

Backpropagation with BFGS Optimizer for Covid-19 Prediction Cases in Surabaya

Prediksi Kasus Covid-19 Kota Surabaya Menggunakan *Backpropagation* dengan BFGS Optimizer

Zuraidah Fitriah^{1*}, Mohamad Handri Tuloli^{2*}, Syaiful Anam³, Noor Hidayat⁴, Indah Yanti⁵, Dwi Mifta Mahanani⁶

^{1,2,3,4,5,6} Jurusan Matematika, Universitas Brawijaya, Indonesia

^{1*} zuraidahfitriah@ub.ac.id, ^{2*} mohamadhandri18@gmail.com

Informasi Artikel

Received: March 2021

Revised: May 2021

Accepted: June 2021

Published: August 2021

Abstract

Covid-19 is a new type of corona virus called SARS-CoV-2. One of the cities that has contributed the most to infected Covid-19 cases in Indonesia is Surabaya, East Java. Predicting the Covid-19 is the important thing to do. One of the prediction methods is Artificial Neural Network (ANN). The backpropagation algorithm is one of the ANN methods that has been successfully used in various fields. However, the performance of backpropagation is depended on the architecture and optimization method. The standard backpropagation algorithm is optimized by gradient descent method. The Broyden - Fletcher - Goldfarb - Shanno (BFGS) algorithm works faster then gradient descent. This paper was predicting the Covid-19 cases in Surabaya using backpropagation with BFGS. Several scenarios of backpropagation parameters were also tested to produce optimal performance. The proposed method gives better results with a faster convergence then the standard backpropagation algorithm for predicting the Covid-19 cases in Surabaya.

Keywords: Covid-19, Surabaya, prediction, backpropagation, BFGS
Kata kunci: Covid-19, Surabaya, prediksi, backpropagation, BFGS

Abstrak

Corona Virus Disease 2019 (Covid-19) merupakan tipe baru dari virus corona yang disebut Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2). Indonesia sempat menjadi negara dengan kasus terinfeksi Covid-19 tertinggi di Asia Tenggara. Salah satu kota di Indonesia yang menjadi penyumbang kasus terinfeksi terbanyak adalah Kota Surabaya, Jawa Timur. Memprediksi kasus Covid-19 merupakan langkah awal untuk mengetahui kondisi terburuk yang mungkin terjadi

dan langkah terbaik untuk menekan kasus Covid-19. Banyak metode prediksi yang telah ditemukan, salah satunya Jaringan Syaraf Tiruan (JST). Algoritma backpropagation merupakan salah satu metode JST yang telah berhasil digunakan di berbagai bidang. Namun performa algoritma backpropagation bergantung pada arsitektur dan metode optimisasi yang digunakan. Algoritma backpropagation standar dioptimalkan dengan metode gradient descent. Algoritma Broyden – Fletcher – Goldfarb – Shanno (BFGS) bekerja lebih cepat daripada metode gradient descent. Makalah ini mengusulkan prediksi kasus Covid-19 di Kota Surabaya menggunakan data pada rentang 23 Maret hingga 15 November 2020 menggunakan JST backpropagation yang dioptimasi dengan algoritma BFGS. Beberapa skenario parameter backpropagation juga diuji untuk menghasilkan metode backpropagation dengan kinerja yang optimal. Berdasarkan hasil percobaan, algoritma backpropagation yang dioptimasi oleh algoritma BFGS memberikan hasil yang lebih baik dengan konvergensi yang lebih cepat dibandingkan dengan algoritma backpropagation standar untuk prediksi kasus Covid-19 di Kota Surabaya.

1. Pendahuluan

Corona Virus Disease 2019 (Covid-19) merupakan tipe baru dari virus corona yang disebut *Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2)*. Virus ini bertransmisi atau ditularkan dari manusia terinfeksi ke manusia sehat melalui udara. Covid-19 diketahui pertama kali menginfeksi manusia di Kota Wuhan, China pada Desember 2019 [5]. Pada Maret, 2020, WHO menyatakan Covid-19 sebagai suatu pandemi karena telah menginfeksi lebih dari 150 negara termasuk Indonesia. Indonesia sempat menjadi negara dengan kasus terinfeksi Covid-19 tertinggi di Asia Tenggara. Salah satu kota di Indonesia yang menjadi penyumbang kasus terinfeksi terbanyak adalah Kota Surabaya, Jawa Timur. Berdasarkan Data Harian Tanggap Covid-19 Surabaya hingga pertengahan November, 2020, kasus Covid-19 di Surabaya mencapai lebih dari 15.000 kasus positif dengan lebih dari 1000 kasus kematian [6]. Banyak faktor yang mempengaruhi kecepatan penyebaran Covid-19, salah satunya adalah taat atau tidaknya penduduk terhadap kebijakan penanganan Covid-19 yang disosialisasikan pemerintah.

