

Meningkatkan Kinerja Backpropagation Neural Network Menggunakan Algoritma Adaptif

Abdul Rahman Bohari

Fakultas Teknik dan Informatika
Universitas Bina Sarana Informatika
e-mail: abdul.lrb@bsi.ac.id

Abstrak - Aplikasi Jaringan Syaraf Tiruan di berbagai bidang kehidupan manusia semakin luas, khususnya di bidang industri. Salah satu struktur Jaringan Syaraf Tiruan yang cukup sering digunakan adalah *Feedforward Neural Network* dengan algoritma belajarnya yang cukup dikenal yaitu *Backpropagation*. Akan tetapi sebagaimana dilaporkan beberapa peneliti, *Backpropagation* memiliki beberapa kelemahan seperti membutuhkan waktu lama untuk konvergensi dalam proses training, cukup sensitif dengan kondisi awal weight dan relatif sering terjebak pada lokal minima yang bisa menggagalkan proses training. Pada penelitian ini, algoritma Adaptif diusulkan sebagai alternatif algoritma belajar *Backpropagation*. Algoritma yang diusulkan tersebut memberikan harapan dalam mengatasi kelemahan yang dihadapi oleh *Backpropagation*. Sebagaimana dilaporkan pada hasil pengujian, dibandingkan dengan *Backpropagation*, algoritma Adaptif jauh lebih kuat menghadapi variasi terhadap kondisi awal weight. Dari 100 kali pengujian pada penelitian ini untuk masing-masing *Backpropagation* dan algoritma Adaptif, dengan variasi random untuk nilai awal weight, tingkat keberhasilan proses training algoritma Adaptif mencapai 100% dibandingkan *Backpropagation* yang berada di level 77%. Dari sisi kecepatan, pada penelitian ini algoritma Adaptif berhasil melakukan proses training dengan jumlah rata-rata iterasi sebanyak 37 kali dibandingkan dengan *Backpropagation* yang membutuhkan rata-rata 162 kali iterasi.

Kata Kunci: Feedforward Neural Network, Backpropagation, Algoritma Adaptif

Abstract - The application of Artificial Neural Networks in various fields of human life is getting wider, especially in the industrial sector. One of the artificial neural network structures that are quite often used is the *Feedforward Neural Network* with its well-known learning algorithm, namely *Backpropagation*. However, as reported by several researchers, *Backpropagation* has several weaknesses such as it takes a long time to converge in the training process, it is quite sensitive to initial weight conditions and is relatively often trapped in a local minima which can thwart the training process. In this study, the Adaptive algorithm is proposed as an alternative to the *Backpropagation* learning algorithm. The proposed algorithm provides hope in overcoming the weaknesses faced by *Backpropagation*. As reported in the test results, compared to *Backpropagation*, the Adaptive algorithm is much stronger in dealing with variations in the initial weight conditions. From 100 tests in this study for each *Backpropagation* and Adaptive algorithm, with random variations for the initial weight value, the success rate of the Adaptive algorithm training process reaches 100% compared to *Backpropagation* which is at the level of 77%. In terms of speed, the Adaptive algorithm has successfully carried out the training process with an average number of iterations of 37 times compared to *Backpropagation* which requires an average of 162 iterations.

Keywords : Feedforward Neural Network, Backpropagation, Adaptive Algorithm

PENDAHULUAN

Pemakaian Jaringan Syaraf Tiruan (JST), khususnya *Feedforward Neural Network* (FFNN) dengan algoritma *Backpropagation* semakin mendapatkan tempat diberbagai industri. Hal ini bisa kita lihat misalnya pada aplikasi yang digunakan untuk memprediksi karkarakteristik material komposit nano (Ma, J., Zhu, S. G., Wu, C. X., & Zhang, M. L., 2009), aplikasi pada pengelasan mikro berbasis laser (Ismail, M. I. S., Okamoto, Y., & Okada, A., 2013), atau aplikasi pada estimasi biaya produk injeksi plastik (Wang, H. S., 2007). Tentu masih begitu banyak lagi aplikasi atau penggunaan

Backpropagation Neural Network (BPN) di dunia industri bahkan diberbagai bidang kehidupan, yang tidak cukup untuk dituliskan di ruang terbatas di sini.

