Jaringan ini akan mengalami dua tahap pembelajaran dengan metode yang berbeda yaitu :

1. *Unsupervised* Learning

*Unsupervised* Learning terjadi pada lapisan masukan dan lapisan tersembunyi. Selama pembelajaran, jaringan saraf tiruan akan mengelompokkan kumpulan data input ke dalam beberapa kluster. Setiap kluster ini akan menjadi data input untuk *Supervised* Learning pada lapisan berikutnya. Pada penelitian ini, metode *Unsupervised* Learning yang digunakan adalah *Self-Organizing Map* (SOM)

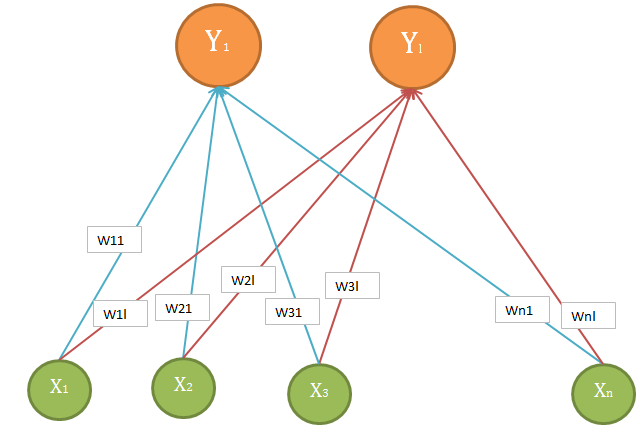
1. *Supervised* Learning

*Supervised* Learning terjadi pada lapisan tersembunyi dan lapisan keluaran. Selama pembelajaran, data input yang sudah dikelompokkan dalam beberapa kluster akan dilatih dengan nilai bobot tertentu dan nilai target keluaran yang sudah diketahui.

1. *Self-Organizing Map*

*Self-Organizing Map* atau SOM merupakan suatu metode jaringan saraf tiruan tidak terbimbing (*unsupervised*) yang biasanya digunakan untuk pengelompokan atau klasifikasi data. Metode SOM pertama kali diteliti oleh Teuvo Kohonen. Dengan menggunakan algoritma SOM, data input pada neuron input yang tadinya belum diketahui hubungan antar neuronnya dapat diklasifikasi ke dalam kelas-kelas tertentu yang terdiri atas sekumpulan neuron dengan karakteristik yang sama. Dengan data input berupa vektor, karakteristik tiap kelas dapat dinyatakan dalam dalam vektor pewakil. Vektor pewakil akan berubah nilainya setiap anggota pada kelas tersebut bertambah.

Dari segi arsitektur jaringan, SOM terdiri atas 2 lapisan, yaitu lapisan masukan (*input layer*) dan lapisan keluaran (*output layer*). Pada masing-masing kelas pada Untuk lebih jelas, arsitektur jaringan SOM dapat dilihat pada Gambar 2.6.



Gambar 2. Arsitektur Jaringan SOM

Jaringan Saraf Tiruan SOM dapat dilakukan dengan algoritma berikut :

Langkah 0 : Inisialisasi bobot atau vektor pewakil () dengan memilih salah satu vektor secara random sebanyak *l* jumlah kelas yang diinginkan

Langkah 1 : Jika kondisi henti gagal, lakukan langkah 2-6

Langkah 2 : Untuk setiap vektor masukan , lakukan langkah 3-6

Langkah 3 : Untuk setiap j, hitung

(2.8)

Untuk mencari jarak antar vektor

Langkah 4 : Tentukan vektor pemenang, yaitu vektor dengan indeks *l* dimana minimum

(2.9)

Langkah 5 : Untuk setiap neuron *l*, sesuaikan nilai bobot yang baru dengan

(2.10)

Langkah 6 : Periksa kondisi henti

1. JARINGAN SARAF TIRUAN *RADIAL BASIS FUNCTION* DENGAN METODA BACKWARD
2. Metode Backward pada Jaringan Saraf Tiruan *Radial Basis Function*

Pada Jaringan Saraf Tiruan *Radial Basis Function* terdapat dua jenis pembelajaran yaitu pembelajaran *unsupervised* dan *supervised*. Pembelajaran *unsupervised* dilakukan dengan metode SOM, seperti yang sudah dijelaskan pada bagian 2.3. Data input pada lapisan masukan (input layer) yang dilatih dengan SOM akan menghasilkan kumpulan data dengan masing-masing vektor pewakil yang merupakan nilai tengah dari kumpulan data tersebut. Kumpulan data dan vektor pewakilnya masing-masing akan diteruskan untuk kemudian melanjutkan pembelajaran *supervised* dengan metode backward atau *backpropagation* pada lapisan tersembunyi (hidden layer).

Pada lapisan tersembunyi, data input beserta vektor pewakil yang merupakan nilai tengah masing-masing kluster akan mengalami perhitungan dengan menggunakan fungsi Gaussian

(3.1)

Yang disederhanakan menjadi

(3.2)

(3.3)

Keluaran dari lapisan tersembunyi, Z, akan mengalami perhitungan dengan pengaruh dari bias serta bobot antara lapisan tersembunyi dengan lapisan keluaran. Keluaran pada lapisan keluaran merupakan hasil perhitungan fungsi sigmoid unipolar.

(3.4)

(3.5)

Proses perhitungan nilai keluaran Y disebut dengan propagasi maju atau *feedforward*. Setelah mengetahui nilai keluaran proses *feedforward* ini akan dihitung besarnya kesalahan yang terjadi pada keluaran terhadap nilai target yang bersesuaian.

(3.6)

Proses yang terjadi selanjutnya adalah proses propagasi balik atau *backpropagation*, yang akan mengoreksi nilai bobot dan bias dari lapisan tersembunyi ke lapisan keluaran. Selain itu, juga terjadi koreksi nilai tengah dan lebar data (parameter *spread*). Koreksi nilai-nilai tersebut didapatkan dari penurunan rumus kesalahan perhitungan (*error*) terhadap komponen koreksi. Nilai koreksi tersebut adalah :

1. Koreksi nilai bobot

Koreksi nilai bobot akan didapatkan dari turunan rumus kesalahan perhitungan (*error*) terhadap nilai bobot

(3.7)

Dimana,

(3.8)

(3.9)

Sehingga persamaan 3.7 menjadi,

(3.10)

1. Koreksi nilai tengah ()

Koreksi nilai tengah akan didapatkan dari turunan rumus kesalahan perhitungan (*error*) terhadap nilai tengah

(3.11)

Dimana,

(3.12)

(3.13)

(3.14)

(3.15)

Sehingga persamaan 3.11 menjadi,

(3.16)

1. Koreksi lebar data atau parameter *spread* ()

Koreksi lebar data akan didapatkan dari turunan rumus kesalahan perhitungan (*error*) terhadap lebar data

(3.17)

Dimana,

(3.18)

(3.19)

(3.20)

(3.21)

Sehingga persamaan 3.17 menjadi,

(3.22)

