

# ALGORITMA CONJUGATE GRADIENT POLAK RIBIERE UNTUK PENINGKATAN PERFORMANSI BACKPROPAGATION PADA SISTEM PREDIKSI TEMPERATUR UDARA

Untari Novia Wisesty<sup>1</sup>, Adiwijaya<sup>2</sup>, Tjokorda Agung BW<sup>3</sup>

<sup>1,3</sup> Fakultas Informatika – Institut Teknologi Telkom, Bandung

<sup>2</sup> Fakultas Sains – Institut Teknologi Telkom, Bandung  
[1untarimail@gmail.com](mailto:1untarimail@gmail.com), [2adw@ittelkom.ac.id](mailto:2adw@ittelkom.ac.id), [3cok@ittelkom.ac.id](mailto:3cok@ittelkom.ac.id)

---

## Abstrak

*Conjugate Gradient* merupakan suatu metode optimasi yang dapat meminimasi suatu fungsi, dimana arah pencariannya berdasarkan arah konjugasi yang nilainya ortogonal. Karena sifat pencariannya yang ortogonal, sehingga *Conjugate Gradient* cepat mencapai konvergensi pada solusi yang dicari. *Conjugate Gradient* tidak hanya dapat digunakan untuk menyelesaikan fungsi yang linier, tetapi juga dapat digunakan untuk masalah non linier, salah satunya digunakan untuk pelatihan jaringan syaraf tiruan. Pada makalah ini, digunakan algoritma *Conjugate Gradient Polak Ribiere* dalam pelatihan jaringan syaraf tiruan sebagai system peramalan temperature udara. Data klimatologi yang digunakan sebagai parameter masukan yaitu temperature udara, kelembaban, tekanan udara, curah hujan, lama penyinaran matahari, dan kecepatan angin. Berdasarkan hasil penelitian ini, kita dapat menginterpretasikan bahwa sistem peramalan temperatur udara ini menghasilkan akurasi lebih dari 90%.

**Kata kunci :** *Conjugate Gradient Polak Ribiere*, temperatur udara, klimatologi, pelatihan JST.

---

## Abstract

*Conjugate Gradient* is an optimization method that minimize a function, where the search directions are constructed by its conjugation which orthogonal values. Due to the orthogonal direction, *Conjugate Gradient* can be convergence to the solution. *Conjugate Gradient* is used not only for solving a linier function, but also non linier problems, such as Artificial Neural Network training.

*In this paper, Conjugate Gradient Polak Ribiere algorithm is used in Artificial Neural Network training as air temperature forecasting system. Climatology data that used as input variables are air temperature, humidity, air pressure, rainfall, sunshine radiation, and wind velocity. Based on the research, we can interpret that forecasting system of air temperature produce an accuracy more than 90%.*

**Keyword :** *Conjugate Gradient Polak Ribiere*, air temperature, climatology, Artificial Neural Network training.

---

## 1. Pendahuluan

Dalam bidang *Artificial Intelligence*, terdapat sub bidang yang disebut jaringan syaraf tiruan. Dalam perkembangannya, jaringan syaraf tiruan sempat ditinggalkan karena tidak dapat menyelesaikan permasalahan yang kompleks, sampai ditemukan Jaringan Syaraf Tiruan *Backpropagation* yang terdiri dari multilayer. *Backpropagation* dapat menyelesaikan beberapa kasus, misalkan pengenalan pola, peramalan, klasifikasi, dan lain-lain. Tetapi algoritma *Backpropagation* standar juga menemui beberapa kendala, yaitu lamanya pelatihan apabila pemilihan parameter-parameternya tidak tepat, misalkan pemilihan learning rate yang berpengaruh pada perubahan bobot. Beberapa kendala tersebut telah banyak ditangani dengan metode adaptif learning rate, tetapi masih belum mencapai konvergensi dalam waktu yang cepat [1, 3, 7]. Oleh karena itu, dibutuhkan algoritma yang dapat mempercepat pelatihan *Backpropagation*, salah satunya algoritma *Conjugate Gradient Polak Ribiere*.