Akan tetapi, sebagaimana dilaporkan oleh beberapa peneliti, *Backpropagation* sebagai algoritma training FFNN memiliki kelemahan. Cilimkovic, M. (2015) menyebutkan bahwa kelemahan terbesar pada *Backpropagation* adalah waktu untuk training yang dibutuhkan cukup besar. Ray, S. (2019) mengidentifikasi beberapa kelemahan dari *Backpropagation* yaitu memerlukan waktu yang cukup lama untuk konvergensi (*slow convergence*) dan sering terjebak pada lokal minima (local



minima) yang menyebabkan kegagalan proses training. Kelemahan lain dari *Backpropagation* adalah cukup sensitif terhadap kondisi awal dari weight (Abdul Hamid, N., et.al, 2012).

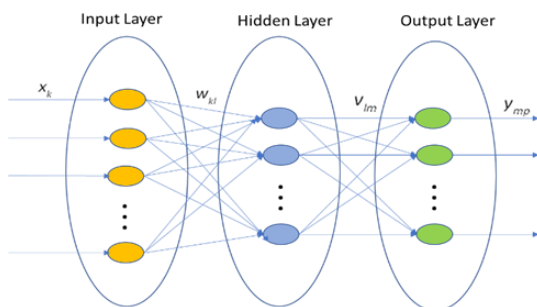
Untuk mengatasi beberapa kelemahan mendasar pada *Backpropagation*, diperlukan alternatif algoritma belajar (learning algorithm). Beberapa peneliti telah mencoba mengatasi permasalahan ini, diantaranya dengan menggunakan Algoritma Genetik pada tahapan pemilihan nilai awal wight untuk sehingga bisa meningkatkan kinerja *Backpropagation* (Yi, X., 2015; Dasuki, M., 2021). Pada penelitian kali ini, penulis mengusulkan algoritma Adaptif sebagai alternatif algoritma belajar *Backpropagation*. Bagian Jaringan Syaraf Tiruan, sesudah bagian Pendahuluan ini, menjelaskan lebih lanjut terkait dengan algoritma Adaptif.

JARINGAN SYARAF TIRUAN

JST adalah sistem yang dibuat berdasarkan prinsip kerja otak manusia, yang terkoneksi secara massif dan parallel serta memiliki kemampuan untuk belajar dan beradaptasi terhadap lingkungannya dengan melakukan penyesuaian internal secara otomatis. Ada berbagai bentuk struktur yang dimiliki JST dan yang paling umum digunakan adalah *Feed-forward Neural Networks* (FFNN). FFNN secara umum terdiri dari tiga kelompok layer, yaitu *input layer*, *hidden layer* dan *ouput layer* dan terkoneksi dengan elemen-elemen non-linear yang dinamakan dengan neuron. Informasi dalam JST diproses melalui koneksi yang dilakukan pada neuron tersebut.

Gambar 1 menunjukkan struktur FFNN yang terdiri dari satu *hidden layer*. Aktivitas neuron di hidden layer ditentukan oleh sejumlah sinyal input x_k , dimana $k = 1, 2, \dots, i$. dan synaptic weight w_{ki} , dimana $l = 1, 2, \dots, j$. Output untuk neuron pada *hidden layer* (hl) digambarkan pada persamaan (1), berikut ini.

$$h_i = f\left(\sum_{k=1}^i x_k w_{ki} + w_0\right). \quad (1)$$



Sumber: Karayiannis, N.B. (1992)

Gambar 1 Struktur Feed-forward Neural Network

Dengan cara yang sama, output untuk neuron pada output layer (ymp) dapat dihasilkan dari output pada hidden layer dan *synaptic weight* antara *hidden layer* dengan *output layer* (v_{im}), sebagaimana digambarkan pada persamaan (2) berikut ini.

$$y_{mp} = f\left(\sum_{i=1}^j h_i v_{im} + v_0\right) \quad (2)$$

dimana, aktivasi fungsi yang digunakan pada teori ini adalah fungsi sigmoid:

$$f(s) = \frac{1}{(1 + e^{-as})}, \quad a > 0 \quad (3)$$

dimana,

- i : jumlah input
- j : jumlah neuron di *hidden layer*
- m : jumlah neuron di *output layer*
- w_0 dan v_0 : *threshold*
- p : jumlah *pattern*
- a : parameter fungsi sigmoid