1. Algoritma dan Program Percobaan Jaringan Saraf Tiruan *Radial Basis Function* Metode Backward

Algoritma yang digunakan dalam melakukan pembelajaran jaringan saraf tiruan *Radial Basis Function* dengan metode backward adalah sebagai berikut :

1. Inisialisasi
2. Inisialisasi bobot dan bias antara lapisan tersembunyi dengan lapisan keluaran dengan menggunakan metode Nguyen-Widrow dan inisialisasi vektor pewakil dengan metode SOM.
3. Tentukan vektor masukan, vektor target yang bersesuaian, lebar data RBF, serta vektor nilai tengah data

%% Inisialisasi Bobot dengan Metode Nguyen

Wml = rand (m\_unit\_hidden, l\_unit\_output)-0.5 ;

beta2 = 0.7\*(m\_unit\_hidden^1/(l\_unit\_output));

norm\_Wml = sqrt(sum ((Wml.^2),1));

% Update bobot W

for m = 1:m\_unit\_hidden

Wml(m,:) = beta2\*(Wml(m,:)./norm\_Wml);

end

% ====== set bias W\_0l ========

W0l = 2\*beta2\*rand(l\_unit\_output,1)-0.5;

%% Inisialisasi Vektor Pewakil

for temp\_a = 1:2

for class = 1:3

random\_value = (randsample(1:50,1));

Vpewakil = dataz(:, class+(random\_value-1)\*3);

Wn (:,class+(temp\_a-1)\*3)= Vpewakil;

end

end

n\_cluster = temp\_a \*class;

%alpha = 0.5;

%alpha = 0.7;

alpha = 0.3;

beta = 0.1\*alpha;

gamma = 0.1\*alpha;

m\_unit\_hidden = n\_cluster;

l\_unit\_output = class;

target = eye(3);

%% Find Parameter C dan S

C = Wn;

sn\_kuadrat = zeros(6,1);

% Mencari nilai standard deviasi data input per kelas

for i = 1:size(data\_train,2)

for j = 1:n\_cluster

sn\_kuadrat(j,:) = sn\_kuadrat(j,:) + sum((data\_train(:,i)-C(:,j)).^2);

end

end

sn = size(data\_train,2);

for j = 1:n\_cluster

S(j,:)= sqrt(sn\_kuadrat(j,:)/(sn-1));

end

%% SOM

for datake = 1:size(data\_train,2)

%Perhitungan Jarak Euclidean Input dengan masing-masing cluster

for n\_cluster = 1:6

ed\_cluster (:,n\_cluster) = sqrt(abs(sum(data\_train(:,datake)- Wn(:,n\_cluster)).^2));

end

% Mengambil Jarak Yang Dekat (clustering)

[~,vpvalue\_min] = min(ed\_cluster);

vpmatrix(:,datake) = vpvalue\_min;

% Update Vektor Pewakil

Wn(:,vpvalue\_min) = Wn(:,vpvalue\_min) + alpha\*(data\_train(:,datake)- Wn(:,vpvalue\_min));

end

1. Apabila kondisi henti (stopping condition) belum terpenuhi jalankan langkah 2-4
2. Melakukan proses propagasi maju (*feedforward*)
3. Setiap unit input menerima sinyal masukan dan menyebarkan sinyal tersebut ke lapisan tersembunyi (hidden layer)
4. Setiap unit hidden layer mengalami perhitungan terhadap vektor nilai tengah. Nilai didapatkan dengan menggunakan persamaan 3.2
5. Hasil perhitungan menjalankan fungsi aktivasi untuk menghitung sinyal keluaran unit hidden. Nilai Z didapatkan dengan menggunakan persamaan 3.3
6. Setiap unit keluaran menjumlahkan sinyal masukan yang telah diberi bobot. Nilai didapatkan dengan menggunakan persamaan 3.4
7. Setiap unit keluaran menjalankan fungsi aktivasi untuk menghitung sinyal keluaran unit output. Nilai Y didapatkan dengan menggunakan persamaan 3. 5

%% Learning Backward

etotal = 0;

epoch\_max = 10000;

epoch = 1;

*error*\_max = 0.01;

tic;

while (1)

[row,col]=size(data\_train);

Zin\_m = zeros();

*error* = 0;

% ========== Proses Feedforward ============== %%

% Perhitungan Keluaran Hidden Layer (Fungsi Gaussian)

for dataKe = 1:col

for m = 1:m\_unit\_hidden

% Zin\_m = sum(x-xbar)/2sigma^2

Zin\_m(m,:) = sum(((data\_train(:,dataKe)-C(:,m))).^2)/(2\*S(m)^2);

% Fungsi Gaussian Zm = phi ()

Zm = exp(-Zin\_m);

end

% Perhitungan Keluaran Output Layer

for l = 1:l\_unit\_output

Yin\_l (l,1) = W0l(l,1) + Zm'\*Wml(:,l);

end

Yl = 1./(1+exp(-Yin\_l));

output = Yl;

1. Melakukan propagasi balik (*backpropagation*)
2. Setiap unit keluaran menerima target yang bersesuaian dengan data masukan lalu menghitung kesalahan perhitungan dengan menggunakan persamaan berikut yang merujuk pada persamaan 3.8

(3.23)

1. Menghitung koreksi bobot dengan menambahkan konstanta laju pembelajaran koreksi bobot menggunakan persamaan berikut

(3.24)

(3.25)

Serta menghitung koreksi bias dengan persamaan

(3.26)

1. Menghitung koreksi nilai tengah , dengan menambahkan konstanta laju pembelajaran , menggunakan persamaan

(3.27)

1. Menghitung koreksi lebar data, dengan menambahkan konstanta laju pembelajaran , menggunakan persamaan

(3.28)

1. Memperbaharui nilai bobot, bias, nilai tengah dan lebar data dengan persamaan berikut

(3.29)

(3.30)

(3.31)

(3.32)

% Update Parameter Bobot, C dan S

Wml = Wml + delta\_wml;

W0l = W0l+ delta\_w0l;

C = C + delta\_c;

S = S + delta\_s;

end

epoch = epoch+1

etotal = *error*

clc

%Kondisi stopping epoch

if ((etotal<*error*\_max || (epoch>=epoch\_max) ))

break

end

end

time = toc

epoch

etotal

benar = 0;

% ============ Proses backforward ============== %%

beta = 0.1\*alpha;

gamma = 0.1\*alpha;

% Perhitungan *Error*

n\_target = mod(dataKe,3);

if n\_target == 0;

n\_target = 3;

end

*error* = *error* + 0.5\*(sum((output-target(:,n\_target)).^2,1));

*error*\_all(dataKe,:) = *error*;

% Perhitungan turunan fungsi *error* terhadap bobot

for l = 1:l\_unit\_output

dho(l,:) = (target(l,n\_target)-output(l,:))\*output(l,:)\*(1-output(l,:));

delta\_wml(:,l) = alpha\*dho(l,:)\*Zm;

end

delta\_w0l = alpha\*dho;

% Perhitungan turunan fungsi *error* terhadap Center

[dim,ncluster] = size (C); % [4,6]

for ncluster = 1:ncluster

for dim = 1:dim

delta\_c(dim,ncluster) = beta\*Wml(ncluster,:)\*dho\*-Zm(ncluster,:)\*-(sum((data\_train(dim,dataKe)-C(dim,ncluster)))/(2\*S(ncluster).^2));

end

end

% Perhitungan turunan fungsi *error* terhadap Std Deviasi

[ncluster, x] = size (S); % [4,6]

for ncluster = 1:ncluster

delta\_s(ncluster,:) = gamma\*Wml(ncluster,:)\*dho\*-Zm(ncluster,:)\*-(sum((data\_train(:,dataKe)-C(:,ncluster)).^2)/(2\*S(ncluster).^3));

end

1. Pengujian kondisi henti

Pembelajaran berhenti jika nilai *error* telah mencapai 0.01 atau kondisi epoch telah mencapai epoch maksimal.

