

PENERAPAN ALGORITMA *BACKPROPAGATION* DAN OPTIMASI *CONJUGATE GRADIENT* UNTUK KLASIFIKASI HASIL TES LABORATORIUM

Wahyu Tiara Rosaamalia^{1*}, Rukun Santoso², Suparti³

^{1,2,3}Departemen Statistika, Fakultas Sains dan Matematika, Universitas Diponegoro

*e-mail: wtiarar@students.undip.ac.id

DOI: 10.14710/j.gauss.11.4.506-511

Article Info:

Received: 2022-07-28

Accepted: 2022-10-15

Available Online: 2023-02-25

Keywords:

Backpropagation; Conjugate Gradient, Blood; Classification

Abstract: A blood test is generally used to evaluate the condition of the blood and its components, conduct screening, and aid diagnosis. Blood tests in the laboratory are commonly used to deliberate whether a patient needs to be hospitalized or treated as an outpatient. Backpropagation algorithm was selected for its ability to solve complex problems. Conjugate gradient optimization is used because it facilitates faster solution search. An electronic medical record containing the results of patient laboratory examinations was obtained from Mendeley. The data was divided into training and testing with a 95:5 ratio, which was discovered to be the best ratio from the experiments. The best architecture was achieved by a combination of 10 neurons in the input layer, 16 neurons in the first hidden layer, 2 neurons in the second hidden layer, and a neuron in the output layer. Purelin is used as the activation function for both the first hidden and output layers, whereas the binary sigmoid is used for the second hidden layer. The analysis revealed that for 100 bootstraps in training data, the network worked with an average accuracy of 60.17% and a recall of 99.77%, while the accuracy results in testing data were 69.23%.

1. PENDAHULUAN

Pemeriksaan darah merupakan pemeriksaan yang sering diminta dalam rumah sakit untuk menginformasikan keadaan darah dan komponennya (Rosida & Hendriyono, 2015). Pemeriksaan darah digunakan untuk menerapkan diagnose, membuat diagnosis banding, dan menentukan pemeriksaan (Esa, *et al.*, 2006).

Salah satu metode yang dapat digunakan untuk mengklasifikasi hasil tes laboratorium adalah algoritma *backpropagation*. Algoritma ini bersifat adaptif dan dapat menghasilkan *output* dengan cepat, namun membutuhkan waktu yang lama untuk melakukan pelatihan (Fausett, 1994). *Conjugate gradient* dipilih sebagai salah satu metode optimasi yang dapat mempercepat proses pelatihan karena kemampuannya untuk mencapai konvergensi dengan lebih cepat (Wisesty, *et al.*, 2010).

2. TINJAUAN PUSTAKA

Backpropagation merupakan algoritma yang menggunakan nilai *error* pada *output* untuk mengubah bobotnya dalam arah mundur serta menggunakan fungsi aktivasi yang terdiferensiasi untuk langkah maju (Kusumadewi, 2004). Suatu jaringan syaraf tiruan biasanya terdiri dari lapisan *input* yang menerima masukan dari luar, lapisan *output* yang menghasilkan *output* dari jaringan, dan lapisan tersembunyi yang terletak diantara lapisan *input* dan *output* (Warsito, 2009). Langkah-langkah pelatihan dengan *backpropagation* dimulai dengan menginisialisasi bobot dengan bilangan acak kecil, kemudian dilanjutkan langkah berikut (Wanto, 2017):