Banyaknya kasus dan korban Covid-19 membuat banyak peneliti mencoba untuk memodelkan dan meneliti Covid-19 dengan mempertimbangkan sudut pandang yang berbeda-beda mulai dari aspek psikologis, perilaku masyarakat, cuaca dan lain-lain. [5] mencoba membahas aspek psikologis yang diakibatkan oleh Covid-19, [12] menjelaskan korelasi antara cuaca dan pandemi Covid-19 di Jakarta, [14] memodelkan, menganalisis dan menyimulasikan model SEIR penyebaran Covid-19 di Indonesia, [1] memprediksi kasus Covid-19 di Mesir, [4] memodelkan

dan menganalisis pengaruh *social distancing* dan *rapid assesment* terhadap penyebaran Covid-19 di Jakarta, serta banyak penelitian lainnya seperti [2], [3], [10], dan [11].

Sejak pertama kali Covid-19 menyebar ke seluruh negara telah banyak usaha yang telah dilakukan seperti pencegahan dengan menjaga jarak, memakai masker, serta menjaga imun tubuh. Hal penting yang dapat dilakukan di situasi pandemi seperti sekarang adalah semaksimal mungkin menekan angka penyebaran Covid-19. Oleh karena itu, salah satu langkah awal yang bisa dilakukan adalah dengan memprediksi kasus Covid-19. Berdasarkan prediksi kita dapat mempelajari dan mengaproksimasi perkembangan Covid-19 ke depan dalam jangka pendek. Kemudian dapat diketahui pula situasi terburuk dan langkah terbaik yang bisa dilakukan untuk paling tidak menekan angka penyebaran Covid-19.

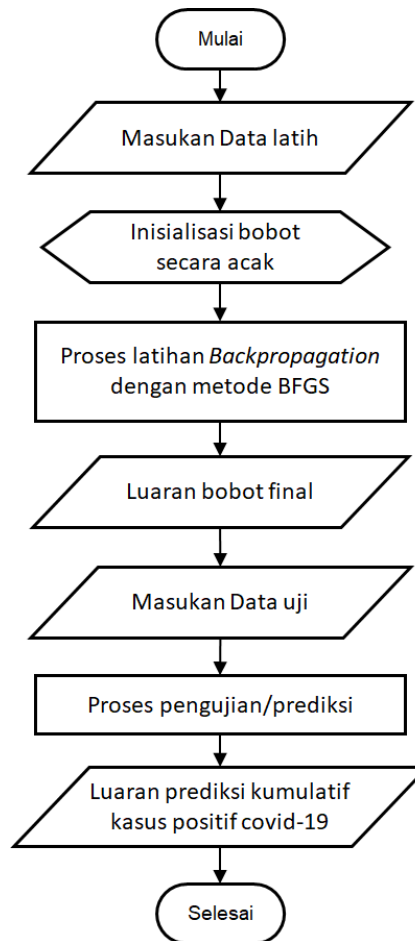
Salah satu metode yang populer untuk memprediksi adalah jaringan syaraf tiruan (JST). JST merupakan metode yang tidak membutuhkan asumsi dengan performa yang tetap baik, sehingga lebih fleksibel. Terdapat banyak kasus yang berhasil diaproksimasi menggunakan JST di antaranya prediksi curah hujan [13] prediksi tingkat kematian akibat Covid-19 [15], melihat korelasi beberapa faktor pada konsumsi listrik di sebuah bangunan [7], desain kurikulum [8] serta masih banyak lagi. Metode jaringan syaraf tiruan dapat mempelajari pola kasus Covid-19 sehingga dapat memprediksi kasus di masa depan. *Backpropagation* adalah salah satu metode jaringan syaraf tiruan dengan banyak lapisan yang bergantung pada *node* dan bobot yang digunakan. Pada *backpropagation* konvensional, bobot yang digunakan dioptimasi dengan *gradient descent* secara iteratif. Namun, metode *gradient descent* tidak dapat bekerja secara maksimum karena membutuhkan iterasi yang relatif lebih banyak. Menurut [13] metode Quasi Newton Broyden-Fletcher-Goldfarb-Shanno (BFGS), merupakan opsi yang dapat digunakan untuk mengatasi kelemahan dari *gradient descent* karena lebih cepat konvergen dengan iterasi yang relatif lebih sedikit dibandingkan *gradient descent*. Di samping itu telah banyak penelitian yang menggunakan atau mengembangkan optimasi BFGS seperti [9] dan [13]

Pada paper ini, kasus Covid-19 di kota Surabaya akan diprediksi menggunakan JST *backpropagation* dengan metode BFGS. Hasil eksperimen kemudian dibandingkan dengan hasil prediksi menggunakan *backpropagation* dengan *gradient descent*.