1. Hasil Percobaan Jaringan Saraf Tiruan *Radial Basis Function* Metode Backward

Dalam melakukan percobaan Jaringan Saraf Tiruan *Radial Basis Function* dengan metode *Backward* digunakan suatu program yang telah dibuat dengan mengatur parameter-parameter yang dibutuhkan sehingga hasil percobaan dapat dianalisa pengaruhnya terhadap perubahan parameter yang ada. Percobaan dilakukan dengan Data Iris dengan kriteria percobaan sebagai berikut :

1. Data yang digunakan adalaha Data Iris yang diambil dari UCI Repository
2. Melakukan pengujian pengaruh alpha, beta dan gamma dengan hubungan alpha, alpha beta dan gamma adalah :

(3.33)

(3.34)

1. Terdapat tiga jenis pengujian pengaruh variasi nilai alpha pada *Recognition Rate* percobaan jaringan saraf tiruan Radial Basis Function. Nilai alpha yang diuji adalah pada nilai 0.5-0.7 untuk mengetahui kondisi parameter alpha yang optimal
2. Membandingkan pengujian dengan menggunakan klasifikasi SOM dengan C-Means untuk nilai alpha 0.5.

Dari percobaan yang dilakukan selama masing-masing 10 kali iterasi didapatkan hasil seperti berikut :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nilai Alpha | Iterasi | Epoch | Time | *Recognition Rate* |
| 0.5 | 1 | 4861 | 161.1305 | 96.67 |
| 0.5 | 2 | 5821 | 197.082 | 95.33 |
| 0.5 | 3 | 6321 | 222.3282 | 94.67 |
| 0.5 | 4 | 5581 | 201.049 | 94.67 |
| 0.5 | 5 | 5768 | 220.978 | 95.3 |
| 0.5 | 6 | 5869 | 221.521 | 94.67 |
| 0.5 | 7 | 5169 | 187.7546 | 94.67 |
| 0.5 | 8 | 4598 | 163.2517 | 94.67 |
| 0.5 | 9 | 5698 | 214.9115 | 96 |
| 0.5 | 10 | 6362 | 238.0106 | 94.67 |
| Rata-Rata | | 5604.8 | 202.8017 | 95.132 |

Tabel 3.3. Percobaan RBF Backward dengan Alpha = 0.5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nilai Alpha | Iterasi | Epoch | Time | *Recognition Rate* |
| 0.6 | 1 | 4858 | 169.7086 | 95.33 |
| 0.6 | 2 | 4959 | 172.6618 | 94.67 |
| 0.6 | 3 | 4167 | 142.0573 | 95.33 |
| 0.6 | 4 | 4052 | 135.0175 | 96 |
| 0.6 | 5 | 4671 | 159.5687 | 95.3 |
| 0.6 | 6 | 4910 | 174.8266 | 95.33 |
| 0.6 | 7 | 4648 | 161.0249 | 95.33 |
| 0.6 | 8 | 4727 | 171.3987 | 94.67 |
| 0.6 | 9 | 3699 | 129.7074 | 96 |
| 0.6 | 10 | 6362 | 147.6872 | 94.67 |
| Rata-Rata | | 4705.3 | 156.3659 | 95.263 |

Tabel 3.3. Percobaan RBF Backward dengan Alpha = 0.6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nilai Alpha | Iterasi | Epoch | Time | *Recognition Rate* |
| 0.7 | 1 | 2322 | 76.2833 | 96.67 |
| 0.7 | 2 | 2227 | 74.123 | 96 |
| 0.7 | 3 | 2269 | 71.8756 | 96 |
| 0.7 | 4 | 2365 | 76.078 | 96 |
| 0.7 | 5 | 2578 | 85.5391 | 96 |
| 0.7 | 6 | 2651 | 83.5683 | 96 |
| 0.7 | 7 | 2344 | 73.3281 | 96.67 |
| 0.7 | 8 | 2782 | 87.7747 | 96 |
| 0.7 | 9 | 2577 | 81.2732 | 96 |
| 0.7 | 10 | 2322 | 78.2693 | 96.67 |
| Rata-Rata | | 2443.7 | 78.81126 | 96.201 |

Tabel 3.3. Percobaan RBF Backward dengan Alpha = 0.7

Dari percobaan yang dilakukan terlihat bahwa nilai parameter alpha yang optimal adalah pada rentang 0.5 – 0.7 dengan nilai alpha optimal adalah 0.7

Sedangkan komparasi hasil pengujian metode klasifikasi SOM dengan C-Means untuk 10 kali iterasi pada nilai alpha = 0.5 adalah sebagai berikut :

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Alpha** | **Iterasi** | **SOM** | | | **C-Means** | | |
| Epoch | Time | RR | Epoch | Time | RR |
| 0.5 | 1 | 4861 | 161.1305 | 96.67 | 10000 | 92.6706 | 97.33333 |
| 0.5 | 2 | 5821 | 197.082 | 95.33 | 10000 | 85.4597 | 97.33333 |
| 0.5 | 3 | 6321 | 222.3282 | 94.67 | 10000 | 82.8034 | 97.33333 |
| 0.5 | 4 | 5581 | 201.049 | 94.67 | 10000 | 84.7768 | 97.33333 |
| 0.5 | 5 | 5768 | 220.978 | 95.3 | 10000 | 88.2285 | 97.33333 |
| 0.5 | 6 | 5869 | 221.521 | 94.67 | 10000 | 80.8064 | 97.33333 |
| 0.5 | 7 | 5169 | 187.7546 | 94.67 | 10000 | 79.3095 | 97.33333 |
| 0.5 | 8 | 4598 | 163.2517 | 94.67 | 10000 | 84.009 | 97.33333 |
| 0.5 | 9 | 5698 | 214.9115 | 96 | 10000 | 80.9548 | 97.33333 |
| 0.5 | 10 | 6362 | 238.0106 | 94.67 | 10000 | 84.8159 | 97.33333 |
| Rata-Rata | | 5604.8 | 202.8017 | 95.132 | 10000 | 84.38346 | 97.33333 |