1. Jika kondisi penghentian belum terpenuhi, lakukan langkah 2-9
2. Untuk setiap pasang data *training*, lakukan langkah 3-8
3. Tiap *input* menerima *sinyal* dan meneruskannya ke unit tersembunyi
4. Menghitung semua *output* di *hidden layer* dengan $Z_{netj} = v_{j0} + \sum_{i=1}^n X_i v_{ji}$ kemudian mengaplikasikan fungsi aktivasi $z_j = \psi(Z_{netj})$ untuk menghitung keluaran, kemudian mengirim sinyal ke seluruh lapisan unit keluaran
5. Menghitung semua jaringan di unit keluaran dengan menjumlahkan sinyal masukkan terboboti $Y_{netk} = w_{k0} + \sum_{j=1}^p z_j w_{kj}$ dan mengaplikasikan fungsi aktivasi untuk menghitung sinyal keluaran $y_k = \psi(Y_{netk})$
6. Menghitung faktor δ unit keluaran berdasarkan kesalahan di setiap unit keluaran $\delta_k = (t_k - y_k)\psi'(Y_{netk})$ kemudian menghitung faktor koreksi bobot untuk melakukan pembaruan terhadap w_{kj} dengan $\Delta w_{jk} = \alpha \delta_k z_j$ dan menghitung pembaruan terhadap w_{k0} dengan $\Delta w_{k0} = \alpha \delta_k$
7. Menghitung penjumlahan kesalahan unit tersembunyi $\delta_{netj} = \sum_{k=1}^m \delta_k w_{kj}$ dan mengalikan dengan turunan fungsi aktivasi $\delta_j = \delta_{netj} \psi'(Z_{netj})$ kemudian menghitung perubahan bobot untuk melakukan pembaruan nilai v_{ij} dengan $\Delta v_{ij} = \alpha \delta_j x_i$ serta perubahan bobot bias dengan $\Delta v_{0j} = \alpha \delta_j$
8. Menghitung semua perubahan bobot unit keluaran $w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj}$ dan perubahan bobot tersembunyi $v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij}$
9. Kondisi tes berhenti

Algoritma *conjugate gradient* merupakan algoritma yang menyesuaikan bobot dengan arah konjugasinya (Kusumadewi, 2004). Algoritma *conjugate gradient* dapat dideskripsikan sebagai berikut (Warsito, 2009):

1. Melakukan inisialisasi bobot awal dengan bilangan acak kecil, $epoch=0$, $error=0$, dan menetapkan $epoch$ maksimum serta ambang batas $error$
2. Menghitung $g_j = \frac{\partial E}{\partial w_j}$, $g_{b2} = \frac{\partial E}{\partial b_2}$, $g_{ij} = \frac{\partial E}{\partial v_{ij}}$, dan $g_{b1j} = \frac{\partial E}{\partial b_1}$ dan $E = \sum_k \frac{1}{2} (\text{output}_k - \text{target}_k)^2$
3. Menghitung $d_0 = -g_0$
4. Menghitung *line search* $\alpha_t = \text{argmin}_\alpha [f(w_t + \alpha_t d_t)]$
5. Mengubah vektor bobot baru dengan $w_{t+1} = w_t + \alpha_t d_t$ dan $d_t = -g_t$
6. Jika $MSE < \text{ambang batas } error$ atau $epoch = \text{maksimum } epoch$, maka kondisi penghentian terpenuhi.

Klasifikasi merupakan proses untuk menemukan sekumpulan model yang membedakan kelas data sehingga model dapat digunakan untuk memprediksi suatu nilai kelas yang belum diketahui (Warsito, 2009). Umumnya, pengukuran klasifikasi dilakukan dengan matriks konfusi (Utomo & Mesran, 2020).

Tabel 1. Matriks Konfusi

F_{ij}		Kelas Hasil Prediksi (j)	
		Kelas = 1	Kelas = 0
Kelas Asli (i)	Kelas = 1	F_{11}	F_{10}
	Kelas = 0	F_{01}	F_{00}

Berdasarkan matriks konfusi, dapat diperoleh akurasi yang merupakan jumlah total prediksi yang diperlukan untuk memastikan bahwa sistem bekerja dengan benar dan diestimasi dalam $Akurasi = \frac{F_{11}+F_{00}}{F_{11}+F_{00}+F_{01}+F_{10}}$ (Yuvaraj, et al., 2021). Tingkat keberhasilan model dalam menemukan informasi dapat dihitung dengan $Recall = \frac{F_{11}}{F_{11}+F_{10}}$ (Utomo & Mesran, 2020).

Bootstrap merupakan metode berbasis komputer untuk menentukan perkiraan statistik. Algoritma *bootstrap* dimulai dengan menghasilkan sampel *bootstrap* independen $x^{*1}, x^{*2}, \dots, x^{*B}$ yang masing-masing berukuran n (Efron & Tibshirani, 1993). Jika pengambilan sampel acak dilakukan berkali-kali tanpa batas, maka $100(1 - \alpha)\%$ titik yang dihasilkan akan mewakili wilayah kepercayaan dan $100\alpha\%$ akan mewakili daerah penolakan. Interval kepercayaan dapat dihitung dengan $\hat{\theta} \pm z^{(1-\alpha)} \hat{\sigma}$ (DiCiccio & Efron, 1996).