1. Backpropagation

Penyesuaian (*adjustment*) *synaptic weight* dilakukan dengan tujuan untuk meminimalkan perbedaan antara output pada neuron di *output layer* dengan output yang diharapkan, yang disebut juga dengan nama *teacher's signal* (y_{dmp}). Penyesuaian *synaptic weight* yang dilakukan berdasarkan prinsip *backpropagation* digambarkan pada persamaan (4) dan (5) berikut ini (Karayiannis, N.B., et.al, 1992)

$$\Delta v_{im}(t) = -\mu_1 \frac{\partial E_m(t)}{\partial v_{im}(t)} + \alpha_1 \Delta v_{im}(t-1) \quad (4)$$

$$\Delta w_{kl}(t) = -\mu_2 \frac{\partial E_m(t)}{\partial w_{kl}(t)} + \alpha_2 \Delta w_{kl}(t-1) \quad (5)$$

Dimana,

- t : Jumlah Iterasi
- μ : *Learning Rate*
- α : *Momentum Rate*

2. Algoritma Adaptif

Algoritma Adaptif adalah algoritma yang digunakan untuk menyesuaikan *weight* yang berada di posisi antara *input layer* dan *hidden layer* serta antara *hidden layer* dan *output layer*. Berbeda dengan *Backpropagation*, sinyal error (*error signal*) yang digunakan pada algorithm Adaptif berasal dari perbedaan (*gap*) antara sinyal neuron (*output layer*) sebelum melewati fungsi aktivasi dan inverse fungsi sigmoid sinyal guru (*teacher's signal*), sebagaimana ditunjukkan pada persamaan (6).

$$e_{Mmp} = f^{-1}(y_{Dmp}) - s_{mp}. \quad (6)$$

dimana, $f^{-1}(y_{Dmp})$ adalah *inverse* fungsi sigmoid,

y_{Dmp} : sinyal guru (*teacher's signal*)

$$s_{mp} = \sum_{l=1}^j v_{lm} h_{lp} = V_m^T H_p, \quad (7)$$

dan,

$$V_m = [v_{1m}, v_{2m}, \dots, v_{jm}]^T$$

$$H_p = [h_{1p}, h_{2p}, \dots, h_{jp}]^T.$$

Kuadrat sinyal error untuk semua pola didefinisikan sebagaimana pada persamaan (8).

$$E_{Mm} = \frac{1}{2} \sum_p (e_{Mmp})^2 \quad (8)$$

Penyesuaian *weight* dilakukan dengan meminimalisir kuadrat sinyal error menggunakan metode gradien sebagaimana pada persamaan (9).

$$\frac{\partial E_{Mm}}{\partial V_m} = \sum_p H_p H_p^T V_k - \sum_p H_p f^{-1}(y_{Dmp}) = 0. \quad (9)$$

Selanjutnya didapatkan persamaan (10), yang digunakan untuk melakukan penyesuaian terhadap *weight* antara *output layer* dengan *hidden layer*.

$$\Delta v_{mp}(t) = \frac{\Gamma_{p-1} H_p}{1 + H_p^T \Gamma_{p-1} H_p} e_{Mmp} + \alpha_1 \Delta v_{mp}(t-1). \quad (10)$$

Dimana,

Γ_{p-1} : gain matrix

Dengan cara yang sama, penyesuaian *weight* antara *hidden layer* dan *input layer* dilakukan dengan mengikuti persamaan (11).

$$\Delta w_{kl}(t) = -\eta \frac{\partial E_m(t)}{\partial w_{kl}(t)} + \alpha_2 \Delta w_{kl}(t-1) \quad (11)$$