Tabel 3.3. Perbandingan Pengujian RBF Backward Klasifikasi SOM Dengan Klasifikasi Cmeans Untuk Alpha = 0.5

Gambar 3.3 Grafik Perbandingan Recognition Rate Klasifikasi SOM Dengan C-Means Untuk Setiap Iterasi

Gambar 3.3 Grafik Perbandingan Cost Time RBF Klasifikasi SOM Dengan C-Means Untuk Setiap Iterasi

Gambar 3.3 Grafik Perbandingan Epoch RBF Klasifikasi SOM Dengan C-Means Untuk Setiap Iterasi

1. Analisis Percobaan Jaringan Saraf Tiruan *Radial Basis Function* Metode Backward

Pada percobaan kali ini, akan diamati performa dari jaringan saraf tiruan *Radial Basis Function* metode *Backward*. Data yang didapat adalah jumlah epoch selama pembelajaran, waktu pembelajaran, serta *Recognition Rate* dari pengujian data. Untuk nilai alpha yang berkisar antara 0.5-0.7, pembelajaran menunjukkan performa yang baik, dengan nilai *Recognition Rate* rata-rata pada alpha 0.5, 0.6 dan 0.7 masing-masing adalah 95.132%, 95.263% dan 96.201%. Namun, dapat dilihat bahwa waktu komputasi pembelajaran untuk ketiga nilai alpha ini juga tinggi dikarenakan proses komputasi yang akan menghitung kesalahan tiap data dan akan memperbaharui seluruh parameter pembelajarn untuk tiap data. Selain itu, karena adanya dua tahap pembelajaran yang berbeda membuat watu komputasi yang dibutuhkan juga akan tinggi.

Untuk percobaan yang dilakukan dengan alpha diluar rentang nilai alpha yang telah disebutkan diatas mempunyai hasil yang tidak cukup baik. Untuk nilai alpha < 0.5, hasil yang kurang baik terjadi dikarenakan perubahan nilai parameter yang terjadi sangat kecil sehingga tidak berpengaruh apa-apa pada komputasi pembelajaran. Di sisi lain, untuk nilai alpha > 0.7 hasil yang kurang baik terjadi karena ketidak telitian parameter pembelajaran yang menyebabkan kondisi kesalahan sulit menjadi konvergen.

Sedangkan untuk pengujian data untuk membandingkan metode klasifikasi SOM dan C-means dapat dilihat pada Grafik pada bagian 3.3. Metode klasifikasi C-Means akan menghasilkan perfoma yang lebih baik, walaupun epoch pembelajaran sangat besar dibandingkan dengan klasifikasi SOM, nilai cost time nya akan lebih kecil. Hal ini terjadi dikarenakan nilai parameter spread yang lebih terdefinisi jika menggunakan klasifikasi C-Means. Nilai *Recognition Rate* metode klasifikasi C-Means adalah 97,3333% , dan mempunyai *cost time* sebesar 84.38346s.

1. JARINGAN SARAF TIRUAN *RADIAL BASIS FUNCTION* DENGAN METODA OLS
2. Metode OLS pada Jaringan Saraf Tiruan *Radial Basis Function*

Jaringan saraf tiruan *Radial Basis Function* memiliki metode lain dalam melakukan pembelajaran, yaitu dengan metode Orthogonal Least Square. Dekomposisi orthogonal merupakan suatu metode yang biasa digunakan dalam penyelesaian fungsi least square. Fungsi least square itu sendiri biasanya digunakan dalam estimasi parameter suatu model untuk mendapatkan model yang menggambarkan keadaan asli suatu sistem.

Misalkan pada suatu kumpulan data pada suatu d-dimensi ruang masukan terdiri atas sebanyak N vektor masukan. Data tersebut akan dicocokkan dengan suatu nilai target pada sebuah ruang dimensi target menggunakan suatu fungsi terhadap , yaitu . Sehingga nilai dapat didefinisikan sebagai

(4.1)

Jika menggunakan fungsi Gaussian , yang merupakan fungsi jarak Euclidean, sebagai salah satu fungsi Radial pada jaringan saraf tiruan *Radial Basis Function*, maka fungsi dapat didefinisikan sebagai

(4.2)

Sehingga persamaan 4.1 dapat didefinisikan ulang ke dalam fungsi Gaussian

(4.3)

(4.4)

Persamaan 4.4 dapat disederhanakan menjadi

(4.5)

Karena fungsi merupakan matriks persegi dengan ukuran , maka persamaan 4.5 dapat diselesaikan dengan menghasilkan

(4.6)

Namun, jika matriks Gaussian mempunyai basis sehingga matriks Gaussian tidak lagi berupa matriks simetris, maka persamaan 4.5 dapat diselesaikan dengan menggunakan pseudo invers sebagai berikut

(4.7)

(4.8)

(4.9)

Dimana adalah pseudo invers dari .

Setelah mendapatkan nilai bobot w dengan menggunakan metode OLS, nilai keluaran yang masih memiliki nilai kesalahan akan dioptimasi sehingga nilai kesalahan atau *error* yang dihasilkan akan semakin kecil, dan keluaran pun akan menjadi lebih baik. Proses optimasi nilai kesalahan akan dilakukan dengan propagasi maju (*feedforward*) dan propagasi balik (*backpropagation*)

Pada propagasi maju, seperti yang sudah dijelaskan pada bagian 3. 1, data masukan yang masuk akan dihitung menggunakan fungsi Gaussian.

(4.10)

(4.11)

Keluaran dari lapisan tersembunyi, Z, akan mengalami perhitungan dengan pengaruh dari bias serta bobot antara lapisan tersembunyi dengan lapisan keluaran. Keluaran pada lapisan keluaran merupakan hasil perhitungan fungsi sigmoid unipolar.

(4.12)

(4.13)

Proses perhitungan nilai keluaran Y disebut dengan propagasi maju atau *feedforward*. Setelah mengetahui nilai keluaran proses *feedforward* ini akan dihitung besarnya kesalahan yang terjadi pada keluaran terhadap nilai target yang bersesuaian.