3. METODE PENELITIAN

Data yang digunakan adalah data sekunder yang diperoleh dari *website* Mendeley yaitu rekam medis elektronik hasil pemeriksaan laboratorium pasien dari rumah sakit swasta di Indonesia. Variabel yang digunakan terdiri dari status pasien sebagai target yang memuat keterangan apakah pasien perlu di rawat inap atau rawat jalan dan hematokrit, hemoglobin, eritrosit, leukosit, trombosit, MCH, MCHC, MCV, umur, serta jenis kelamin sebagai variabel independen. Langkah-langkah yang dilakukan untuk menganalisis data penelitian adalah:

1. Memuat data hasil pemeriksaan laboratorium
2. Melakukan normalisasi data dengan *range* [0,1;0,9]
3. Membagi data menjadi data *training* dan *testing* secara acak dengan rasio 95:5
4. Melakukan *bootstrap* pada data *training* sebanyak 100 kali
5. Mengambil sampel pada data *input* dan data target pada tiap ulangan *bootstrap*
6. Menentukan jumlah *neuron* pada *hidden layer*
7. Menentukan fungsi aktivasi
8. Menentukan bobot awal, iterasi maksimum, dan *goal*
9. Membangun arsitektur jaringan
10. Melakukan proses *training* dengan algoritma *conjugate gradient Beale-Powell Restarts*
11. Menentukan apakah nilai *error* lebih kecil dari ambang atau jumlah iterasi maksimal telah tercapai. Jika iya maka dilanjutkan dengan proses selanjutnya, sementara jika tidak maka kembali pada proses *training*
12. Menentukan bobot akhir
13. Melakukan klasifikasi terhadap data *training* dengan ketentuan jika hasil perhitungan dari jaringan $\geq 0,5$ maka akan dimasukkan ke kelas 1, dan jika $< 0,5$ maka akan dimasukkan ke kelas 0
14. Melakukan evaluasi hasil klasifikasi dengan menghitung hasil akurasi dan *recall*
15. Membandingkan hasil *bootstrap* dan diambil data dengan akurasi terbesar
16. Mengulangi langkah 4 sampai 14 untuk tiap data yang di *bootstrap*
17. Membangun jaringan baru dengan data *training* yang memiliki akurasi terbesar
18. Mengambil data *testing*
19. Melakukan klasifikasi data *testing* dengan memasukkan nilai $\geq 0,5$ kedalam kelas 1 dan $< 0,5$ kedalam kelas 0
20. Melakukan evaluasi hasil klasifikasi data *testing* dengan melihat akurasi

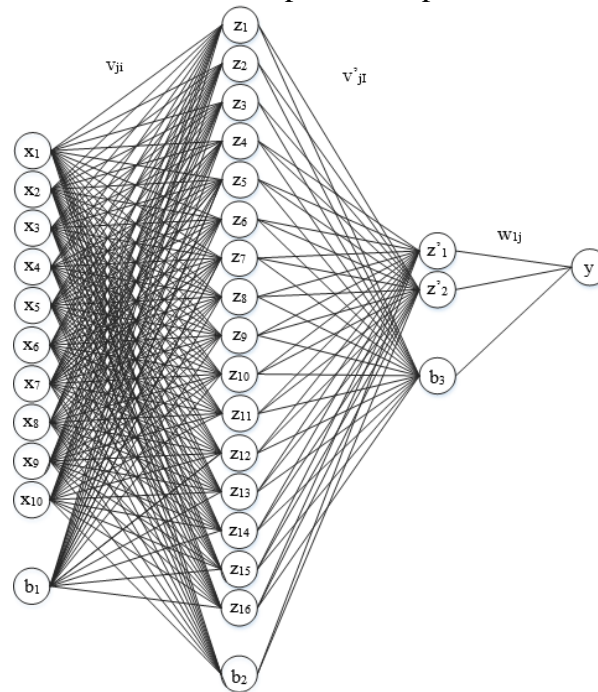
4. HASIL DAN PEMBAHASAN

Data dibagi menjadi data *training* dan *testing* dengan perbandingan 95% (4191 data) dan 5% (221 data). *Resample* data secara berulang dilakukan dengan metode *bootstrapping* sebanyak 100 kali pada data *training* sehingga didapatkan bermacam-macam hasil simulasi sampel. Penentuan jumlah *neuron* dilakukan dengan melakukan percobaan satu hingga 20 *neuron* dengan bermacam-macam kombinasi fungsi aktivasi hingga diperoleh hasil pada Tabel 2.

