Volume 3, Nomor 2, Desember 2019
Numerical: Jurnal Matematika dan Pendidikan Matematika
http://journal.iaimnumetrolampung.ac.id/index.php/numerical
DOI:https://doi.org/10.25217/numerical.v3i2.603

Optimasi Learning Rate Neural Network Backpropagation dengan Seacrh Direction Conjugate Gradient Pada Electrocardiogram

Azwar Riza Habibi, Vivi Aida Fitria, Lukman Hakim

Institut Teknologi dan Bisnis Asia Malang, Indonesia
Correspondence: ✉riza.bj@gmail.com

Article Info

Article History
Received :12-11-2019
Revised : 16-12-2019
Accepted : 26-12-2019
Keywords:

Neural Network;
Conjugate Gradient;
Pembobotan;
Arah Pencarian

Abstract

This paper develops a Neural network (NN) using conjugate gradient (CG). The modification of this method is in defining the direction of linear search. The conjugate gradient method has several methods to determine the steep size such as the Fletcher-Reeves, Dixon, Polak-Ribere, Hestene Steifel, and Dai-Yuan methods by using discrete electrocardiogram data. Conjugate gradients are used to update learning rates on neural networks by using different steep sizes. While the gradient search direction is used to update the weight on the NN. The results show that using Polak-Ribere get an optimal error, but the direction of the weighting search on NN widens and causes epoch on NN training is getting longer. But Hestene Steifel, and Dai-Yua could not find the gradient search direction so they could not update the weights and cause errors and epochs to infinity.

PENDAHULUAN

Neural network meruapakan sistem komputasi dimana arsitektur dan operasi dari sel saraf biologis otak manusia, yang merupakan salah satu model matemaris yang merepresentasikan dari otak manusia yang selalu mencoba menstimulasi proses pembelajaran pada otak manusia tersebut. Secara prinsip *NN* ini dibangkitkan oleh bobot dengan serangkaian data yang masing-masing menggambarkan keluaran neuron yang lain [1].

Neural network merupakan lapisan jamak dengan dua atau lebih lapisan, meskipun sebagian besar jaringan terdiri dari tiga lapisan: lapisan input, lapisan tersembunyi, dan lapisan output [2]. Neural network memiliki beberapa properti yang membuat mereka populer untuk clustering. Pertama, arsitektur neural network merupakan pengolahan *inheren parallel* dan terdistribusi. Kedua, jaringan ini belajar dengan menyesuaikan bobot interkoneksi dengan data, Hal ini memungkinkan neural network untuk "menormalkan" pola dan bertindak sebagai fitur (atribut) *extractors* untuk kelompok yang berbeda. Ketiga, neural network memproses vektor numerik dan membutuhkan pola objek untuk diwakili oleh fitur kuantitatif saja [3].

Setiap masukan dikalikan dengan suatu faktor pembobot tertentu dan kemudian semua masukan terbobot itu dijumlahkan untuk menentukan tingkat aktivitas suatu neuron [4]. Pembobotan dinyatakan berupa elemen-elemen matriks bobot(w) dengan dimensi matriks adalah m baris dan n kolom, di mana m merupakan jumlah masukan dan n menunjukkan jumlah neuron. Misalnya, pembobot dengan yang menghubungkan masukan keempat ke neuron ketiga dinyatakan dengan w [5].

Mengembangkan metode NN dengan menggunakan Conjugate gradient, dengan menggunakan *gradient* pada data [6]. Dari hasil penulis mengembangkan *conjugate gradient - neural network* (CG-NN) dengan merubah fungsi pada *conjugate* untuk mendapatkan *alpha* yang optimal. Penelitian ini bertujuan untuk memperbaiki kerja kecepatan konvergen pada algoritma perambatan balik standart. Algoritma *conjugate gradient* merupakan algoritma iterative yang handal untuk menyeleseikan persamaan linier simultan dalam sekala besar yang dapat digunakan untuk mengoptimalkan laju perambatan pada NN [7].