(4.14)

Proses yang terjadi selanjutnya adalah proses propagasi balik atau *backpropagation*, yang akan mengoreksi nilai bobot dan bias dari lapisan tersembunyi ke lapisan keluaran. Selain itu, juga terjadi koreksi nilai tengah dan lebar data (parameter *spread*). Koreksi nilai-nilai tersebut didapatkan dari penurunan rumus kesalahan perhitungan (*error*) terhadap komponen koreksi. Nilai koreksi tersebut adalah :

1. Koreksi nilai bobot

Koreksi nilai bobot akan didapatkan dari turunan rumus kesalahan perhitungan (*error*) terhadap nilai bobot

(4.15)

Dimana,

(4.16)

(4.17)

Sehingga persamaan 4.15 menjadi,

(4.18)

1. Koreksi nilai tengah ()

Koreksi nilai tengah akan didapatkan dari turunan rumus kesalahan perhitungan (*error*) terhadap nilai tengah

(4.19)

Dimana,

(4.20)

(4.21)

(4.22)

(4.23)

Sehingga persamaan 4.19 menjadi,

(4.24)

1. Koreksi lebar data atau parameter *spread* ()

Koreksi lebar data akan didapatkan dari turunan rumus kesalahan perhitungan (*error*) terhadap lebar data

(4.25)

Dimana,

(4.26)

(4.27)

(4.28)

(4.29)

Sehingga persamaan 4.25 menjadi,

(4.30)

1. Algoritma dan Program Percobaan Jaringan Saraf Tiruan *Radial Basis Function* Metode OLS

Algoritma yang digunakan dalam melakukan pembelajaran Jaringan Saraf Tiruan *Radial Basis Function* dengan metode OLS adalah sebagai berikut:

1. Inisialisasi
2. Menentukan jumlah M fungsi basis yang dibutuhkan yang jumlahnya tidak sama dengan N jumlah data utama, dimana M< N
3. Menentukan nilai tengah fungsi basis
4. Menentukan nilai standar deviasi sebagai parameter *spread* atau lebar data
5. Memasukkkan parameter bias ke dalam penjumlahan linier
6. Mencari bobot fungsi linier

Terdapat 2 kondisi untuk menentukan bobot fungsi linier, yaitu dengan menggunakan fungsi aktivasi dan tidak menggunakan fungsi aktivasi. Bobot dicari dengan menggunakan persamaan 4.9, yaitu

(4.9)

Dimana

(4.31)

Untuk pembelajaran yang tidak menggunakan fungsi aktivasi, serta

(4.32)

Untuk pembelajaran yang menggunakan fungsi aktivasi.

%% Vektor Pewakil

for temp\_a = 1:2

for class = 1:3

random\_value = (randsample(1:50,1));

Vpewakil = dataz(:, class+(random\_value-1)\*3);

Wn (:,class+(temp\_a-1)\*3)= Vpewakil;

end

end

n\_cluster = temp\_a \*class;

alpha = 0.5;

beta = 0.1\*alpha;

gamma = 0.1\*alpha;

m\_unit\_hidden = n\_cluster;

l\_unit\_output = class;

%% Find Parameter C dan S

C = Wn;

sn\_kuadrat = zeros(m\_unit\_hidden,1);

% Mencari nilai standard deviasi data input per kelas

for i = 1:size(data\_train,2)

for j = 1:n\_cluster

sn\_kuadrat(j,:) = sn\_kuadrat(j,:) + sum((data\_train(:,i)-C(:,j)).^2);

end

end

sn = size(data\_train,2);

for j = 1:n\_cluster

S(j,:)= sqrt(sn\_kuadrat(j,:)/(sn-1));

end

%% SOM

for datake = 1:size(data\_train,2)

%Perhitungan Jarak Euclidean Input dengan masing-masing cluster

for n\_cluster = 1:m\_unit\_hidden

ed\_cluster (:,n\_cluster) = sqrt(abs(sum(data\_train(:,datake)- Wn(:,n\_cluster)).^2));

end

% Mengambil Jarak Yang Dekat (clustering)

[~,vpvalue\_min] = min(ed\_cluster);

vpmatrix(:,datake) = vpvalue\_min;

% Update Vektor Pewakil

Wn(:,vpvalue\_min) = Wn(:,vpvalue\_min) + alpha\*(data\_train(:,datake)- Wn(:,vpvalue\_min));

end

%% Membentuk Matriks Target

b = [1e-10 1e-10 1e-10; 1e-10 1e-10 1e-10; 1e-10 1e-10 1e-10];

% b = [1 1e-10 1e-10; 1e-10 1 1e-10; 1e-10 1e-10 1];

for a = 1:3

for n = 1:25

t((a+(n-1)\*3),:) = b (a,:);

end

end

% tin = -log((1-t)./t);

tin = t;

%% Inisialisasi Bobot dengan OLS

% Matriks Gaussian

col = size(data\_train,2);

for i = 1:m\_unit\_hidden

for a = 1:col

phi (a,i) = exp(-sum(((data\_train(:,a))-C(:,i)).^2/(2\*S(i).^2)));

end

end

phi(:,m\_unit\_hidden+1)= 1;

D = tin;

Wml = inv(phi'\*phi)\*phi'\*D;

target = [1 1e-10 1e-10; 1e-10 1 1e-10; 1e-10 1e-10 1];

% target = eye(3);

1. Apabila kondisi henti belum terpenuhi, jalankan langkah 2-4
2. Melakukan proses propagasi maju (*feedforward*)
3. Setiap unit input menerima sinyal masukan dan menyebarkan sinyal tersebut ke lapisan tersembunyi (hidden layer)
4. Setiap unit hidden layer mengakami perhitungan terhadap vektor nilai tengah. Nilai didapatkan dengan menggunakan persamaan 4.10
5. Hasil perhitungan menjalankan fungsi aktivasi untuk menghitung sinyal keluaran unit hidden. Nilai Z didapatkan dengan menggunakan persamaan 4.11
6. Setiap unit keluaran menjumlahkan sinyal masukan yang telah diberi bobot. Nilai didapatkan dengan menggunakan persamaan 4.12
7. Setiap unit keluaran menjalankan fungsi aktivasi untuk menghitung sinyal keluaran unit output. Nilai Y didapatkan dengan menggunakan persamaan 4.13

while(1)

[row, col] = size (data\_train);

Zin\_m = zeros();

*error* = 0;

for dataKe = 1:col

for m = 1:m\_unit\_hidden

Zin\_m(m,:) = sum(((data\_train(:,dataKe)-C(:,m))).^2)/(2\*S(m)^2);

end

Zm = exp(-Zin\_m);

Zm\_all(:,dataKe) = Zm;

for l = 1: l\_unit\_output

Yin\_l(l,1) = Zm'\*Wml(:,l);

end

Yl = 1./(1+exp(-Yin\_l));

output = Yl;

output\_all (:,dataKe) = output;

1. Melakukan proses propagasi balik
2. Setiap unit keluaran menerima target yang bersesuaian dengan data masukan lalu menghitung kesalahan perhitungan dengan menggunakan persamaan berikut yang merujuk pada persamaan 3.8

(4.33)

1. Menghitung koreksi bobot dengan menambahkan konstanta laju pembelajaran koreksi bobot menggunakan persamaan berikut

(4.34)

(4.35)

Serta menghitung koreksi bias dengan persamaan

(4.36)

1. Menghitung koreksi nilai tengah , dengan menambahkan konstanta laju pembelajaran , menggunakan persamaan

(4.37)

1. Menghitung koreksi lebar data, dengan menambahkan konstanta laju pembelajaran , menggunakan persamaan

(4.38)

1. Memperbaharui nilai bobot, bias, nilai tengah dan lebar data dengan persamaan berikut

(4.39)

(4.40)

(4.41)

(4.42)

*error* = 0.5\*(sum((output-target(:,n\_target)).^2,1));

*error*\_all(dataKe,:) = *error*;

*error*\_mean = mean(*error*\_all);