*Conjugate Gradient* merupakan metode optimasi yang dapat meminimasi suatu fungsi, dimana arah pencariannya berdasarkan arah konjugasi yang nilainya ortogonal. Karena sifat pencariannya yang ortogonal, sehingga *Conjugate Gradient* cepat mencapai konvergensi pada solusi yang dicari. *Conjugate Gradient* tidak hanya dapat digunakan untuk menyelesaikan fungsi yang linier, tetapi juga dapat digunakan untuk masalah non linier, seperti *design engineering*, pelatihan Jaringan syaraf tiruan, financial time series forecasting, dan regresi nonlinier [8]. Terdapat beberapa jenis dari metode *Conjugate Gradient*, salah satunya yaitu *Polak Ribiere* [9, 10]. Untuk beberapa kasus, *Polak Ribiere* dapat menemukan solusi meskipun titik awalnya jauh dari titik minimum dan dapat konvergen lebih cepat dari yang lainnya. Pada makalah ini diimplementasikan algoritma *Conjugate Gradient Polak Ribiere* sebagai algoritma pelatihan *Backpropagation* pada studi kasus peramalan temperatur udara.

Studi kasus peramalan temperatur ini telah diteliti dan dibangun menggunakan metode-metode lainnya [4], salah satunya dengan metode Bayesian Network, tetapi dengan menggunakan algoritma Conjugate Gradient Polak Rieber sebagai algoritma pelatihan Backpropagation termodifikasi diharapkan akan didapat performansi sistem yang lebih baik.

## 2. Algoritma Conjugate Gradient Polak Ribiere untuk Pelatihan Jaringan Syaraf Tiruan (JST)

Metode *conjugate gradient* (CG) merupakan algoritma pencarian yang arah pencariannya berdasarkan pada arah conjugasinya. Secara umum algoritma ini lebih cepat convergen daripada metode penurunan tercepat. CG menggunakan vector tidak nol yang orthogonal dan bebas linier [5, 8, 9]. Dua vector  $d_i$  dan  $d_j$  dikatakan orthogonal (G-conjugate) jika perkalian dalamnya bernilai nol, dapat ditulis sebagai berikut :

$$d_i^T d_j = 0 \quad (1)$$

Sebelum masuk ke algoritma CG harus ditentukan dahulu fungsi objektif yang akan dioptimalisasi. Apabila algoritma CG digunakan sebagai algoritma pelatihan jaringan syaraf tiruan, maka tujuannya adalah meminimasi *error* yang bergantung pada bobot-bobot yang menghubungkan antar neuron. Oleh karena itu, fungsi objektif adalah fungsi *error* yaitu :

$$f(w) = \frac{1}{2N} \sum_n \sum_j (t_{nj} - y_{nj}(w))^2 \quad (2)$$

dimana : N = jumlah pola pada data training  
w = matriks bobot  
 $t_{nj}$  dan  $y_{nj}(w)$  = data target dan output neuron untuk n pola.

*Conjugate gradient* merupakan metode untuk meminimasi fungsi turunan dengan menghitung pendekatan  $w_{k+1}$  secara iterative berdasarkan :

$$w_{k+1} = w_k + \alpha_k d_k \quad (3)$$

$$d_{k+1} = -g_{k+1} + \beta_k d_k \quad (4)$$

dimana :  $\alpha$  dan  $\beta$  adalah parameter momentum untuk menghindari konvergensi local.