Dimana,

t : Jumlah Iterasi
 η : Learning Rate
 α : Momentum Rate

METODE PENELITIAN

1. Pengolahan Data Input

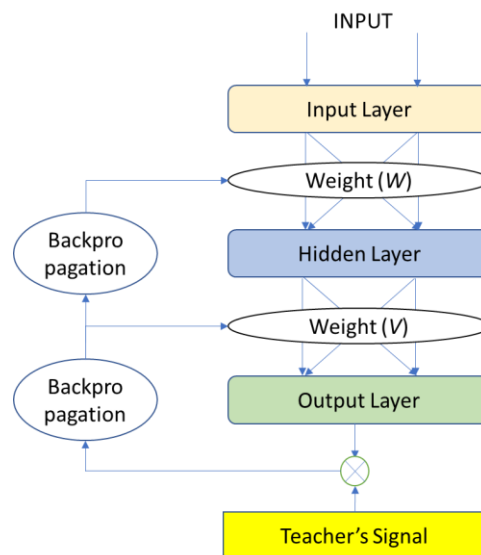
Pada penelitian kali ini, obyek permasalahan yang digunakan adalah Exclusi Or, dimana nilai On "1" dan Off "0" diset atau dinormalisasi mengikuti pada persamaan (12), (Siregar, 2017).

$$x' = \frac{0.8(x - \min \text{value})}{\max \text{value} - \min \text{value}} + 0.1 \quad (12)$$

2. Desain Proses Training

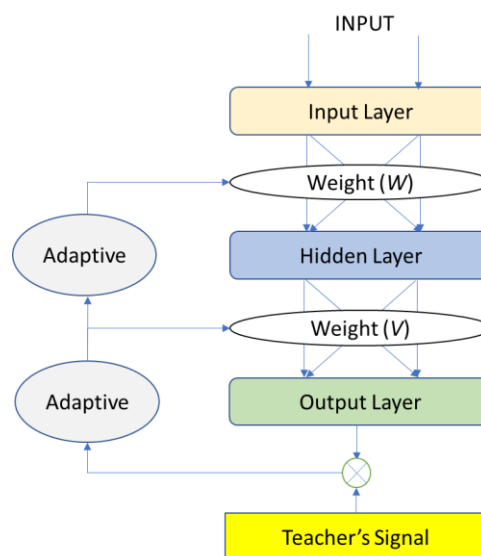
Proses training pada Backpropagation dilakukan dengan melakukan proses penyesuaian

pada *weight* baik yang ada di antara *output layer* dan *hidden layer* maupun pada *weight* yang ada di antara *hidden layer* dan *input layer*. Secara umum proses training untuk *Backpropagation* digambarkan pada Gambar 2.



Gambar 2 Proses Training Pada Backpropagation

Sedangkan proses training pada algoritma Adaptive, proses penyesuaian *weight* di antara *output layer* dengan *hidden layer* dan antara *hidden layer* dengan *input layer* dilakukan dengan menggunakan algoritma Adaptif. Secara umum proses training untuk algoritma Adaptif digambarkan pada Gambar 3.



Gambar 3. Proses Training Pada Algoritma Adaptif

3. Penentuan Parameter dan Struktur Jaringan Syaraf Tiruan (JST)

Setelah penentuan algoritma belajar dan proses training yang akan dijalankan maka selanjutnya menentukan nilai untuk variabel yang digunakan seperti *Learning Rate* dan *Momentum Rate*. Begitu juga terhadap penentuan struktur JST seperti *input layer*, *hidden layer* dan *output layer*, termasuk jumlah neuron pada tiap-tiap layer.

4. Eksperimen

Untuk melihat kinerja *Backpropagation* dan algoritma Adaptif, maka dilakukan pengujian dengan langkah-langkah sebagai Berikut.

1) Penentuan struktur JST (menyesuaikan dengan objek permasalahan) berupa Feedforward Neural Network, yang terdiri dari 1 input layer, 1 hidden layer dan 1 output layer. Untuk input dan hidden layer, masing-masing terdiri dari 2 neuron dan output layer 1 neuron.

2) Penentuan nilai variabel seperti *Momentum Rate* dan *Learning Rate*, yang pada penelitian ini divariasikan untuk melihat pengaruhnya pada kinerja *Backpropagation* dan algoritma Adaptif, sebagaimana dijelaskan pada bagian Hasil dan Pembahasan.

3) Inisialisasi nilai *weight* dengan menggunakan nilai random.