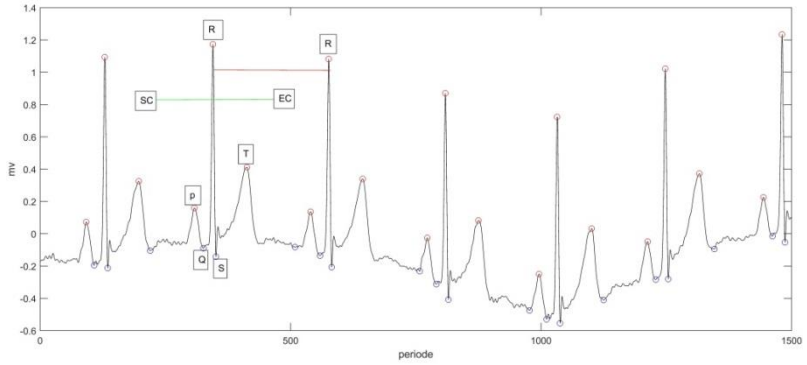
Print ISSN: 2580-3573, Online ISSN: 2580-2437

Metode CG-NN-PSOGA Mengembangkan NN dengan dengan menggunakan *conjugate gradient* untuk meningkatkan bobot yang dipakai untuk optimasi pada NN dan Laju percepatan yang telah di*update* dapat meningkatkan efisiensi dalam waktu komputasi. PSO dan GA digunakan untuk *update* bobot dari factor δ berdasarkan kesalahan di setiap unit keluaran (y_k), sehingga selisih *error* pada setiap unit keluaran (y_k) dapat meningkatkan efisiensi dalam meminimumkan *error*. Tetapi pada modifikasi CG-NN hanya digunakan untuk optimasi error [8].

Peper ini mengembangkan CG-NN (*conjugate gradient-neural network*)dengan menggunakan modifikasi dari arah pencarian yang berbeda dengan metode *Fletcher-Reeves, Dixon, Polak-Ribere, Hestene Steifel,* dan *Dai-Yuan*[9]. Modifikasi arah pencarian ini digunakan untuk update alpha pada NN agar mendapatkan laju percepatan yang optimal, data yang digunakan adalah data *electrocardiogram* diskrit. Dari beberapa modifikasi dari conjugate gradient bertujuan untuk mendapatkan nilai alpha yang optimum.

METODE PENELITIAN

Data yang dipakai adalah data *electrocardiogram* diskrit yang diambil dari http://www.physionet.org, dan hanya membahas dari jaringan AVL. Data *electrocardiogram* ini bertujuan untuk melihat laju pembelajaran dari modifikasi NN tersebut untuk mendeteksi pola pada data *electrocardiogram,* dengan menggunakan lead II pada AVL[10].

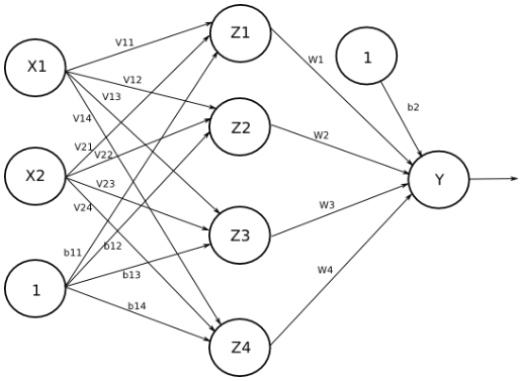


Gambar (1) Variabel PQRST*electrocardiogram*

Pada paper ini digunakan data AVL dengan 2 titik puncak pada data lead II, Metode yang digunakan dalam update *learning rate* pada Neural Network dengan merubah arah pencarian pada *conjugate gradient*. Langkah-langkah update learning rate dan bobot pada neural network sebagai berikut:

1) Penentuan Arsitektur jaringan

Jaringan backpropagation terdiri dari 3 tahap, 1) Perhitungan langkah maju (*feedforwad*) untuk input pola pelatihan, 2) Perhitungan dan perambatan balik (*backpropagation*) dari galat yang bersangkutan dan 3) penyesuaian bobot. Model jaringan *backpropagation* sebagai berikut[11]:



Gambar (2) Arsitektur Jaringan

Print ISSN: 2580-3573, Online ISSN: 2580-2437

2) Penentuan Seacrh Direction Conjugate Gradient

Merode conjugate gradient digunakan untuk menyeleseikan persamaan linier, metode ini efektif untuk sistem persamaan linier dengan ukuran besar, yaitu:

$$Ax = b \tag{1}$$

Dengan x adalah vektor yang tidak diketahui, b adalah vektor yang diketahui dan A adalah matriks simetris. Metode conjugate gradien ini digunakan untuk mecari arah pencarian arah dengan metode penurunan

$$p_0 = -g_0 \tag{2}$$

Pencarian arah tersebut maka perubahan gradient untuk $k+1$, yaitu:

$$\nabla x_k = (x_{k+1} - x_k) = a_k p_k \tag{3}$$

$$a_k = -\frac{\nabla F(x)^T\big|_{x=x} p_k}{p_k^t \nabla F(x)\big|_{x=x} p_k} = -\frac{g_k^T p_k}{p_k^T A_k p_k} \tag{4}$$

Selanjutnya pilih arah pencarian dengan menggunakan

$$p_k = -g_k + \beta_k p_{k-1} \tag{5}$$

Dengan β_k dapat dihitung dengan beberapa metode yang memberikan hasil yang sama untuk fungsi kuadratis, fungsi β_k pada *conjugate gradient* sebagai berikut[12]:

1. *Fletcher-Reeves* = $\beta_{k+1} = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}$
2. *Dixon* = $\beta_{k+1} = \frac{g_{k+1}^T g_{k+1}}{d_k^T g_k}$
3. *Polak-Ribere* = $\beta_{k+1} = \frac{g_{k+1}^T (g_{k+1} - g_k)}{g_k^T g_k}$
4. *Hestene Steifel* = $\beta_{k+1} = \frac{g_{k+1}^T (g_{k+1} - g_k)}{d_k^T (g_{k+1} - g_k)}$
5. *Dai-Yuan* = $\beta_{k+1} = \frac{g_{k+1}^T g_{k+1}}{d_k^T (g_{k+1} - g_k)}$

3) Update Bobot dan *Learning Rate*

Penerapan model *conjugate gradient* pada pembelajaran *neural network* bertujuan untuk meminimumkan *error* yang terjadi dari selisih antara *output* dan target yang diberikan. Pada jaringan *neural network* setiap unit-unit pada *input layer* x_i dan *output layer* y_k, perubahan bobot dengan menurunkan metode *conjugate gradient* pada perubahan bobot w_{jk} dan *learning rate*. Perubahan *learning rate* akan berubah dan akan menyesuaikan dengan perubahan data yang digunakan, akan tetapi *learning rate* akan mendapatkan nilai yang optimal apabila arah gradient menemukan arah pencarianya.

4) Normalisasi Sigmoid

Pelatihan jaringan saraf agar dapat dibuat lebih efisien dengan melakukan preprocessing tertentu pada input dan target jaringan. Fungsi pemrosesan input jaringan mengubah input menjadi bentuk yang lebih baik untuk penggunaan jaringan. Proses normalisasi untuk input mentah memiliki efek besar pada persiapan data

Print ISSN: 2580-3573, Online ISSN: 2580-2437

agar sesuai untuk pelatihan serta bobot yang didapatkan optimal. Pada penelitian ini menggunakan normalisasi Sigmoid untuk pengecekan setiap bobot yang dihasilkan dari metode *conjugate* [13].

$$x' = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}},\ y' = (x_i - x_{i+1})\frac{y}{x} + y \tag{6}$$

HASIL DAN PEMBAHASAN

Pada pembahasan ini penyelesaian permasalah di atas diselesaikan menggunakan metode *conjugate gradient* dengan turunan untuk *update* perubahan bobot w_{jk}, $j = 1,2...,n$, $k = 1,2...,m$