Algoritma *conjugate gradient* sebagai pelatihan jaringan syaraf tiruan adalah sebagai berikut :

1. Inisialisasi semua bobot dengan bilangan acak kecil
2. Jika kondisi penghentian belum terpenuhi, lakukan langkah 2 - 13

### • Propagasi maju

3. Setiap unit masukan menerima sinyal dan meneruskannya ke unit tersembunyi di atasnya.
4. Hitung semua keluaran di unit tersembunyi  $z_j$  ( $j = 1, 2, \dots, p$ ).

$$z_{net_j} = v_{j0} + \sum_{i=1}^n x_i v_{ji} \quad (5)$$

$$z_j = f(z_{net_j}) = \frac{1}{1 + e^{-z_{net_j}}} \quad (6)$$

5. Hitung semua keluaran di unit output  $y_k$  ( $k = 1, 2, \dots, m$ ).

$$y_{net_k} = w_{k0} + \sum_{j=1}^p z_j w_{kj} \quad (7)$$

$$y_k = f(y_{net_k}) = \frac{1}{1 + e^{-y_{net_k}}} \quad (8)$$

### • Propagasi mundur

6. Hitung factor kesalahan di unit output berdasarkan perbedaan (error) nilai actual dan nilai prediksi (output dari unit output).

$$\delta_k = (t_k - y_k) f'(y_{net_k}) \quad (9a)$$

$$\delta_k = (t_k - y_k) y_k (1 - y_k) \quad (9b)$$

7. Hitung factor kesalahan di unit tersembunyi berdasarkan factor kesalahan pada unit di atasnya.

$$\delta_{net_j} = \sum_{k=1}^m \delta_k w_{kj} \quad (10)$$

$$\delta_j = \delta_{net_j} f'(z_{net_j}) \quad (11a)$$

$$\delta_j = \delta_{net_j} z_j (1 - z_j) \quad (11b)$$

8. Hitung gradient di unit output dari fungsi objektif yang sudah ditentukan.

$$g_{k+1} = \frac{1}{N} \sum_{n=1}^p \delta_{nk} y_{nk} \quad (12)$$

9. Hitung gradient di unit tersembunyi.

$$g_{j+1} = \frac{1}{N} \sum_{n=1}^p \delta_{nj} z_{nj} \quad (13)$$

10. Hitung parameter  $\beta$  untuk semua neuron di unit tersembunyi dan unit output. Untuk *conjugate gradient polak ribiere* persamaannya sebagai berikut.

$$\beta_{t+1} = \frac{g_{t+1}^T (g_{t+1} - g_t)}{g_t^T g_t} \quad (14)$$

11. Hitung direction untuk semua neuron di unit tersembunyi dan unit output.

$$d_{t+1} = -g_{t+1} + \beta_t d_t \quad (15)$$

Untuk direction awal :

$$d_1 = -g_1 \quad (16)$$

12. Hitung parameter  $\alpha$  untuk semua neuron di unit tersembunyi dan unit output yang merupakan seberapa besar langkah yang diambil untuk setiap direction. Parameter ini dapat dicari dengan teknik line search.

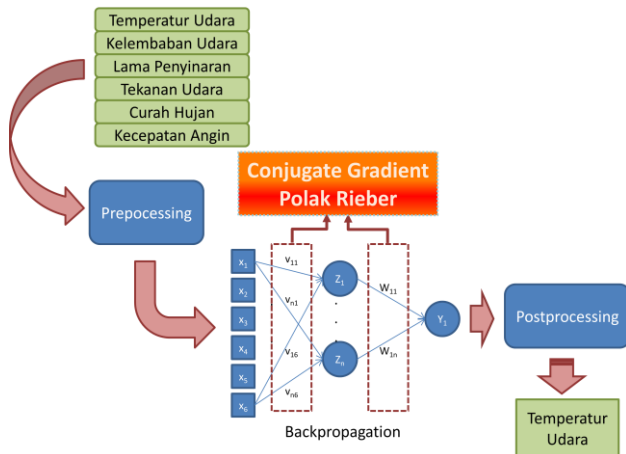
### • Perubahan bobot

13. Update bobot dilakukan dengan cara sebagai berikut :

$$w_{t+1} = w_t + \alpha_{t+1} d_{t+1} \quad (17)$$

### 3. Implementasi Algoritma dan Skenario Pengujian

Dalam penelitian ini dibangun suatu sistem untuk memprediksi nilai temperatur udara pada keesokan harinya (H+1). Sistem ini dibangun dengan menggunakan jaringan syaraf tiruan yang dilatih dengan metode conjugate gradient polak ribiere. Berikut ini diagram sistem yang dibangun secara umum.