*error*\_semua(dataKe,epoch) = *error*;

for l = 1:l\_unit\_output

dho(l,:) = (target(l,n\_target)-output(l,:))\*output(l,:)\*(1-output(l,:));

end

dho\_all(:,dataKe) = dho;

dho\_mean = mean(dho\_all, 2);

% end

% dho = dho\_mean;

% Zm = mean(Zm\_all,2);

%% Update

for l = 1:l\_unit\_output

delta\_wml(:,l) = alpha\*dho(l,:)\*Zm;

end

[dim, ncluster] = size(C);

for ncluster = 1:ncluster

for dim = 1:dim

delta\_c(dim,ncluster)= beta\*Wml(ncluster,:)\*dho-Zm(ncluster,:)\*-(sum((data\_train(dim,dataKe)-C(dim,ncluster)))/(2\*S(ncluster).^2));

end

end

[ncluster, x] = size (S); % [4,6]

for ncluster = 1:ncluster

delta\_s(ncluster,:) = gamma\*Wml(ncluster,:)\*dho\*-Zm(ncluster,:)\*-(sum((data\_train(:,dataKe)-C(:,ncluster)).^2)/(2\*S(ncluster).^3));

end

% Update Parameter Bobot, C dan S

Wml = Wml + delta\_wml;

C = C + delta\_c;

S = S + delta\_s;

end

epoch = epoch+1

etotal = *error*\_mean

clc

if ((etotal<*error*\_max || (epoch>=epoch\_max) ))

break

end

end

time = toc

epoch

etotal

1. Pengujian kondisi henti

Proses pembelajaran akan berhenti jika nilai *error* telah mencapai 0.01 atau kondisi epoch telah mencapai nilai epoch maksimum

1. Hasil Percobaan Jaringan Saraf Tiruan *Radial Basis Function* Metode OLS

Dalam melakukan percobaan Jaringan Saraf Tiruan *Radial Basis Function* dengan metode OLS digunakan suatu program yang telah dibuat dengan mengatur parameter-parameter yang dibutuhkan sehingga hasil percobaan dapat dianalisa pengaruhnya terhadap perubahan parameter yang ada. Percobaan dilakukan dengan Data Iris dengan kriteria percobaan sebagai berikut :

1. Data yang digunakan adalaha Data Iris yang diambil dari UCI Repository
2. Melakukan pengujian pengaruh alpha, beta dan gamma dengan hubungan alpha, alpha beta dan gamma adalah :

(4.43)

(4.44)

1. Terdapat dua jenis pengujian pengaruh bias pada komputasi jaringan saraf tiruan Radial Basis Function, dengan tidak menggunakan fungsi aktivasi. Nilai parameter alpha yang digunakan adalah 0.5
2. Melakukan pengujian RBF OLS metode klasifikasi SOM dengan metode klasifikasi C-Means untuk alpha 0.5 dan tidak menggunakan fungsi aktivasi.

Dari percobaan yang dilakukan selama masing-masing 10 kali iterasi didapatkan hasil seperti berikut :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Alpha | Iterasi | Epoch | Time | Error | *Recognition Rate* |
| 0.5 | 1 | 10000 | 336.1934 | 0.1876 | 60.677 |
| 0.5 | 2 | 10000 | 323.2964 | 0.1876 | 60.677 |
| 0.5 | 3 | 10000 | 323.9512 | 0.1876 | 60.677 |
| 0.5 | 4 | 10000 | 321.0071 | 0.1876 | 60.6677 |
| 0.5 | 5 | 10000 | 319.7646 | 0.1876 | 60.6677 |
| 0.5 | 6 | 10000 | 320.8423 | 0.1891 | 60.6667 |
| 0.5 | 7 | 10000 | 330.0585 | 0.1876 | 60.677 |
| 0.5 | 8 | 10000 | 328.529 | 0.1876 | 60.677 |
| 0.5 | 9 | 10000 | 322.4048 | 0.1876 | 60.677 |
| 0.5 | 10 | 10000 | 324.9407 | 0.1876 | 60.677 |
| Rata Rata | | 10000 | 325.0988 | 0.18775 | 60.67411 |

Tabel 4.3. Percobaan RBF OLS Tanpa Fungsi Aktivasi Dan Tanpa Bias

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Alpha | Iterasi | Epoch | Time | Error | *Recognition Rate* |
| 0.5 | 1 | 10000 | 309.5168 | 0.1888 | 59.333 |
| 0.5 | 2 | 10000 | 320.1496 | 0.1881 | 59.333 |
| 0.5 | 3 | 10000 | 314.6714 | 0.1881 | 59.33 |
| 0.5 | 4 | 10000 | 326.8341 | 0.1882 | 59.333 |
| 0.5 | 5 | 10000 | 326.0.255 | 0.1882 | 59.33 |
| 0.5 | 6 | 10000 | 325.252 | 0.1884 | 59.33 |
| 0.5 | 7 | 10000 | 308.266 | 0.1885 | 60.667 |
| 0.5 | 8 | 10000 | 304.4954 | 0.1886 | 58.667 |
| 0.5 | 9 | 10000 | 305.4741 | 0.1882 | 59.333 |
| 0.5 | 10 | 10000 | 305.0632 | 0.1883 | 59.333 |
| Rata Rata | | 10000 | 281.9723 | 0.18834 | 59.3989 |

Tabel 4.3. Percobaan RBF OLS Tanpa Fungsi Aktivasi dengan Bias

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Alpha | Iterasi | Epoch | Time | Error | *Recognition Rate* |
| 0.5 | 1 | 10000 | 238.6582 | 0.2297 | 57.333 |
| 0.5 | 2 | 10000 | 267.7918 | 0.2297 | 57.333 |
| 0.5 | 3 | 10000 | 288.2163 | 0.2297 | 57.333 |
| 0.5 | 4 | 10000 | 227.4143 | 0.2297 | 57.333 |
| 0.5 | 5 | 10000 | 264.9677 | 0.2297 | 57.333 |
| 0.5 | 6 | 10000 | 260.8236 | 0.2297 | 57.333 |
| 0.5 | 7 | 10000 | 266.665 | 0.2297 | 57.333 |
| 0.5 | 8 | 10000 | 252.6223 | 0.2297 | 57.333 |
| 0.5 | 9 | 10000 | 247.4215 | 0.2297 | 57.333 |
| 0.5 | 10 | 10000 | 243.4564 | 0.2297 | 57.333 |
| Rata Rata | | 10000 | 255.8037 | 0.2297 | 57.333 |

Tabel 4.3. Percobaan RBF OLS Menggunakan Fungsi Aktivasi

Sedangkan untuk hasil pengujian RBF OLS metode klasifikasi SOM dengan metode klasifikasi C-Means untuk alpha 0.5 dan tidak menggunakan fungsi aktivasi adalah sebagai berikut :