2. Metode

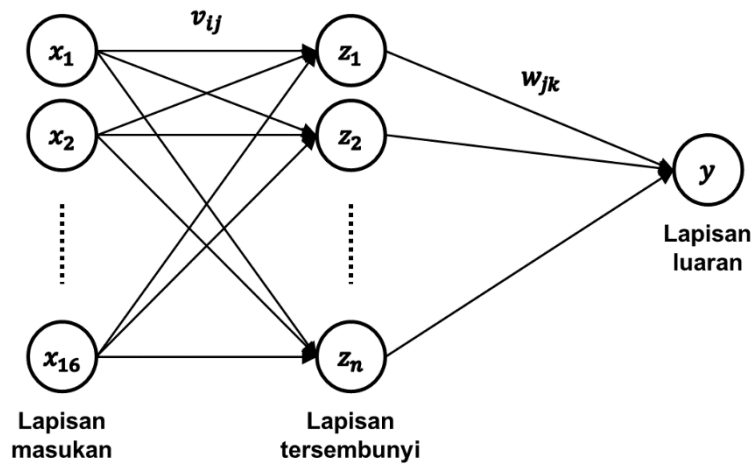
Langkah awal, dipilih 90% data awal Covid-19 pada rentang 23 Maret hingga 15 November 2020 dan dijadikan sebagai data latih (*training data*) dan 10% sisanya dijadikan sebagai data uji prediksi. Data Covid-19 yang digunakan merupakan kumulatif terkonfirmasi positif, kumulatif kematian, dan kumulatif kesembuhan. Bobot awal pada *backpropagation* mula-mula dibangkitkan secara acak. Selanjutnya *backpropagation* akan melakukan langkah latihan (*train*) pada data latih. Pada langkah ini, bobot pada *backpropagation* diubah dan ditingkatkan hingga menghasilkan bobot yang optimum dengan meminimalkan selisih (*error*) keluaran *backpropagation* dan target. Selisih ini diminimumkan dengan metode BFGS. Bobot akhir yang didapatkan pada tahap latihan akan digunakan pada data uji prediksi. Pada langkah uji prediksi, data uji menjadi data masukan dan data keluaran merupakan representasi prediksi kumulatif kasus terkonfirmasi Covid-19. Diagram alir proses prediksi kumulatif kasus terkonfirmasi Covid-19 ditunjukkan pada **Gambar 1**.

Pada makalah ini, diasumsikan kumulatif kasus terkonfirmasi positif beberapa hari sebelumnya, kumulatif kematian, dan kumulatif kesembuhan akan berdampak pada kasus terkonfirmasi pada hari selanjutnya. Oleh karena itu, kumulatif kasus konfirmasi positif dua pekan berturut-turut, kumulatif kematian, dan kumulatif kesembuhan menjadi data masukan.



Gambar 1. Diagram alir proses prediksi kasus kumulatif terkonfirmasi Covid-19

Arsitektur JST dibuat menggunakan banyak variabel yang berdampak pada kasus terkonfirmasi Covid-19. Variabel tersebut termasuk kumulatif kasus terkonfirmasi positif pada dua pekan berturut-turut sebelum, kumulatif kematian, dan kumulatif kesembuhan atas Covid-19. Oleh karena itu, variabel yang digunakan sebagai data masukan ada sebanyak 16 variabel. Variabel x_1, x_2, \dots, x_{14} secara berturut-turut merupakan kumulatif kasus terkonfirmasi Covid-19 pada dua pekan hingga satu hari sebelumnya, x_{15} merupakan kumulatif kematian dan x_{16} merupakan kumulatif kesembuhan. Visualisasi arsitektur JST dapat dilihat pada **Gambar 2**. Terdapat tiga lapisan yaitu lapisan masukan, lapisan tersembunyi, dan lapisan luaran. Banyaknya lapisan tersembunyi ditentukan berdasarkan percobaan. Tujuannya untuk mendapatkan banyak lapisan tersembunyi yang dapat mengoptimalkan kinerja *backpropagation*. Skenario banyaknya lapisan tersembunyi yang digunakan pada makalah ini yaitu 16-5-1, 16-20-1, 16-50-1, 16-100-1, dan 16-150-1. Arti arsitektur model 16-150-1 adalah jaringan dibangun atas 1 neuron pada lapisan luaran, 150 neuron pada lapisan tersembunyi, dan 16 lapisan masukan.



Gambar 2. Arsitektur Jaringan Syaraf Tiruan *Backpropagation*

Proses latihan pada algoritma *backpropagation* terdiri atas tiga tahapan yaitu *feedforward* untuk data masukan, kesalahan *backpropagation*, dan penentuan bobot. Algoritma *backpropagation* dengan BFGS *Optimizer* ditunjukkan pada Algoritma 1.

Algoritma 1: *Backpropagation* dengan BFGS optimizer

Langkah 0. Inisialisasi bobot *backpropagation* dengan nilai acak dan $t = 1$ (iterasi ke- t)

Langkah 1. Selama $error > \text{minimum error}$ atau iterasi $<$ maksimum iterasi, lakukan langkah 2-8

Langkah 2. Untuk setiap data, lakukan langkah 3-8

Feedforward:

Langkah 3. Setiap unit masukan ($x_i, i = 1, 2, \dots, n$) didistribusikan ke unit lapisan tersembunyi.