Gambar 1 : Diagram sistem prediksi temperatur udara.

#### 3.1 Implementasi Algoritma untuk Sistem Prediksi Temperatur Udara

Secara umum proses peramalan temperatur udara dibagi dalam beberapa tahap sebagai berikut :

- Input data**  
Data yang digunakan dalam sistem peramalan ini merupakan data klimatologi yang diperoleh dari BMG (Badan Meteorologi dan Geofisika). Data ini meliputi temperature udara ( $^{\circ}\text{C}$ ), kelembaban (%), lama penyinaran (lpm), tekanan udara (mili bar), curah hujan (mili meter), dan kecepatan angin (knot).
- Preprocessing**  
Data yang digunakan dalam sistem peramalan mempunyai satuan dan range yang berbeda, oleh karena itu perlu dinormalisasi ke dalam range [0,1]. Setelah seluruh data tersebut dinormalisasi, kemudian data tersebut dibagi ke dalam tiga bagian, yaitu training set, validation set, dan testing set.
- Pelatihan Jaringan Syaraf Tiruan dengan Algoritma Conjugate Gradient Polak Ribiere**  
Dalam pelatihan JST ini menggunakan algoritma conjugate gradient polak ribiere dan juga validation set untuk mengetahui performansi JST dalam proses pelatihan. Proses pelatihan ini bertujuan untuk memperoleh arsitektur JST yang optimal beserta bobot-bobotnya.
- Pengujian Jaringan Syaraf Tiruan**

Tujuan dari tahap ini yaitu untuk menguji berapa performansi dari JST yang telah diperoleh dari proses pelatihan.

- Peramalan Temperatur Udara**  
Setelah melalui tahap pelatihan dan pengujian didapat sistem yang optimal. Sistem tersebut dapat digunakan untuk memprediksi temperatur udara pada H+1.
- Postprocessing**  
Output dari prediksi dengan jaringan syaraf tiruan masih dalam range [0,1]. Oleh karena itu perlu dilakukan postprocessing untuk mendapatkan nilai temperatur pada range yang sebenarnya.

Ukuran akurasi yang digunakan untuk performansi sistem yaitu *Mean Absolute Percentage Error* (MAPE) yang dinyatakan dalam persen.

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right| \quad (18)$$

dimana :  $A_t$  = data aktual  
 $F_t$  = data prediksi  
 $n$  = jumlah data

Apabila semua daya yang diprediksi sama dengan data aktual, maka MAPE bernilai nol.

#### 3.2 Skenario Pengujian

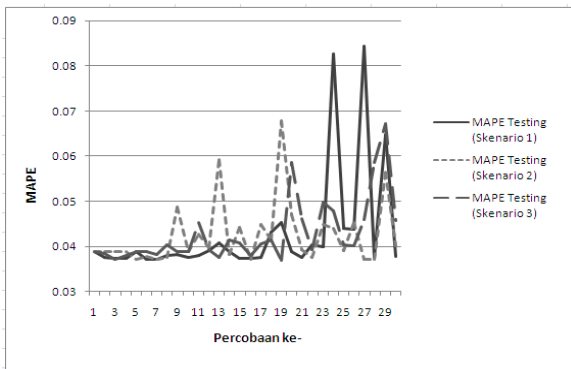
Data yang digunakan untuk pengujian adalah data klimatologi yang sudah dipreprocessing. Dalam proses pelatihan juga dilakukan proses validasi untuk meningkatkan kemampuan generalisasi JST. Dalam skenario pengujian ini data validasi dan data testing menggunakan data dan komposisi yang sama untuk scenario 1, scenario 2, dan scenario 3. Berikut ini adalah beberapa skenario yang akan diuji :