$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial y_k}\frac{\partial y\_in_k}{\partial w_{jk}}\frac{\partial y_k}{\partial y\_in_k} \tag{7}$$

$$= \frac{\partial(\frac{1}{2}\sum_{k=1}^{m}(t_k - y_k)^2)}{\partial y_k}\frac{\partial(w_{ok} + \sum_{j=1}^{p} z_j w_{jk})}{\partial w_{jk}}\frac{\partial(f(y\_in_k))}{\partial y\_in_k} \tag{8}$$

$$\frac{\partial E}{\partial w_{jk}} = -\sum_{k=1}^{m}(t_k - y_k)\frac{\partial f}{\partial x}(y\_in_k)z_j,\ j \neq 0, \tag{9}$$

dengan memisalkan

$$\delta = -\sum_{k=1}^{m}(t_k - y_k)\frac{\partial f}{\partial x}(y\_in_k), \text{ maka} \tag{10}$$

$$\frac{\partial E}{\partial w_{jk}} = -\delta z_j. \tag{11}$$

Untuk memperbarui *gradient* pada (t) berdasarkan persamaan (7) maka

$$g_{w_{jk}}(t) = \frac{\partial E}{\partial w_{jk}} \tag{12}$$

Berdasarkan metode CGFR pada persamaan (6) dan (12) maka

$$d_{w_{jk}}(t) = -g_{w_{jk}}(t) \text{ untuk } t = 0 \tag{13}$$

$$d_{w_{jk}}(t) = -g_{w_{jk}}(t) + \beta_{w_{jk}}(t)d_{w_{jk}}(t-1) \text{ untuk } t = 1,2...,n \tag{14}$$

$$\text{Dengan } \beta_{w_{jk}}(t) = \frac{g^{T}_{w_{jk}} g_{w_{jk}}(t)}{g^{T}_{w_{jk}}(t-1) g_{w_{jk}}(t-1)},\ \alpha = \beta_{w_{jk}} \tag{15}$$

dengan perubahan bobot w_{jk} pada iterasi ke $(t+1)$ adalah sebagai berikut

$$w_{jk}(t+1) = w_{jk}(t) + \eta dw_{jk}(t) \tag{16}$$

Berlaku juga untuk bias w_{0k}, bobot v_{ij}, bobot bias v_{0j} Sehingga didapatkan 4 persamaan *update* untuk bobot w_{jk}, w_{0k}, v_{ij} dan v_{0j} sebagai berikut.

1. Untuk *update* bobot dan bias pada *input layer*ke *hidden layer*
   $v_{ij}(t+1) = v_{ij}(t) + \eta dv_{ij}(t)$
   $v_{0j}(t+1) = v_{0j}(t) + \eta dv_{0j}(t)$
2. Untuk *update* bobot dan bias pada *hidden layer* ke *output layer*
   $w_{jk}(t+1) = w_{jk}(t) + \eta dw_{jk}(t)$
   $w_{0k}(t+1) = w_{0k}(t) + \eta dw_{0k}(t)$

Dengan η adalah laju pembelajaran *update* (*learning rate update*)

Dari perubahan yang dilakukan oleh bobot dan update leraning rate maka didapatkan algoritma dibawah ini:

Tabel (1) Algoritma CG-NN update learning rate

Print ISSN: 2580-3573, Online ISSN: 2580-2437

Algoritma

1 $\alpha = 0, w_{jk} = 0, v_{ij} = 0, epsilon = 0, hiden = 12$

2 **While** (MSE_NNCG>*Epsilon* **do**

3 **For** j=1:Hiden; **For** I i=1:bnykdata ; $z\_in_j = v_{0j} + \sum_{i=1}^{n} x_i v_{ij}$; **end**

4 $z_j = f(z\_in_j)$;**end**

5 **For** j=1: Hiden; $xCG_j = f(data_j) + d$; **end**

6 **For** k = 1 : bnykdata; $delta_k = a_k p_k$, $\eta_k = \frac{g_k P'_k}{p_k delta_k P'_k}$; **end**

7 **For** k=1: bnykdata; **For** j =1: Hiden; $y\_in_k = w_{0k} + \sum_{j=1}^{p} z_j w_{jk}$; **end**