Langkah 4. Setiap unit tersembunyi ($z_j, j = 1, 2, \dots, p$) menghitung bobotnya ($v_{ij}(t) = 1, 2, \dots, p$) dengan sebuah sinyal masukan pada Pers. (1).

$$z_{in_j} = v_{0j}(t) + \sum_i x_i v_{ij}(t) \tag{1}$$

dengan v_{0j} adalah bias pada unit tersembunyi j .

Hitung sinyal luaran dengan fungsi aktivasi pada Pers. (2) atau (3)

$$z_j = f(z_{in_j}) \tag{2}$$

$$z_j = \frac{1}{1 + e^{(-z_{in_j})}} \tag{3}$$

Kemudian kirimkan z_j ke semua unit luaran.

Langkah 5. Setiap unit luaran ($y_k, k = 1, 2, \dots, m$) dihitung berdasarkan sinyal masukan, yang ditunjukkan pada Pers. (4).

$$y_{in_k} = w_{0j}(t) + \sum_j z_j w_{jk}(t) \quad (4)$$

dengan w_{0j} adalah bias pada unit luaran k .

sinyal luaran dihitung dengan fungsi aktivasi pada Pers. (5) atau (6)

$$y_k = f(y_{in_k}) \quad (5)$$

atau

$$y_k = \frac{1}{1 + e^{(-y_{in_k})}} \quad (6)$$

Backpropagation error:

Langkah 6. Setiap unit luaran ($y_k, k = 1, 2, \dots, m$) menerima penyesuaian pola target dengan data masukan, dan menghitung *error* dengan Pers. (7).

$$\delta_k = (t_k - y_k) f'(y_{in_k}) \quad (7)$$

di mana, δ_k adalah *error* koreksi bobot w_{jk}

Langkah 7. Setiap unit tersembunyi ($z_j, j = 1, 2, \dots, p$) menghitung δ_k pada Pers. (8).

$$\delta_{in_j} = \sum_k \delta_k w_{jk} \quad (8)$$

$$\delta_j = \delta_{in_j} f'(z_{in_j}) \quad (9)$$

Memperbarui bobot dan bias:

Langkah 8. Setiap unit luaran ($y_k, k = 1, 2, \dots, m$) memperbarui bias dan bobot $j = 1, 2, \dots, p$ pada Pers. (10).

$$w_{jk}(t + 1) = w_{jk}(t) + \alpha C_k(t) (\delta_k z_j(t)) \quad (10)$$

di mana $C_k(t)$ adalah aproksimasi matriks Hessian. Pada kasus ini, matriks Hessian diperbarui dengan formula Pers. (11).

$$C_{k+1}(t+1) = C_k(t) - \frac{\Delta(-\delta_k z_j(t)) \Delta(-\delta_k z_j(t))^T}{\Delta(-\delta_k z_j(t))^T \Delta w_{jk}(t)} + \frac{(-\delta_k z_j(t)) (-\delta_k z_j(t))^T}{(-\delta_k z_j(t))^T (\Delta w_{jk}(t) / \alpha)} \quad (11)$$

Setiap unit tersembunyi ($z_j, j = 1, 2, \dots, p$) memperbarui bias dan bobot $i = 1, 2, \dots, n$ berdasarkan persamaan Pers. (12)

$$v_{ij}(t+1) = v_{ij}(t) + \alpha G_j(t) (\gamma_j x_i(t)) \quad (12)$$

di mana $G_j(t)$ di definisikan pada Pers. (13).

$$G_{k+1}(t+1) = G_k(t) - \frac{\Delta(-\gamma_j x_i(t)) \Delta(-\gamma_j x_i(t))^T}{\Delta(-\gamma_j x_i(t))^T \Delta v_{ij}(t)} + \frac{(-\gamma_j x_i(t)) (-\gamma_j x_i(t))^T}{(-\gamma_j x_i(t))^T (\Delta v_{ij}(t) / \alpha)} \quad (13)$$

Langkah 9. Mengecek kriteria pemberhentian

Makalah ini menggunakan fungsi sigmoid sebagai fungsi aktivasi pada lapisan tersembunyi dan lapisan luaran. *Learning rate* (α) yang digunakan bersifat statis. *Learning rate* digunakan untuk melihat bagaimana pengaruh yang diberikan pada kemampuan JST dalam mengenali pola. Pada kasus ini, beberapa skenario α digunakan yaitu 0.001, 0.005, 0.01, 0.1, dan 0.2 untuk menganalisis dampak dari *learning rate*.

Kriteria pemberhentian yang digunakan adalah maksimum 1000 iterasi, performa 1E-05, dan gradien 1E-75, yang berarti ketika pembelajaran JST mencapai maksimum iterasi atau batas performa atau batas gradien, proses pembelajaran akan dihentikan, sebaliknya proses pembelajaran akan terus berlanjut.