- Skenario pengujian 1**  
Komposisi data training – validasi – testing : 70% - 15% - 15%  
Kombinasi jumlah neuron hidden dan nilai teta : Jumlah neuron hidden : 5, 10, 15, 20, 25, dan 30  
Nilai teta : 0.01, 0.1, 0.2, 0.3, dan 0.4
- Skenario pengujian 2**  
Komposisi data training – validasi – testing : 50% - 15% - 15%  
Kombinasi jumlah neuron hidden dan nilai teta : Jumlah neuron hidden : 5, 10, 15, 20, 25, dan 30  
Nilai teta : 0.01, 0.1, 0.2, 0.3, dan 0.4
- Skenario pengujian 3**  
Komposisi data training – validasi – testing : 35% - 15% - 15%  
Kombinasi jumlah neuron hidden dan nilai teta : Jumlah neuron hidden : 5, 10, 15, 20, 25, dan 30  
Nilai teta : 0.01, 0.1, 0.2, 0.3, dan 0.4
- Skenario pengujian 4**  
Konfigurasi terbaik yang di dapat dari scenario 1, scenario 2, dan scenario 3 digunakan untuk proses peramalan terhadap data klimatologi Bandung tahun 2000.

#### 4. Analisis Hasil Pengujian

Berikut ini adalah MAPE hasil pengujian terbaik dari ketiga skenario 1, 2, dan 3, dimana ukuran performansi yang digunakan adalah MAPE.

- Skenario 1 : 0.037032 (dengan jumlah neuron hidden = 10 dan nilai teta = 0.01, jumlah epoch = 2)
- Skenario 1 : 0.037026 (dengan jumlah neuron hidden = 30 dan nilai teta = 0.1, jumlah epoch = 36)
- Skenario 1 : 0.036955 (dengan jumlah neuron hidden = 20 dan nilai teta = 0.3, jumlah epoch = 18)



Gambar 1 : Akurasi Backpropagation Termodifikasi

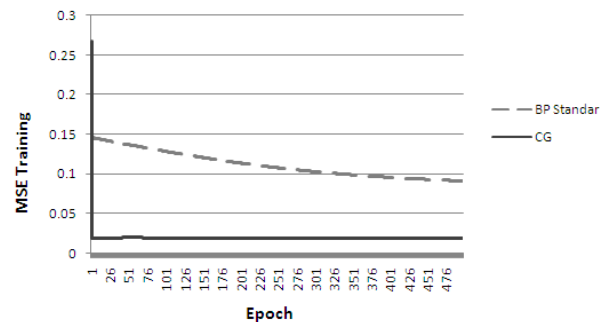
Gambar 1 merupakan pengujian skenario 1, skenario 2, dan skenario 3 dengan ukuran performansi berupa MAPE. Dari grafik di atas dapat terlihat bahwa hasil pengujian sangat dipengaruhi oleh jumlah neuron hidden. Pada jumlah neuron hidden yang kecil (antara 5 – 20 neuron hidden) MAPE hasil pengujian relatif stabil. Sedangkan hasil pengujian dengan jumlah neuron hidden 20 sampai 30 neuron mempunyai perbedaan MAPE yang relative besar antar hasil pengujian. Sedangkan nilai teta memberikan pengaruh kecil terhadap hasil pengujian (meskipun ada perbedaan MAPE, tetapi perbedaannya sangat kecil). Kemudian penggunaan data training dengan komposisi yang kecil atau besar, JST masih mampu melakukan peramalan dengan hasil yang bagus (dengan akurasi diatas 90%). Tetapi dengan menggunakan komposisi data training yang lebih besar (70% data training pada skenario 1) dapat melakukan prediksi dengan hasil yang lebih stabil pada jumlah neuron hidden 5 – 20 neuron daripada skenario lainnya.