8 $y_k = f(y\_in_k)$; **end**

9 **For** k = 1 : bnykdata; $\delta_k = (t_k - y_k)\frac{\partial f}{dx}(y\_in_k)$; **end**

10 **For** k=1: bnykdata; **For** k=1: bnykdata; $\Delta w_{jk} = \eta \delta_k z_j$; **end**

11 $\Delta w_{0k} = \eta \delta_k$; **end**

12 **For** j=1:Hiden; **For** k=1: bnykdata; $\delta\_in_j = \sum_{k=1}^{m} \delta_k w_{jk}$; **end**

13 $\delta_j = \delta\_in_j \frac{\partial f}{dx}(z\_in_j)$, **end**

14 **For** j=1:Hiden; **for** I i=1:bnykdata; $\Delta v_{ij} = \alpha \delta_k x_i + \eta dv_{0j}$; **end**

15 $\Delta v_{0j} = \eta dv_{0j}$; **end**

16 **For** k=1: bnykdata; **For** j = Hiden; $w_{jk}(baru) = w_{jk}(lama) + \eta dw_{jk}$; **end; end**

17 **For** j=1:Hiden; **for** I i=1:bnykdata; $v_{ij}(baru) = v_{ij}(lama) + \eta dv_{ij}$;**end; end**

18 For k=1: bnykdata ; **For** j = Hiden ; $E = -\sum_{k=1}^{m}(t_k - y_k)\frac{\partial f}{\partial x}(y\_in_k)z_j$; **end; end**

19 **End**

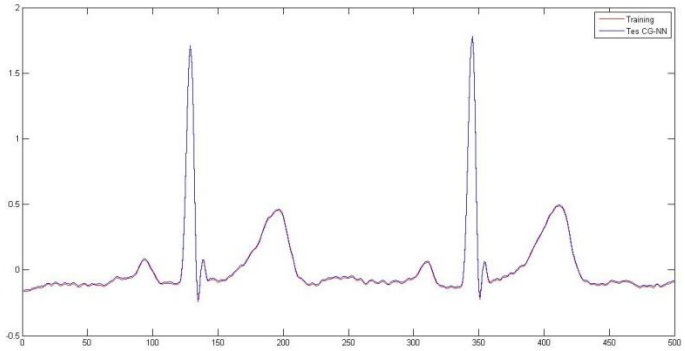
Algoritma diatas terdapat fungsi pada *linier sistem* dengan perhitungan sendiri sehingga algoritma di atas merupakan perunbahan pada NN dengan *conjugate gradient*. Hasil dari beberapa metode yang digunakan dalam *conjugate gradient* dalam pencarian arah gradient pada data *electrocardiogram* diskrit dengan menggunakan nilai awal *learning rate* ($\alpha = 0$), jumlah hidden layer =12, fungsi aktifasi menggunakan sigmoid binner, bobot awal w = 0.sehingga didapatkan hasil sebagai berikut:

Tabel(2) Hasil perhitungan CG-NN update *learning rate*

No	Metode	α (*Learning rate*)	*Epoh*	*Error*
1	*Fletcher-Reeves*	0.0419	7	0.0198
2	*Dixon*	0.0563	5	0.0003
3	*Polak-Ribere*	0.0736	11	0.0001
4	*Hestene Steifel*	0.2373	∞	∞
5	*Dai-Yuan*	0.1884	∞	∞