Tabel 2. Hasil Percobaan Kombinasi Fungsi Aktivasi

Persentase Pembagian Data	Model Arsitektur	Fungsi Aktivasi	Hasil Rata-Rata Akurasi 100 Kali <i>Bootstrap</i>
95:5	10-12-1	Purelin-Logsig	57,21%
95:5	10-8-1	Logsig-Tansig	58,81%
95:5	10-9-1	Logsig-Logsig	60,14%
95:5	10-13-12-1	Purelin-Purelin- Purelin	60,07%
95:5	10-16-2-1	Purelin-Tansig- Purelin	60,17%

Berdasarkan Tabel 2, model terbaik yang dipilih berdasarkan nilai rata-rata akurasi yang terbesar yaitu pada kombinasi 10-16-2-1 dan fungsi aktivasi purelin-tansig-purelin. Arsitektur ini menghasilkan rata-rata akurasi sebesar 60,17% dan *recall* sebesar 99,77%. Arsitektur jaringan yang terbentuk dari kombinasi ini dapat dilihat pada Gambar 1.



Gambar 1. Arsitektur Jaringan

Nilai x_1 hingga x_{10} pada gambar menunjukkan *neuron input* pada *input layer*, v_{ji} adalah bobot yang menghubungkan antara *input layer* dengan *hidden layer* pertama, nilai z_1 hingga z_{16} merupakan *hidden layer* pertama, v'_{ji} merupakan bobot yang menghubungkan antara *hidden layer* pertama ke *hidden layer* kedua, z'_1 dan z'_2 merupakan *hidden layer* kedua, b_1 , b_2 , dan b_3 adalah unit bias, dan y adalah *output layer* yang berfungsi untuk menampilkan keluaran.

Model terbaik untuk melakukan klasifikasi dibangun dengan 10 *neuron* pada *input layer*, 16 *neuron* ada *hidden layer* pertama, 2 *neuron* pada *hidden layer* kedua, dan satu *neuron* pada *output layer*. Persamaan model jaringan $y = \psi_o \left\{ v_{b3} + \sum_{i=1}^2 w_{1i} \psi_{h2} \left\{ \left(\sum_{j=1}^2 v'_{jb2} + \sum_{i=1}^{16} v'_{ji} \psi_{h1} \left(\sum_{j=1}^{16} (v_{jb1} + \sum_{i=1}^{10} v_{ji} x_i) \right) \right) \right\} \right\}$ dengan kombinasi fungsi aktivasi identitas, sigmoid biner, dan identitas. Model ini akan memprediksi dengan kriteria jika nilai y bernilai $\geq 0,5$ maka pasien dikategorikan rawat inap, dan jika bernilai $< 0,5$ maka pasien cukup memerlukan rawat jalan.

Tabel 3. Contoh Hasil Prediksi

Input										Target	Output	Prediksi	Keterangan
x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₈	x ₉	x ₁₀				
33,9	11,7	4,78	7	41	24,5	34,5	70,9	7	2	1	0,5504	1	Benar
44,3	14,9	5,06	5,8	193	29,4	33,6	87,5	23	2	0	0,4746	0	Benar
45,3	14,9	4,96	5,5	212	30	32,9	91,3	46	2	0	0,5684	1	Salah
29,3	9,4	3,41	9,2	395	27,6	32,1	85,9	39	1	1	0,5574	1	Benar
31,2	10,7	3,43	9,6	303	31,2	34,3	91	32	1	1	0,7139	1	Benar
38,6	12,8	4,34	7,1	321	29,5	33,2	88,9	39	1	0	0,5760	1	Salah
47,1	15,8	5,39	9,4	335	29,3	33,5	87,4	68	2	0	0,4766	0	Benar
38,7	13,6	4,5	6,1	245	30,2	35,1	86	48	2	0	0,5492	1	Salah
39,9	12,7	5	6,4	331	25,4	31,8	79,8	58	1	0	0,4781	0	Benar
39,9	12,7	5	6,4	331	25,4	31,8	79,8	58	1	0	0,4779	0	Benar

Kinerja model klasifikasi diuji dengan melihat akurasi ketika data *testing* digunakan pada model. Perintah untuk menampilkan matriks konfusi pada MATLAB adalah `matriks_konfusi_testing = (confusionmat (tgt_ts', hasil_testing'))`. Tabel 4 merupakan contoh matriks konfusi dari hasil klasifikasi dengan menggunakan data *testing*.