Dengan pelatihan Backpropagation termodifikasi menggunakan algoritma Conjugate Gradient Polak Ribiere, jumlah epoch yang diperlukan hanya sedikit (rata-rata 46 epoch). Hal ini dikarenakan update bobot pada algoritma CG bersifat orthogonal terhadap bobot sebelumnya. Kemudian dengan adanya factor pencarian direction, parameter  $\alpha$  dan  $\beta$  membuat proses pelatihan cepat konvergen (update bobot tidak hanya berdasarkan factor kesalahan dan learning rate).

Table 1 : Performansi terbaik pengujian setiap skenario

Skenario	MAPE
1	0.038365
2	0.041410
3	0.036975

Dari table 1 dapat terlihat bahwa konfigurasi JST yang didapat dari proses pelatihan dengan algoritma Conjugate Gradient Polak Ribiere mampu memprediksi data klimatologi pada tahun yang berbeda, yaitu tahun 2000. Dan juga hanya dengan data training yang lebih sedikit (pada skenario 3) dapat memberikan akurasi yang lebih baik.



Gambar 2 : Proses pelatihan dengan Backpropagation standar dan Backpropagation termodifikasi.

Table 2 : Perbandingan Pengujian Backpropagation Standar dan Backpropagation termodifikasi.

	Epoch	MSE Training	MAPE Testing
Backpro Standar	500	0.090251	0.086769
Algoritma CG	18	0.020504	0.036955

Berdasarkan Gambar 2 dan Table 2 dapat dilihat bahwa pelatihan dengan algoritma CG memberikan hasil yang lebih baik daripada *backpropagation* standar. Hal ini dapat terjadi karena pada *backpropagation* standar update bobot hanya dipengaruhi oleh factor kesalahan dan nilai learning rate yang nilainya selalu tetap selama proses pelatihan. Sehingga pada saat jumlah epoch sudah mencapai 500 proses pelatihan masih belum menemukan bobot-bobot yang sesuai (belum konvergen). Sedangkan dengan algoritma CG, update bobot dipengaruhi oleh parameter  $\alpha$ ,  $\beta$ , dan direction yang akan mencari kemana arah pencarian dan seberapa besar lebar langkah yang diambil dalam mengupdate bobot. Sehingga dengan algoritma CG tersebut hanya diperlukan jumlah epoch yang kecil untuk mendapatkan nilai bobot-bobot yang sesuai.

Pada Tabel 4-2 dapat dilihat bahwa dengan algoritma CG didapat MAPE sebesar 0.036955 (atau 3.6955%). Dengan MAPE tersebut dapat dikatakan bahwa apabila data temperatur terdapat pada range 20 sampai 30 °C, maka *error* data prediksi berkisar antara 0.7 °C sampai 1.1 °C. Begitu pula dengan *Backpropagation* standar yang mempunyai MAPE

sebesar 0.086769 (8.6769%), maka error data prediksi sebesar 1.7 °C sampai 2.6 °C.

## 5. Kesimpulan dan Saran

Berdasarkan implementasi dan analisis sistem yang telah dibangun dan hasil pengujian, maka dapat ditarik beberapa kesimpulan sebagai berikut :