4) Meneruskan sinyal dari semua neuron pada input layer ke semua neuron pada hidden layer. Neuron pada hidden layer menjumlahkan semua sinyal yang masuk bersama dengan weight masing-masing untuk diteruskan lagi pada neuron di output layer sesudah melewati fungsi aktivasi. Selanjutnya neuron yang ada pada output layer menjumlahkan semua sinyal yang masuk bersama dengan weight masing-masing

5) Penyesuaian weight

a) *Backpropagation*

Sinyal dari neuron pada output layer, sesudah melewati fungsi aktivasi akan membentuk sinyal error (error signal), sebagaimana pada persamaan.... Dengan menggunakan sinyal error tersebut, *Backpropagation* melakukan penyesuaian weight dengan memakai persamaan (4) dan (5). (Lihat penjelasan bagian Jaringan Syaraf Tiruan)

b) Algoritma Adaptif

Berbeda dengan *Backpropagation*, sinyal error yang digunakan bersal dari sinyal neuron pada output layer yang sebelum melewati fungsi aktivasi, sebagaimana ditunjukkan pada persamaan.... Dengan menggunakan sinyal error tersebut penyesuaian weight dilakukan dengan memakai persamaan (10) dan (11). (Lihat penjelasan bagian Jaringan Syaraf Tiruan)

6) Evaluasi

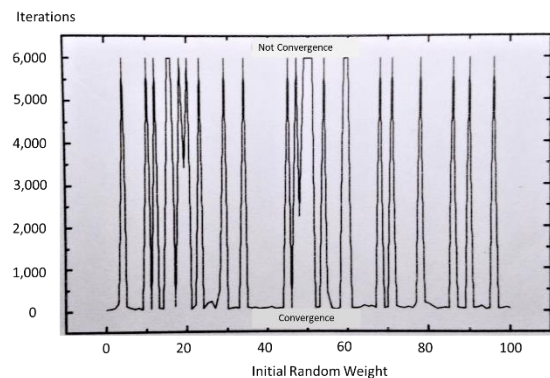
Langkah 4) dan 5) dilakukan secara kontinu dan dievaluasi menggunakan nilai *Root Mean Square Error* (RMSE), sebagaimana ditunjukkan pada persamaan (13) (Lee, 1997).

$$MSE = \sqrt{\frac{\sum(\text{Aktual}-\text{Prediksi})^2}{n}} \quad (13)$$

HASIL DAN PEMBAHASAN

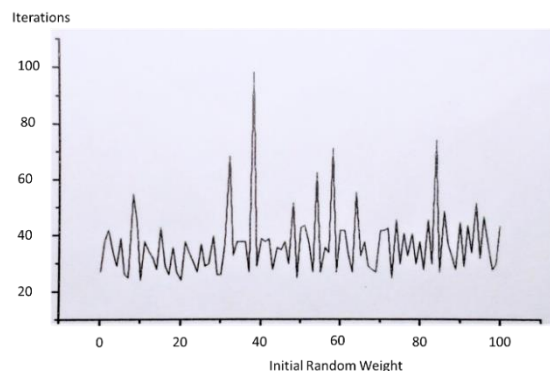
1. Sensitifitas Terhadap Kondisi Awal Weight

Untuk melihat sensitifitas algoritma *Backpropagation* terhadap kondisi awal weight maka dilakukan pengujian sebanyak 100 kali, dengan menggunakan nilai random. Dari 100 kali pengujian tersebut, sebagaimana digambarkan pada gambar 4, *Backpropagation* mengalami kegagalan dalam proses belajar sebanyak 33 kali dan keberhasilan sebanyak 77 kali.



Gambar 4. Tingkat Keberhasilan *Backpropagation*

Pengujian yang sama dilakukan terhadap algoritma Adaptif. Nilai awal weight dipilih secara random dan dilakukan trial sebanyak 100 kali dan hasilnya, sebagaimana ditunjukkan pada gambar 5, proses belajar mengalami konvergensi atau kesuksesan sebanyak 100 kali, atau 100% dengan tingkat iterasi yang jauh lebih singkat dibandingkan dengan algoritma *Backpropagation*.