3. Hasil dan Pembahasan

Evaluasi prediksi kasus Covid-19 kota Surabaya menggunakan data yang diambil dari data harian Surabaya Tanggap COVID-19. Data yang diambil merupakan data sejak 23 Maret hingga 13 November 2020. Data dipisahkan menjadi dua bagian yaitu data latih dan data uji. 90% dari data awal merupakan data latih serta 10% sisanya merupakan data uji. Eksperimen menggunakan perangkat keras dengan prosesor *Core i5*, 3.4 GHz, sistem operasi *windows 10*, dan perangkat lunak MATLAB R2014b. Perbandingan performa BFGS dan *gradient descent* dalam mengoptimasi JST *backpropagation* dengan arsitektur yang berbeda ditunjukkan pada **Tabel 1** dan **Tabel 2**. Akurasi yang dihasilkan oleh metode dengan BFGS lebih tinggi dibandingkan *gradient descent*. Semakin banyak lapisan tersembunyi yang digunakan semakin lama waktu komputasi yang dibutuhkan. Jika lapisan tersembunyi yang digunakan semakin banyak maka kemungkinan besar terjadi *overfitting*, suatu kondisi yang mengakibatkan akurasi yang tinggi pada data latih namun akurasi yang rendah pada data uji. Sehingga pemilihan lapisan tersembunyi haruslah tidak terlalu sedikit dan tidak pula terlalu banyak. **Tabel 3** dan **Tabel 4** menunjukkan perbandingan performa metode BFGS dan *gradient descent* dalam memprediksi

kasus Covid-19 dengan skenario *learning rate* yang berbeda. *learning rate* dengan performa terbaik bernilai 0.01.

Tabel 1. Perbandingan *Mean Square Error (MSE) Backpropagation* dengan *Learning Rate* 0.005

| No. | Skenario arsitektur JST | MSE data latih | | MSE data uji | |
|-----|-------------------------|-------------------------|------------------|-------------------------|------------------|
| | | <i>Gradient descent</i> | BFGS | <i>Gradient descent</i> | BFGS |
| 1 | 16-5-1 | 2,600E-03 | 5,611E-05 | 9,783E-04 | 9,577E-06 |
| 2 | 16-20-1 | 2,177E-04 | 3,028E-05 | 1,227E-04 | 8,695E-06 |
| 3 | 16-50-1 | 5,504E-04 | 1,566E-05 | 3,522E-04 | 9,155E-06 |
| 4 | 16-100-1 | 2,000E-03 | 3,664E-05 | 1,100E-03 | 8,966E-06 |
| 5 | 16-150-1 | 1,500E-03 | 1,706E-05 | 8,306E-04 | 8,928E-06 |

Tabel 2. Perbandingan Waktu Komputasi *Backpropagation* dengan *Learning Rate* 0.005

| No. | Skenario arsitektur JST | Waktu Komputasi | |
|-----|-------------------------|-------------------------|---------------|
| | | <i>Gradient descent</i> | BFGS |
| 1 | 16-5-1 | 2,5846 | 1,6817 |
| 2 | 16-20-1 | 2,4137 | 2,1903 |
| 3 | 16-50-1 | 2,315 | 8,6481 |
| 4 | 16-100-1 | 3,5392 | 99,5206 |
| 5 | 16-150-1 | 4,0239 | 398,2719 |

Tabel 3. Perbandingan *Mean Square Error (MSE) Backpropagation* dengan 20 Lapisan Tersembunyi

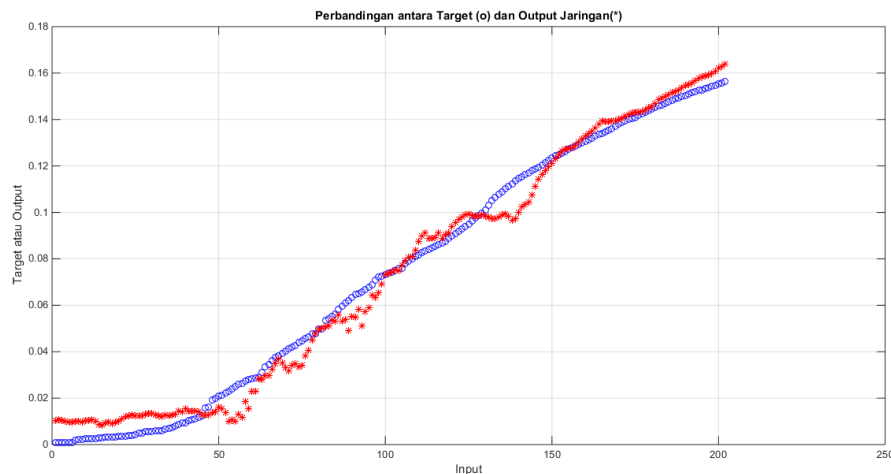
| No. | Skenario <i>learning rate</i> | MSE data latih | | MSE data uji | |
|-----|-------------------------------|-------------------------|------------------|-------------------------|------------------|
| | | <i>Gradient descent</i> | BFGS | <i>Gradient descent</i> | BFGS |
| 1 | 0,001 | 3,411E-04 | 1,412E-05 | 2,003E-04 | 9,095E-06 |
| 2 | 0,005 | 3,506E-04 | 3,761E-05 | 1,808E-04 | 9,047E-06 |
| 3 | 0,01 | 2,689E-04 | 3,392E-05 | 1,520E-04 | 8,899E-06 |
| 4 | 0,1 | 3,389E-04 | 1,204E-05 | 1,753E-04 | 9,428E-06 |
| 5 | 0,2 | 3,162E-04 | 2,111E-05 | 1,992E-04 | 9,232E-06 |