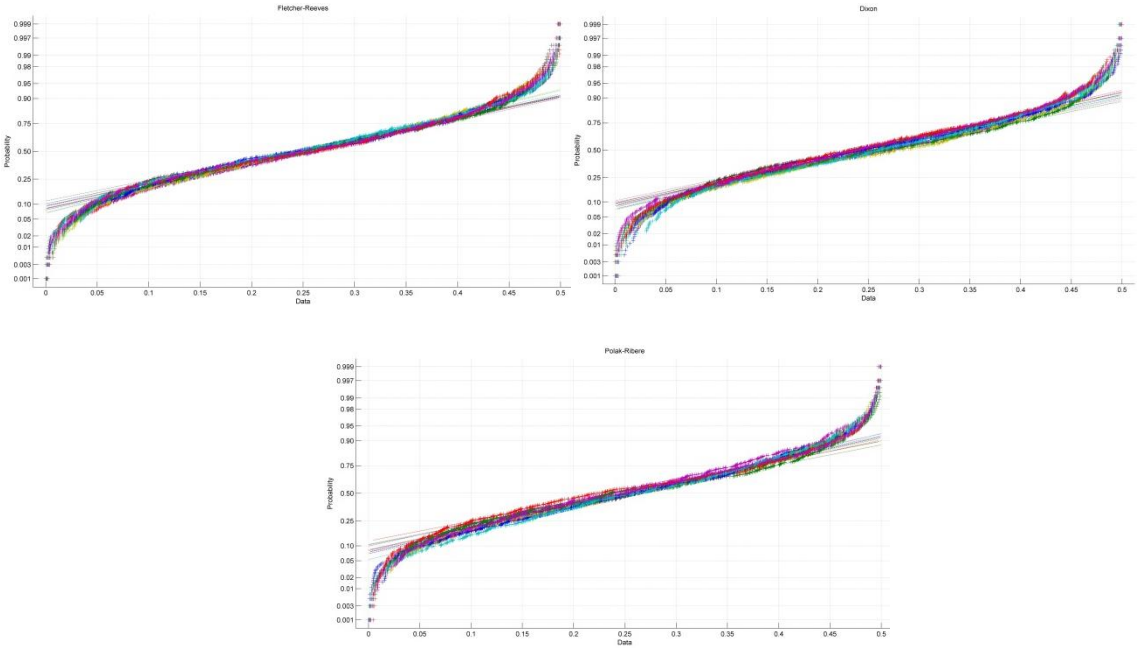
Print ISSN: 2580-3573, Online ISSN: 2580-2437

Tabel diatas didapatkan beberapa bahwa metode *Polak-Ribere* mendapatkan epoh lebih lama dan *error* lebih baik dari pada *error* dari metode *conjugate gradient* dengan . Hal ini menunjukan bahwa arah pencarian pada data *electrocardiogram* diskit bias terdeteksi pola dari data tersebut. Tetapi pada *Hestene Steifel* dan *Dai-Yuan* tidak dapat menemukan bobot yang optimal sehingga epoh yang didapatkan menuju tak terhingga dan nilai error tidak dapat ditentukan karena nilai bobot pada jaringan NN dengan menggunakan *Hestene Steifel* dan *Dai-Yuan* tidak dapat menemukan bobot yang optimal.



Gambar (3)Data tranning *electrocardiogram* dan Tes CG-NN

Dari data *electrocardiogram* diskrit pada arah *Polak-Riber* pada update *learning rate* dan bobot pada NN terlihat bahwa pengecekan hasil training mendapatkan bobot yang optimal dan minimum *error*.



Gambar (4) Persebaran Bobot yang ter-*update*

Persebaran data pada bobot dengan menggunakan *Polak-Riber* terlihat bahwa semakin lebar bobot yang terbentuk dari update bobot tersebut maka semakin banyak pola yang akan terbentuk. Hal ini mengakibatkan epoh yang terjadi semakin lama tetapi error yang diberikan semakin kecil. Tetapi pada *Hestene Steifel* dan *Dai-Yuan* tidak dapat menentukan bobot yang optimal sehingga persebaran pada bobot tersebut tidak terbentuk.

Print ISSN: 2580-3573, Online ISSN: 2580-2437

SIMPULAN DAN SARAN

Metode CG-NN dengan meng-update learning rate dan bobot yang terbentuk dengan menggunakan beberapa pengembangan *Conjugate Gradinet*, tidak semua metode pencarian arah menemukan bobot optimal dengan contoh data *electrocardiogram* pada AVL. Hal ini ditunjukan dengan perubahan bentuk dari conjugate gradient mengakibatkan perbuahan persebaran bobot tidak merata. Apabila arah pencarian gradient dengan data *electrocardiogram* tidak dapat ditemukan maka bobot tidak dapat terbentuk dan tidak dapat melanjutkan proses Trainning NN. Kelemahan tersebut bisa diatasi dengan memproses data terlebih agar mendapatkan data yang baik untuk dapat digunakan pada proses training pada CG-NN. Sehingga dapat dikembangkan dengan pemilihan data yang ternormalisasi dengan metode statistik.