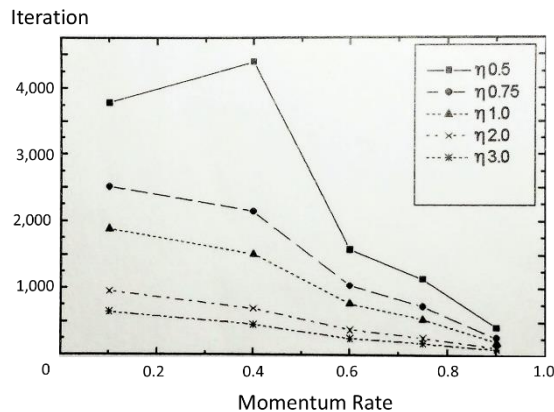
Gambar 5. Tingkat Keberhasilan Algoritma Adaptif

Tabel 1 Sensitifitas Terhadap Kondisi Awal Weight

Algoritma	Pengujian	Keberhasilan	Rata-Rata Iterasi
BP	100 X	77%	162
Adaptif	100 X	100%	37

2. Sensitifitas Terhadap Beberapa Parameter

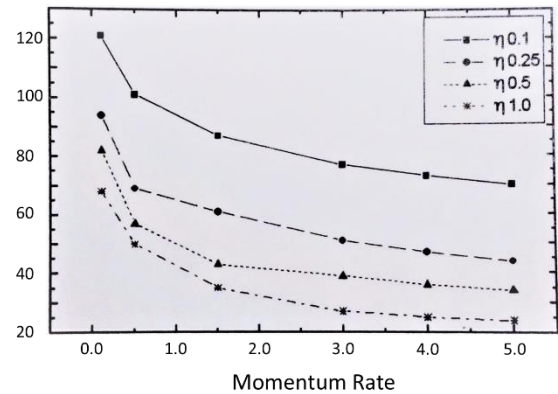
Pengujian juga dilakukan untuk melihat sejauh mana perubahan pada nilai parameter yang digunakan dalam proses belajar, memepengaruhi kecepatan proses belajar. Ada dua parameter yang diuji pada kali ini yaitu *Learning Rate* dan *Momentum Rate*. Sebagaimana pada Gambar 4, *Learning Rate* pada *Backpropagation* divariasikan dari 0,5 sampai 3,0. Sedangkan *Momentum Rate* divariasikan dari 0,1 sampai dengan 0,9. Sebagaimana pada Gambar 6, hasil pengujian menunjukkan bahwa semakin tinggi nilai *Momentum Rate* dan semakin tinggi nilai *Learning Rate*, proses belajar pada *Backpropagation* cenderung semakin cepat. Angka tercepat pada pengujian kali ini yaitu sebanyak 29 iterasi pada saat *Learning Rate* berada di posisi 3,0 dan *Momentum Rate* di posisi 0,9.



Gambar 6. Perubahan Parameter Terhadap Jumlah Iterasi Pada Bckpropagation

Pengujian sejenis juga dilakukan terhadap algoritma Adaptif, dimana *Learning Rate* divariasikan pada nilai 0,1 sampai 1,0. Sedangkan *Momentum Rate* disebar pada dari nilai 0,1 sampai 5,0. Sebagaimana ditunjukkan pada Gambar 7, semakin tinggi nilai *Learning Rate* dan *Momentum Rate* maka proses belajar juga semakin cepat. Proses tercepat dicapai dengan 24 kali iterasi pada posisi dimana *Learning Rate* sama dengan 1,0 dan *Momentum Rate* sama dengan 5,0.

Iteration



Gambar 7. Perubahan Parameter Terhadap Jumlah Iterasi Pada Algoritma Adaptif

3. Kecepatan Proses Training

Tabel 1 pada kolom “Rata-Rata Iterasi”, menunjukkan perbandingan kecepatan proses training secara rata-rata yang dilakukan antara *Backpropagation* dan algoritma Adaptif. Pada *Backpropagation* dari 77 kali keberhasilan melakukan proses training, jumlah rata-rata iterasi yang dibutuhkan sebanyak 162 kali. Sedangkan pada algoritma Adaptif, dari 100 kali keberhasilan training, rata-rata iterasi yang dibutuhkan sebanyak 37 kali. Dari sisi jumlah iterasi, algoritma Adaptif membutuhkan jauh lebih sedikit iterasi dalam proses training untuk permasalahan yang diujikan pada kali ini, dibandingkan dengan *Backpropagation*.