Tabel 4. Perbandingan Waktu Komputasi *Backpropagation* dengan 20 Lapisan Tersembunyi

| No. | Skenario arsitektur JST | Waktu Komputasi | |
|-----|-------------------------|-------------------------|---------------|
| | | <i>Gradient descent</i> | BFGS |
| 1 | 0,001 | 2,2514 | 2,4476 |
| 2 | 0,005 | 2,3475 | 2,1404 |
| 3 | 0,01 | 2,0566 | 2,2364 |
| 4 | 0,1 | 2,3707 | 2,2921 |
| 5 | 0,2 | 2,3513 | 2,1547 |

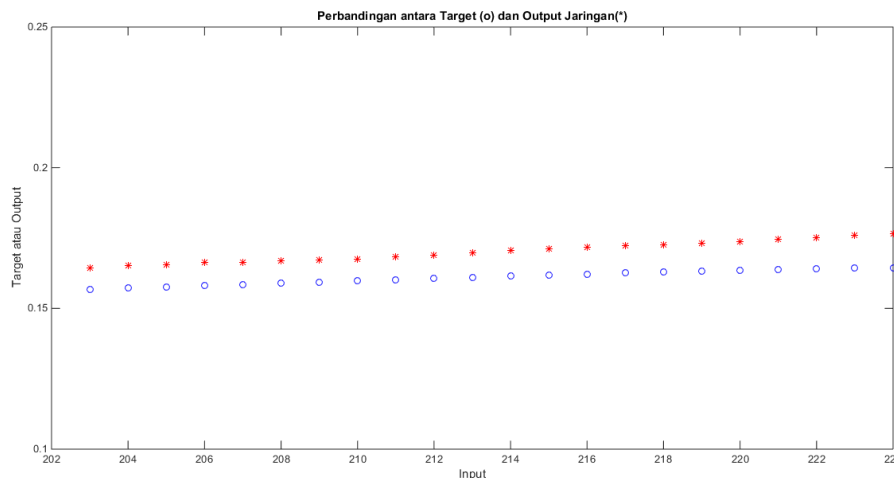
Performa *backpropagation* dengan metode BFGS menghasilkan MSE (*Mean Square Error*) $3,392E-05$ pada data latih dan $8.899E-06$ pada data uji. Sementara untuk *backpropagation* standar menghasilkan MSE $2,689E-04$ pada data latih dan $1,520E-04$ pada data uji. **Gambar 3** dan **Gambar 4** menunjukkan perbandingan data kasus Covid-19 asli dengan hasil luaran proses JST *backpropagation*. Berdasarkan Gambar tersebut terlihat bahwa *backpropagation* yang dioptimasi dengan metode BFGS menghasilkan performa yang lebih baik dibandingkan dengan *backpropagation* konvensional.

Perbandingan antara Target (o) dan Luaran Jaringan (*)