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Alpha** | **Iterasi** | **SOM** | | | **C-Means** | | |
| Epoch | Time | RR | Epoch | Time | RR |
| 0.5 | 1 | 10000 | 309.5168 | 59.333 | 10000 | 76.2051 | 97.33333 |
| 0.5 | 2 | 10000 | 320.1496 | 59.333 | 10000 | 80.7157 | 97.33333 |
| 0.5 | 3 | 10000 | 314.6714 | 59.33 | 10000 | 86.2743 | 97.33333 |
| 0.5 | 4 | 10000 | 326.8341 | 59.333 | 10000 | 88.3925 | 97.33333 |
| 0.5 | 5 | 10000 | 326.0.255 | 59.33 | 10000 | 94.5402 | 97.33333 |
| 0.5 | 6 | 10000 | 325.252 | 59.33 | 10000 | 92.0037 | 97.33333 |
| 0.5 | 7 | 10000 | 308.266 | 60.667 | 10000 | 94.7769 | 97.33333 |
| 0.5 | 8 | 10000 | 304.4954 | 58.667 | 10000 | 87.93 | 97.33333 |
| 0.5 | 9 | 10000 | 305.4741 | 59.333 | 10000 | 93.48 | 97.33333 |
| 0.5 | 10 | 10000 | 305.0632 | 59.333 | 10000 | 95.0435 | 97.33333 |
| Rata-Rata | | 10000 | 281.9723 | 59.3989 | 10000 | 88.93619 | 97.33333 |

Tabel 4.3. Perbandingan RBF OLS Klasisfikasi SOM Dengan Klasifikasi C-Means Untuk Kondisi Nilai Alpha 0.5, Tidak Menggunakan Fungsi Aktivasi Dan Menggunakan Bobot Bias

Gambar 4.3 Grafik Perbandingan Recognition Rate RBF NN Klasifikasi SOM Dengan Cmeans

Gambar 4.3 Grafik Perbandingan Cost Time Rbf Nn Klasifikasi Som Dengan Cmeans

1. Analisis Percobaan Jaringan Saraf Tiruan *Radial Basis Function* Metode OLS

Pada percobaan kali ini, akan diamati performa jaringan saraf tiruan metode Orthiginal Least Square. Data yang didapat berupa waktu dan banyak epoch selama pembelajaran, nilai error dan nilai *Recognition Rate*. Ada 4 percobaan yang dilakukan, percobaan dengan fungsi serta tanpa fungsi aktivasi. Dan bagi masing masing fungsi juga akan dilakukan percobaan dengan menggunakan bobot bias dan tanpa bobot bias.

Pada percobaan RBF OLS tanpa fungsi aktivasi dengan tanpa menggunakan bias, rata-rata nilai *Recognition Rate* yang didapat adalah 60.67411%, dengan error 0.18775 selama waktu komputasi 325.0988 s. Sedangkan percobaan RBF OLS tanpa fungsi aktivasi dengan menggunakan bias, rata-rata nilai *Recognition Rate* yang didapat adalah 59.3989 dengan error komputasi 0.18834 selama waktu komputasi 281.9723s.

Percobaan dengan menggunakan bias akan berlangsung pada waktu komputasi yang singkat, dikarenakan terdapat koreksi tambahan yaitu pada bobot bias, sehingga percobaan bisa dioptimasi. Sedangkan percobaan tanpa menggunakan bias akan mengalami lebih sedikit kenaikan *Recognition Rate* dengan error komputasi yang juga sedikit lebih baik. Dalam hal ini, secara empiris keberadaan bias pada komputasi pembelajaran akan sedikit mempengaruhi performa pembelajaran. Tetapi untuk data dan sistem yang membutuhkan keteletian dibandingkan kecepatan, penggunaan bobot bias pada jaringan saraf tiruan, terutama *Radial Basis Function* akan sangat penting dalam kualitas performa jaringan.

Pada percobaan RBF OLS dengan fungsi aktivasi yang dilakukan dengan menggunakan bias, rata-rata nilai *Recognition Rate* yang didapat adalah 57.3333%, dengan error 0.2297 selama waktu komputasi 255.80371s.

Selain pengujian yang dipaparkan sebelumnya, terdapat perbandingan pengujian RBF OLS menggunakan klasifikasi SOM dengan menggunakan klasifikasi C-means. Dapat dilihat pada grafik yang ditunjukkan oleh Gambar 4.3.1 dan Gambar 4.3.2 penggunaan klasifikasi C-Means akan menunjukkan performa yang lebih baik dari segi *Recognition Rate* dan *Cost Time*, dengan rata-rata Recognition Rate metode klasifikasi C-Means adalah 97.33333% dan mempunyai *cost time* 88.93619s.

1. KOMPARASI JARINGAN SARAF TIRUAN ALGORITMA *RADIAL BASIS FUNCTION* DENGAN ALGORITMA *BACKPROPAGATION*

Bab ini akan membahas mengenai perbandingan antara jaringan saraf tiruan *Radial Basis Function* dengan metode klustering SOM dengan jaringan saraf tiruan *Backpropagation*. Dalam melakukan perbandingan, seluruh parameter yang digunakan akan disamakan. Perbandingan ini akan melihat jumlah epoch, waktu pembelajaran serta *Recognition Rate* pengujian.

Spesifikasi alat dan *software* simulasi yang digunakan dalam penelitian ini adalah

Tipe : Laptop

Operating System : Windows 7 Professional 64-bit

Processor : Intel(R) Core(TM) i5-2450M @ 2.5 Ghz

Memory : 4.00GB RAM

Software : MATLAB (R2013a)

1. Komparasi Hasil Percobaan Jaringan Saraf Tiruan Algoritma *Radial Basis Function* Metode Backforward, Hasil Percobaan Jaringan Saraf Tiruan Algoritma *Radial Basis Function* Metode OLS, dan Hasil Percobaan Jaringan Saraf Tiruan Algoritma *Backpropagation*

Karena pada saat melakukan percobaan parameter yang digunakan berbeda, maka dalam melakukan perbandingan ketiga metode ini diambil nilai rata-rata hasil pembelajaran yang sudah dipaparkan sebelumnya.

Untuk pembelajaran dengan metode *Radial Basis Function* dimabil nilai rata-rata untuk ketiga percobaan dengan alpha yang berbeda. Begitu pula dengan pembelajaran metode RBF *Orthogonal Least Square*, data yang digunakan adalah data rata-rata pembelajaran RBF OLS tanpa menggunakan fungsi aktivasi, dengan alpha 0.5 sebagai parameter universal penelitian perbandingan ini. Terdapat dua jenis pengujian untuk metode RBF yaitu dengan menggunakan klasifikasi SOM dan klasifikasi C-Means. Untuk pembelajaran dengan metode *Backpropagation* menggunakan alpha 0.5, nilai momentum 0.5 untuk pembelajaran data sebanyak 50% dari jumlah data. Dari data yang didapat akan diambil nilai rata-ratanya untuk nantinya dibandingkan dengan 2 metode lain yang sudah disebutkan.