- a. Parameter-parameter JST dan algoritma conjugate gradient polak ribiere, yaitu jumlah neuron hidden dan nilai teta dapat mempengaruhi performansi system. Pada kasus yang berbeda, semakin besar nilai dari parameter tersebut belum tentu memberikan performansi yang lebih baik. Pada tugas akhir ini didapat bahwa jumlah neuron hidden yang memberikan performansi terbaik terletak pada range 5 sampai 20 neuron, sedangkan nilai teta pengaruhnya sangat kecil pada hasil pengujian.
- b. Komposisi data training yang digunakan pada proses pelatihan juga mempengaruhi kemampuan system dalam proses peramalan. Komposisi data training sebesar 70% memberikan hasil yang lebih stabil pada jumlah neuron 5 sampai 20 neuron. Sedangkan untuk data klimatologi pada tahun yang berbeda hanya dibutuhkan komposisi data training untuk memberikan performansi yang baik dengan akurasi sebesar 96,3% (MAPE sebesar 0.036975).
- c. Untuk mendapatkan performansi system yang baik hanya dibutuhkan sedikit epoch (rata-rata 46 epoch) dalam proses training. Dengan kata lain, dengan pelatihan Backpropagation termodifikasi menggunakan Conjugate Gradient Polak Ribiere lebih cepat konvergen dalam mendapatkan solusi dibandingkan dengan Backpropagation standar yang harus mencapai minimal 500 epoch.

## Daftar Pustaka

- [1] Battiti, R., 1989, *Accelerated Backpropagation Learning : Two Optimization Methods*, Complex Syatem, 3, 331-342.
- [2] Cresswell, M.P., Thomson, M.C., and Connor, S.J, 1999, *Estimating surface air temperatures, from meteosal land surface temperature using an empirical solar zenith angle model*. International Journal of Remote Sensing, Vol. 20,No.6, pp. 1125-1132.
- [3] Enrique Castillo, B. G. Berdinas, O. F. Romero, and A. A. Betanzos, 2006, *A Very Fast Learning Method for Neural Networks Based on Sensitivity Analysis*, Spain, University of Cantabria.  
[jmlr.csail.mit.edu/papers/volume7/castillo06a/castillo06a.pdf](http://jmlr.csail.mit.edu/papers/volume7/castillo06a/castillo06a.pdf)
- [4] Jae-Dong Jang Dr. Alain A. Viau and Dr. Francois Anctil, 2003, *Estimation of Air Temperatures*, Kanada, Canadian Journal of Forest Research.  
<http://www.theses.ulaval.ca/>
- [5] Jian Wang, Xuebin Chi, TongXiang Gu, *Nonlinier Conjugate Gradient Methods and Their Implementations by TAO on Dawning 2000-II<sup>+</sup>*.  
[www.sccas.cn/ICPACE/papers/Jian%20Wan%20g.pdf](http://www.sccas.cn/ICPACE/papers/Jian%20Wan%20g.pdf)
- [6] Magy. M. Kandil, Fayza. A. Mohamed, Fathy Saleh, Magda Fayek, 2005, *A New Approach For Optimizing Backpropagation Training with Variable Gain Using PSO*, Cairo University.  
[www.icgst.com](http://www.icgst.com)
- [7] Salakhutdinov, Ruslan, 2003, *Efficient Optimization Algorithms for Learning*, University of Toronto.  
[www.cs.toronto.edu/~roweis/papers/Ruslan\\_msc\\_thesis.pdf](http://www.cs.toronto.edu/~roweis/papers/Ruslan_msc_thesis.pdf)
- [8] Shewchuk, Jonathan Richard, 1994, *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*, Pittsburgh, School of Computer Science Carnegie Mellon University.
- [9] Wong Chi-Cheong, Chan Man-Chung, Lam Chi-Chung, 1998, *Financial Time Series Forecasting by Neural Network Using Conjugate Gradient Learning Algorithm and Multiple Linier Regression Weight Initialization*, Journal of Artificial Intelligence.
- [10] Y.H. Dai, Y. Yuan, 2002, *A Three-Parameter Family of Nonlinier Conjugate Gradient Methods*, Beijing, Chinese Academy of Sciences.  
[www.ams.org/mcom/2001-70-235/S0025-5718-00-01253-9/S0025-5718-00-01253-9.pdf](http://www.ams.org/mcom/2001-70-235/S0025-5718-00-01253-9/S0025-5718-00-01253-9.pdf)