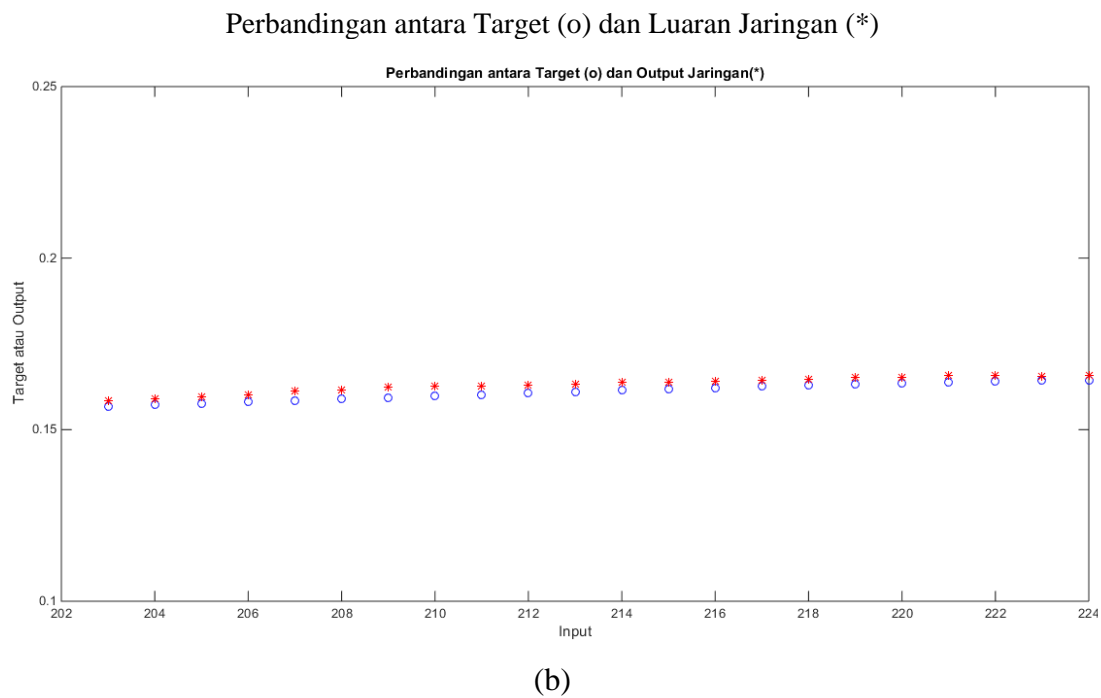
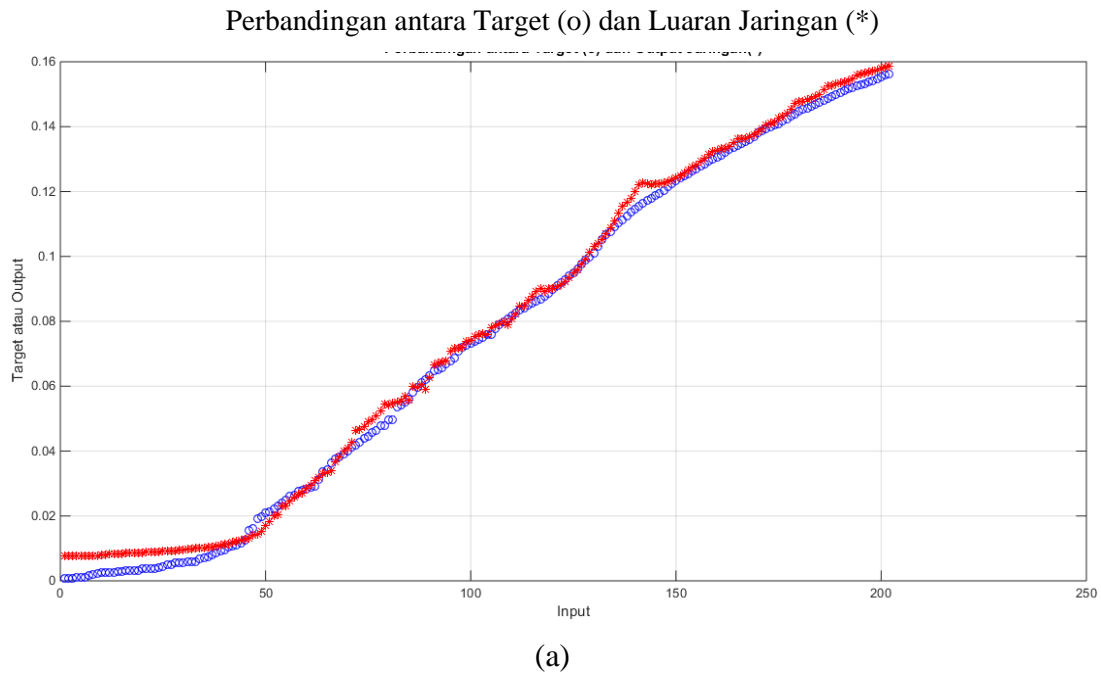
(a)

Perbandingan antara Target (o) dan Luaran Jaringan (*)



(b)

Gambar 3. Perbandingan data target dan luaran *backpropagation* dengan *gradient descent*. (a) data latih. (b) data uji



Gambar 4. Perbandingan data target dan luaran *backpropagation* dengan BFGS. (a) data latih. (b) data uji

4. Kesimpulan

Berdasarkan hasil eksperimen diperoleh bahwa jumlah lapisan tersembunyi dan parameter pembelajaran berpengaruh terhadap performa *backpropagation*. Jumlah lapisan tersembunyi haruslah tidak terlalu kecil dan tidak pula terlalu besar. Semakin banyak jumlah lapisan tersembunyi maka semakin lama waktu komputasi *backpropagation* dengan BFGS. Sedangkan waktu komputasi *backpropagation* dengan *gradient descent* tidak secara signifikan naik seperti halnya *backpropagation* dengan BFGS. Performa *backpropagation* dengan *gradient descent* maupun BFGS mencapai performa terbaik dengan *learning rate* yang tidak terlalu besar dan tidak terlalu kecil. Performa *backpropagation* dengan BFGS memiliki akurasi yang lebih baik dalam memprediksi kasus Covid-19 di kota Surabaya dibandingkan *backpropagation* dengan *gradient descent*.