Hasil perbandingan masing-masing metode jaringan saraf tiruan diperlihatkan pada grafik berikut :

Gambar 5.1. Grafik Perbandingan Waktu Komputasi Jaringan Saraf Tiruan

Gambar 5.1. Grafik Perbandingan Recognition Rate Jaringan Saraf Tiruan

Gambar 5.1. Grafik Perbandingan Jumlah Epoch Jaringan Saraf Tiruan

Gambar 5.1. Grafik Perbandingan Error Jaringan Saraf Tiruan

1. Analisis Komparasi Hasil Percobaan Jaringan Saraf Tiruan Algoritma *Radial Basis Function* Metode Backforward, Hasil Percobaan Jaringan Saraf Tiruan Algoritma *Radial Basis Function* Metode OLS, dan Hasil Percobaan Jaringan Saraf Tiruan Algoritma *Backpropagation*

Dari hasil perbandingan yang disajikan dalam bentuk grafik pada bagian 5.1, masing-masing metode jaringan saraf tiruan mempunyai kelebihan dan kekurangan masing-masing.

Waktu komputasi *Backpropagation* jauh lebih kecil jika dibandingkan dengan RBF *Backward* dan RBF OLS sendiri. Hal ini terjadi karena pada komputasi RBF terdapat dua tahap pertama klasifikasi data dan kemudian komputasi data yang sama-sama melibatkan perhitungan data yang besar sekaligus. Data hasil klasifikasi SOM merupakan data nilai tengah kluster, dan seperti yang diketahui, tingkat akurasi SOM cukup kecil sehingga diperlukan perbaikan atau optimasi nilai error yang membutuhkan waktu lama pada proses komputasi feedforward dan backpropagation pada jaringan saraf tiruan metode *Radial Basis Function*. Hal ini juga berlaku untuk data epoch hasil komputasi masing-masing metode jaringan saraf tiruan.

Pada perhitungan error, metode RBF *Backward* dan *Backpropagation* dapat mencapai kondisi konvergen, sedangkan metode RBF OLS tidak, baik dengna menggunakan klasifikasi SOM maupun C-Means. Hal ini terjadi karena perhitungan error pada metode RBF OLS merupakan nilai rata-rata error masing-masing data, bukan pejumlahan nilai error. Sehingga nilai error pada metode ini akan berkisar pada suatu nilai yang merupakan nilai rata-ratanya, walaupun masih ada kemungkinan nilai rata-rata error komputasi sangat kecil sehingga bisa mencapai kondisi konvergen.

Hasil *Recognition Rate* dari masing-masing metode juga berbeda. RBF OLS mempunyai tingkat *Recognition Rate* yang paling kecil dengan nilai 59.1353%. sedangkan metode *Backpropagation* memiliki performa cukup baik dengan tingkat *Recognition Rate* 92.137%. Sedangkan metode RBF *Backward* mempunyai tingkat akurasi yang paling tinggi yaitu dengan *Recognition Rate* 95.532%. Dari nilai-nilai ini, metode RBF *Backward* mampu melakukan komputasi dengan tingkat akurasi yang tinggi, namun dengan waktu yang lebih lama. Maka solusi dari persamaan ini adalah dengan menggunakan klasfikasi C-Means dyang mempunyai Recognition Rate 97.333% untuk waktu komputasi 84.383 – 88.93 s.

# KESIMPULAN

Berdasarkan rangkaian penelitian yang dimulai dari studi literatur, pengembangan program, percobaan dan pengujian data serta analisis yang telah dilakukan, maka dapat ditarik beberapa kesimpulan, yaitu :

1. Pemilihan algoritma dan metode pengolahan data untuk pembelajaran yang digunakan pada percobaan berpengaruh pada performa jaringan saraf tiruan Radial Basis Function.
2. Parameter yang mempengaruhi *Recognition Rate* pada proses pembelajaran dengan menggunakan jaringan saraf tiruan Radial Basis Function adalah nilai tengah data, lebar data, konstanta laju pembelajaran serta metode yang digunakan dalam pembelajaran.
3. Jaringan saraf Tiruan dengan metode OLS tanpa menggunakan fungsi aktivasi mempunyai performa yang lebih baik jika dibandingkan dengan algoritma yang menggunakan fungsi aktivasi
4. Pada data Iris, jaringan saraf tiruan *Radial Basis Function* dengan metode Backward memiliki performa yang lebih baik dibandingkan dengan jaringan saraf tiruan *Radial Basis Function* dengan metode OLS.
5. Jika dibandingkan dengan jaringan saraf tiruan *Backpropagation,* secara umum metode *Radial Basis Function* dengan metode klasifikasi SOM masih mempunyai beberapa kekurangan pada performa yang digunakan. Tetapi, akurasi pada pembelajaran *Radial Basis Function* dengan metode *Backward* lebih baik dibandingkan metode *Backpropagation*. Pengembangan algoritma klasifikasi atau klustering pada *Radial Basis Function* dapat dikembangkan lebih lanjut.
6. Penggunaan metode klasifikasi atau klustering C-Means menunjukkan performa yang lebih baik dibandingkan dengan menggunakan SOM.

# DAFTAR PUSTAKA

1. F.L Lewis, S. Jagannathan, A. Yeilderek, (1999). *Neural Network Control of Robot Manipulators and Nonlinear Systems*. Taylor&Francis, West Sussex
2. Faqih, A. (2013). *Studi Karakteristik Jaringan Saraf Tiruan Radial Basis Function Menggunakan Metode Backward dan Metode Orthogonal Least Square*. Depok : Universitas Indonesia
3. Kusumoputro, B etal. (2013).*Quality Classification of Green Pellet Nuclear Fuels using Radial Basis Function Neural Networks*. IEEE Energy Conference, Miami, USA.
4. Liu, J. (2013). *Radial Basis Function Neural Network Control for Mechanical Systems*. Tsinghua University Press and Spriger-Verlag Berlin Heidelberg, New York.
5. Wen Chen, Zhuo-Jia Fu and C.S Chen. (2014) *Recent Advances in Radial Basis Function Collocation Methods*. Springer, London.
6. Buhman, M. (2003). *Radial Basis Functions*.Cambridge University Press, Cambridge, UK.
7. N. Sudararajan, P. Saratchandran and Yan Li, (2002). F*ully Tuned Radial Basis Function Neural Network for Flight Control*. Springer Science+Bussiness Media, New York
8. R.J Howlet and L.C Jain. (2001). *Radial Basis Functions 2, New Advances in Design*. Spriger-Verlag Berlin Heidelberg, New York.
9. Kusumoputro, Benyamin. (2001). *Jaringan Neural Buatan : Bahan Belajar Mahasiswa*. Fakultas Ilmu Komputer Universitas Indonesia, Depok : Universitas Indonesia.
10. Sudarno Putra, D. (2011).*Pengembangan Jaringan Saraf Tiruan Dengan Metode SOM Fuzzy dan LVQ Fuzzy*. Depok : Universitas Indonesia
11. Ai-Junkie, *Kohonen’s Self Organizing Feature Maps* [Online] Available : http://www.ai-junkie.com/ann/som/som1.html [Accessed 22 12 13]