Tabel 4. Matriks Konfusi Hasil Klasifikasi

Observasi	Prediksi	
	Rawat Inap (1)	Rawat Jalan (0)
Rawat Inap (1)	0	68
Rawat Jalan (0)	0	153

Berdasarkan Tabel 3, diketahui bahwa ketika model diterapkan dalam data *testing*, dihasilkan matriks konfusi yang memprediksi hasil rawat inap ke rawat jalan (*false negative*) sebesar 68 data, dan data rawat jalan ke rawat jalan (*true negative*) sebesar 153 data, dan selebihnya 0. Nilai akurasi yang diperoleh sebesar $akurasi = \frac{0+153}{0+68+0+153} = \frac{153}{221} = 0,6923$ atau 69,23%.

5. KESIMPULAN

Berdasarkan analisis dan pembahasan, didapatkan kesimpulan bahwa arsitektur model terbaik yang didapatkan untuk menetapkan tindak lanjut terhadap hasil tes laboratorium (pasien dirawat inap atau rawat jalan) adalah 10 *neuron input*, 16 *neuron hidden layer* pertama, dua *neuron hidden layer* kedua, dan satu *neuron output*. Fungsi aktivasi yang digunakan adalah fungsi identitas pada *hidden layer* pertama dan *output* serta sigmoid biner pada *hidden layer* kedua. Rata-rata akurasi dengan 100 kali percobaan *bootstrap* adalah 60,17% dengan *recall*

99,77%. Sementara pada data *testing*, model jaringan syaraf tiruan *backpropagation* dengan algoritma *conjugate gradient Beale-Powell Restarts* mampu menghasilkan akurasi untuk klasifikasi tindak lanjut terhadap hasil tes laboratorium sebesar 69,23%. Percobaan yang dilakukan dengan beberapa kombinasi model arsitektur jaringan menghasilkan bahwa fungsi aktivasi serta pembagian data *training* dan *testing* cukup berpengaruh pada nilai akurasi.

DAFTAR PUSTAKA

- DiCiccio, T. J. & Efron, B., 1996. Bootstrap Confidence Intervals. *Statistical Science*, 11(3), pp. 189-228. <https://www.jstor.org/stable/2246110>
- Efron, B. & Tibshirani, R. J., 1993. *An Introduction to The Bootstrap*. Berlin: Springer Science+Business Media Dordrecht.
- Fausett, L., 1994. *Fundamentals of Neural Networks: Architectures, Algorithms, and Application*. New Jersey: Prentice-Hall.
- Kusumadewi, S., 2004. *Membangun Jaringan Syaraf Tiruan: Menggunakan Matlab dan Excel Link*. 1 penyunt. Yogyakarta: Graha Ilmu.
- Rosida, A. & Hendriyono, F., 2015. Nilai Rujukan Hematologi Orang Dewasa Normal di RSUD Ulin Banjarmasin. *Berkala Kedokteran*, 11(1), Hal. 101-109. <http://dx.doi.org/10.20527/jbk.v11i1.190>
- Utomo, D. P. & Mesran, 2020. Analisis Komparasi Metode Klasifikasi Data Mining dan Reduksi Atribut pada Dataset Penyakit Jantung. *Media Informatika Budidarma*, 4(2), Hal. 437-444. <http://dx.doi.org/10.30865/mib.v4i2.2080>
- Wanto, A., 2017. Optimasi Prediksi Dengan Algoritma Backpropagation Dan Conjugate Gradient Beale-Powell Restarts. *Jurnal Nasional Teknologi dan Sistem Informasi*, 3(3), Hal. 370-380. <https://doi.org/10.25077/TEKNOSI.v3i3.2017.370-380>
- Warsito, B., 2009. *Kapita Selekta Neural Network*. Semarang: BP Universitas Diponegoro.
- Wisesty, U. N., Adiwijaya & B.W, T. A., 2010. Algoritma Conjugate Gradient Polak Ribiere untuk Peningkatan Performansi Backpropagation Pada Sistem Prediksi Temperatur Udara. *Penelitian dan Pengembangan Telekomunikasi*, 15(2). http://digilib.mercubuana.ac.id/manager/t!@file_artikel_abstrak/Isi_Artikel_600618627378.pdf
- Yuvaraj, N. et al., 2021. Nature-Inspired-Based Approach for Automated Cyberbullying Classification on Multimedia Social Networking. *Hindawi Mathematical Problems in Engineering*, Volume 2021. <https://doi.org/10.1155/2021/6644652>