DAFTAR PUSTAKA

[1] M. Badrul, N. Mandiri Jakarta, J. Damai No dan W. Jati Barat Jakarta Selatan, "Optimasi Neural Network Dengan Algoritma Genetika Untuk Prediksi Hasil Pemilukada," pp. 2355-3421.

[2] O. Soesanto, S. S. M. Si, A. E. Fahrudin, S. Si, M. Eng, D. Turianto dan S. Kom, "Optimasi Learning Radial Basis Function Neural Network dengan Extended Kalman Filter," *Kumpulan jurnaL Ilmu Komputer (KLIK),* vol. 03, no. 02, 2015.

[3] J. Bernal dan J. Torres-Jimenez, "SAGRAD: A program for neural network training with simulated annealing and the conjugate gradient method," *Journal of Research of the National Institute of Standards and Technology,* vol. 120, pp. 113-128, 2015.

[4] A. G. Karegowda, A. Manjunath dan M. Jayaram, "Application of Genetic Algorithm Optimized Neural Network Connection Weights for Medical Diagnosis of PIMA Indians Diabetes," *International Journal on Soft Computing,* vol. 2, no. 2, pp. 15-23, 31 5 2011.

[5] A. Y. Prathama, "Pendekatan Ann (Artificial Neural Network) Untuk Penentuan Prosentase Bobot Pekerjaan Dan Estimasi Nilai Pekerjaan Struktur Pada Rumah Sakit Pratama," *Jurnal Teknosains,* vol. 7, no. 1, p. 14, 13 7 2018.

[6] X.-B. Jin, X.-Y. Zhang, K. Huang dan G.-G. Geng, "Stochastic Conjugate Gradient Algorithm with Variance Reduction," 26 10 2017.

[7] F. Jing Wang, C. L. Philip Chen, "On the Conjugate Gradients (CG) Training Algorithm of Fuzzy Neural Networks (FNNs) via Its Equivalent Fully Connected Neural Networks (FFNNs)," *IEEE International Conference on Systems, Man, and Cybernetics,* vol. October 14, no. 17, pp. 2446-2451, 2012.

[8] A. Riza Habibi, R. Bagus Edy Wibowo dan P. Student, "Modification Of Neural Network Algorithm Using Conjugate Gradient With Addition Of Weight Initialization," *Journal of Theoretical and Applied Information Technology,* vol. 10, no. 1, 2015.

[9] A. Y. Al Bayati, N. A. Sulaiman dan G. W. Sadiq, "A Modified Conjugate Gradient Formula for Back Propagation Neural Network Algorithm," *Journal of Computer Science,* vol. 5, no. 11, pp. 849-856, 2009.

[10] A. R. Habibi, S. I. Putri dan L. Hakim, "Aplikasi Representasi Real Time Gelombang Electrocardiograph Diskrit," *Jurnal Ilmiah Teknologi Informasi Asia,* vol. 12, no. 01, 2018.

[11] G. Taylor, R. Burmeister, Z. Xu, B. Singh, A. Patel dan T. Goldstein, "Training Neural Networks Without Gradients: A Scalable ADMM Approach," 2016.

[12] A. Abashar, M. Mamat, A. Alhawarat, F. Susilawati, Z. Salleh dan Z. A. Zakaria, A Modified Fletcher-Reeves Conjugate Gradient Method For Unconstrained Optimization, 2016.

[13] A. T.Jayalakshmi, "Statistical Normalization and Back Propagation for Classification," *International Journal of Computer Theory and Engineering,* vol. 3, no. 1, pp. 89-93, 2011.

[14] A. S. Anagun dan T. Sarac, "Optimization of performance of genetic algorithm for 0-1 knapsack problems using taguchi method," dalam *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2006.

Print ISSN: 2580-3573, Online ISSN: 2580-2437