Daftar Pustaka

- [1] A. I. Saba and A. H. Elsheikh, "Forecasting the prevalence of COVID-19 outbreak in Egypt using nonlinear autoregressive artificial neural networks," in *Process Safety and Environmental Protection*, pp. 1-8, 2020.
- [2] A. Rodríguez, G. Moreno, J. Gómez, R. Carbonell, E. Picó-Plana, C. B. Bofill, R. S. Parrilla, S. Trefler, E. E. Pitarch, L. Canadell, X. Teixido, L. Claverias, M. Bodí, and on behalf of the HJ23-COVID-19 working group, "Severe infection due to the SARS-CoV-2 coronavirus: Experience of a tertiary hospital with COVID-19 patients during the 2020 pandemic," in *Medicina Intensiva*, pp. 525-533, 2020.
- [3] D. Annane, N. Heming, L. Grimaldi-Bensouda, V. Fremeaux-Bacchi, M. Vigan, A.-L. Roux, A. Marchal, H. Michelon, M. Rottman, and P. Moine, "Eculizumab as an emergency treatment for adult patients with severe COVID-19 in the intensive care unit: A proof-of-concept study," in *EclinicalMedicine*, 2020.
- [4] D. Aldila, H. A. Sarbaz, Khoshnaw, E. Safitri, Y. R. Anwar, R. Q. Aanisah, Bakry, B. M. Samiadji, D. A. Anugerah, M. F. Alfarizi, I. D. Ayulani, and S. N. Salim, "A mathematical study on the spread of COVID-19 considering social distancing and rapid assessment: The case of Jakarta, Indonesia," in *Chaos, Solitons and Fractals*, 2020.
- [5] H. D. Windarwati, W. Oktaviana, I. Mukarromah, N. A. L. Ati, A. F. Rizzal, and A. D. Sulaksono, "In the middle of the COVID-19 outbreak: Early practical guidelines aspects of COVID-19 in East Java, Indonesia", in *Psychiatry Research*, 2020.
- [6] <https://lawanCovid-19.surabaya.go.id/visualisasi/graph> diakses pada 16 November 2020.
- [7] M. K. Kim, Y. S. Kim, and J. Srebric, "Impact of Correlation of Plug Load Data, Occupancy Rates and Local Weather Conditions on Electricity Consumption in a Building Using Four Back-propagation Neural Network Models Sustainable Cities and Society," 2020, doi: <https://doi.org/10.1016/j.scs.2020.102321>
- [8] M. Somasundaram, P. Latha, and S. A. S. Pandian, "Curriculum Design Using Artificial Intelligence (AI) Back Propagation Method," in *9th World Engineering Education*

- Forum, WEEF, 2019.*
- [9] N. Andrei 2017. "Double parameter scaled BFGS method for unconstrained optimization," in *Journal of Computational and Applied Mathematics*, 2017, doi: <https://doi.org/10.1016/j.cam.2017.10.009>
- [10] R. Djalante, J. Lassa, D. Setiamarga, A. Sudjatma, M. Indrawan, B. Haryanto, C. Mahfud, M. S. Sinapoy, S. Djalante, I. Rafliana, L. A. Gunawan, G. A. K. Surtiari, H. Warsilah, "Review and analysis of current responses to COVID-19 in Indonesia: Period of January to March 2020," in *Progress in Disaster Sciencs*, 2020.
- [11] R. Golpe, L. A. Pérez-de-Llano, D. Dacal, H. Guerrero-Sande, B. Pombo-Vide, P. Ventura-Valcárcel, and On behalf of the Lugo Covid-19 team, "Risk of severe COVID-19 in hypertensive patients treated with renin-angiotensin-aldosterone system inhibitors" in *Medicina Clinica*, pp. 488–490, 2020.
- [12] R. Tosepu, J. Gunawan, D. S. Effendy, L. O. A. I. Ahmad, H. Lestari, H. Bahar, and P. Asfian, "Correlation between weather and Covid-19 pandemic in Jakarta, Indonesia," in *Science of the Total Environment*, 2020.
- [13] S. Anam, "Rainfall prediction using backpropagation algorithm optimized by Broyden-Fletcher-Goldfarb-Shanno algorithm," in *IOP Conf. Series: Materials Science and Engineering*, 2019.
- [14] S. Annas, M. I. Pratama, M. Rifandi, W. Sanusi, and S. Side, "Stability analysis and numerical simulation of SEIR model for pandemic COVID-19 spread in Indonesia," in *Chaos, Solitons and Fractals*, 2020.
- [15] Z. Malki, E. Atlam, A. E. Hassanien, G. Dagnew, M. A. Elhosseini, and I. Gad, "Association between weather data and COVID-19 pandemic predicting mortality rate: Machine learning approaches," in *Chaos, Solitons and Fractals*, 2020.