KESIMPULAN

Backpropagation dengan struktur Jaringan Syaraf Tiruannya berupa *Feedforward Neural Networks* (FFNN) atau dikenal juga dengan nama *Multilayer Perceptron* (MLP), cukup banyak diaplikasikan di berbagai bidang. Akan tetapi sebagaimana ditunjukkan pada hasil pengujian, *Backpropagation* memiliki beberapa kelemahan seperti: sensitif terhadap kondisi awal nilai weight dan juga sensitif terhadap perubahan nilai pada *Learning Rate* dan *Momentum Rate*. Sensitifitas ini perlu diantisipasi karena dampaknya bisa sampai pada terjadinya kegagalan proses training.

Algoritma yang diusulkan (Algoritma Adaptif), memberikan harapan dalam mengatasi kelemahan yang dihadapi oleh *Backpropagation*. Sebagaimana dilaporkan pada hasil pengujian, apabila dibandingkan dengan *Backpropagation*, algoritma Adaptif jauh lebih kuat menghadapi variasi terhadap parameter-parameter yang digunakan dalam proses training. Lebih dari itu, algoritma Adaptif juga menunjukkan kecepatan dalam proses training dibandingkan dengan

Backpropagation dilihat dari sisi jumlah iterasi yang dibutuhkan untuk konvergensi.

Menimbang aplikasi *Feedforward Neural Network* yang begitu luas di berbagai bidang industri, maka sangat terbuka kemungkinan untuk mengaplikasikan algoritma Adaptif sebagai alternatif algoritma belajar daripada *Backpropagation*..

REFERENSI

- Abdul Hamid, N., Mohd Nawi, N., Ghazali, R., & Mohd Salleh, M. N. (2012). A review on improvement of back propagation algorithm. *Global Journal on Technology*, 1. <http://archives.un-pub.eu/index.php/P-ITCS/article/viewFile/742/1024>
- Cilimkovic, M. (2015). Neural networks and back propagation algorithm. Institute of Technology Blanchardstown, Blanchardstown Road North Dublin, 15(1).
- Dasuki, M. (2021). Optimasi Nilai Bobot Algoritma Backpropagation Neural Network Dengan Algoritma Genetika. *JUSTINDO (Jurnal Sistem dan Teknologi Informasi Indonesia)*, 6(1),38-44. <http://jurnal.unmuhjember.ac.id/index.php/JUSTINDO/article/download/5280/3297>
- Ismail, M. I. S., Okamoto, Y., & Okada, A. (2013). Neural Network Modeling for Prediction of Weld Bead Geometry in Laser Microwelding. *Advances in Optical Technologies*. <https://downloads.hindawi.com/archive/2013/415837.pdf>
- Karayannis, N.B., Venetsopouloos, A.N. (1992). *Artificial Neural Networks*. Kluwer Academic Publisher.
- Lee, Charles W., (1997). Training Feedforward Neural Networks: An Algorithm Giving Improved Generalization.” *Neural Networks* 10 (1): 61–68. [https://doi.org/10.1016/S08936080\(96\)00071-8](https://doi.org/10.1016/S08936080(96)00071-8).
- Ma, J., Zhu, S. G., Wu, C. X., & Zhang, M. L. (2009). Application of back-propagation neural network technique to high-energy planetary ball milling process for synthesizing nanocomposite WC–MgO powders. *Materials & Design*, 30(8), 2867-2874.
- Ray, S. (2019, February). A quick review of machine learning algorithms. In 2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon) (pp. 35-39). IEEE. <https://translateyar.ir/wp-content/uploads/2021/12/A-Quick-Review-of-Machine.pdf>
- Siregar, Muhammad Noor Hasan, (2017). Neural Network Analysis With Backpropogation In Predicting Human Development Index (HDI) Component by Regency/City In North Sumatera.” *IJISTECH (International Journal Of Information System & Technology)* 1 (1): 22. <https://doi.org/10.30645/ijistech.v1i1.3>
- Wang, H. S. (2007). Application of BPN with feature-based models on cost estimation of plastic injection products. *Computers & Industrial Engineering*, 53(1), 79-94.
- Yi, X. (2015, August). Selection of initial weights and thresholds based on the genetic algorithm with the optimized back-propagation neural network. In 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD) (pp. 173-177). IEEE.