

PREDIKSI MENGGUNAKAN BACKPROPAGATION NEURAL NETWORK DENGAN METODE OPTIMASI GENETIC ALGORITHM

Avilian Anniza^{1)*}, Putranto Hadi Utomo²⁾, Titin Sri Martini³⁾

Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Sebelas Maret

email: ¹⁾aviliananniza@student.uns.ac.id, ²⁾putranto@staff.uns.ac.id, ³⁾titinsmartini@staff.uns.ac.id

Abstrak

Model data time series non linear merupakan model data yang cocok untuk diprediksi menggunakan neural network (NN). Prediksi merupakan upaya untuk memperkirakan sesuatu yang paling mungkin terjadi di masa mendatang berdasarkan informasi masa lalu dan masa sekarang secara matematis. Dalam NN terdapat banyak metode dan algoritme salah satunya backpropagation. Backpropagation memiliki kemampuan untuk melatih jaringan agar mendapatkan keseimbangan antara pengenalan pola yang digunakan pada saat training dan pemberian respon terhadap pola input yang serupa dengan pola training. Oleh karena itu backpropagation tepat untuk menyelesaikan masalah prediksi. Namun, backpropagation memiliki kelemahan yaitu pemilihan bobot awal secara random sehingga kerap terjebak pada lokal minimum. Solusi yang tepat untuk mengatasi masalah tersebut adalah dengan dilakukan optimasi terhadap algoritme backpropagation. Terdapat berbagai macam metode optimasi, salah satunya adalah genetic algorithm (GA). GA merupakan algoritme heuristik yang menggunakan pendekatan evolusi biologis. GA dapat digunakan untuk menentukan kombinasi arsitektur dan bobot awal terhadap parameter backpropagation. Pengkajian model backpropagation dengan metode optimasi GA dan estimasi parameternya menjadi tujuan dari penelitian ini. Hasil pengkajian penelitian ini menunjukkan bahwa GA dapat memberikan tingkat akurasi yang lebih baik.

Kata Kunci: *Backpropagation, Genetic Algorithm, Lokal Minimum, Neural Network, Prediksi*

PENDAHULUAN

Data *time series* merupakan data amatan yang diambil dalam kurun waktu tertentu. Analisis terhadap data *times series* memungkinkan untuk melakukan prediksi ataupun peramalan. Prediksi merupakan proses memperkirakan secara sistematis tentang suatu hal yang paling memungkinkan di masa mendatang berdasarkan informasi masa lalu dan masa sekarang. Prediksi tidak harus memberikan jawaban pasti terhadap peristiwa mendatang, melainkan berusaha mencari jawaban sedekat mungkin dengan peristiwa yang akan terjadi (Herdianto, 2013).

Jaringan saraf tiruan atau *neural network* (NN) merupakan salah satu cabang ilmu *artificial intelligence* (AI). NN merupakan tipe *machine learning*

yang meniru cara kerja otak manusia (Willmot, 2019). Terdapat empat tipe *machine learning* yaitu *supervised learning*, *semi-supervised learning*, *unsupervised learning*, dan *reinforcement learning*. NN dapat digunakan untuk memprediksi peristiwa masa mendatang berdasarkan pola peristiwa masa lalu dan masa sekarang.

NN memiliki banyak metode dan algoritme. Salah satunya adalah algoritme *backpropagation*. Menurut (Siang, 2005), *backpropagation* merupakan algoritme yang memiliki kemampuan untuk melatih jaringan agar mendapatkan keseimbangan antara pengenalan pola yang digunakan saat pelatihan dan pemberian respon yang benar terhadap pola masukan yang serupa dengan pola penelitian. Oleh karena itu, *backpropagation* tepat untuk

menyelesaikan masalah prediksi. Namun, *backpropagation* memiliki kelemahan yaitu pemilihan bobot awal secara *random* sehingga berdampak pada terjebaknya *NN* pada lokal minimum.

Terdapat berbagai macam metode optimasi untuk menyelesaikan masalah tersebut, salah satunya ialah *genetic algorithm (GA)*. *GA* dapat digunakan untuk menentukan kombinasi arsitektur dan bobot serta bias awal terhadap parameter *backpropagation NN* (Suhendra & Wardoyo, 2015). Semakin optimal bobot yang ditentukan maka semakin kecil eror dari model tersebut.

AI merupakan ilmu komputer yang memungkinkan mesin bekerja seperti otak manusia. Pada tahun 1940 ditemukannya *Atanasoff Berry Computer* menjadi awal berkembangnya *AI*. *AI* memiliki banyak bidang salah satunya adalah *NN*. Bermula pada tahun 1943 Waffan McCulloch dan Walter Pitts memodelkan secara matematis cara kerja sel-sel otak (Suhari, 2010). Kemudian, pada tahun 1949 Donald Hebb merancang skema pembelajaran untuk memperbaiki koneksi-koneksi antar *neuron*. Pada tahun 1958, konsep dasar *perceptron* untuk klasifikasi pola dikembangkan oleh Rosenblatt (Jaya et al., 2018).

Algoritme *backpropagation* pertama kali dikemukakan pada tahun 1974 oleh Paul Werbos (Rumelhart & Chauvin, 1995). *Backpropagation* memiliki kelemahan yaitu sering terjebak dalam lokal minimum yang mengakibatkan kecepatan *convergence* buruk dan tidak stabil. Kelemahan tersebut dikarenakan pemilihan bobot awal secara *random* (Suhendra & Wardoyo, 2015). Pada tahun 1975, John Holland mengembangkan metode optimasi *GA*. Pada tahun 2015 Haviluddin dan Alfred melakukan penelitian tentang prediksi data *time series* menggunakan *GA* dan *backpropagation* (Haviluddin &

Alfred, 2016). Pada tahun 2018, Setiawan dkk. menerapkan *backpropagation* dengan metode optimasi *GA* untuk memprediksi nilai tukar rupiah terhadap dolar Amerika (Setiawan et al., 2018).

METODE PENELITIAN

Penelitian ini merupakan penelitian berdasarkan teori. Peneliti ini melakukan studi tentang prediksi data *time series* dengan metode optimasi *GA* pada *backpropagation NN*. Metode penelitian yang digunakan dalam penelitian ini adalah *literature review* dari buku referensi, jurnal, dan tulisan tentang *AI* khususnya pada *backpropagation NN* dan *GA*.

Langkah-langkah yang dilakukan untuk melakukan penelitian prediksi data *time series series* dengan metode optimasi *GA* pada *backpropagation NN* adalah

1. Mempelajari materi terkait *backpropagation NN* dan *GA*,
2. Menganalisis hubungan antara *backpropagation NN* dan *GA*, dan
3. Menyatakan kesimpulan.

HASIL DAN PEMBAHASAN

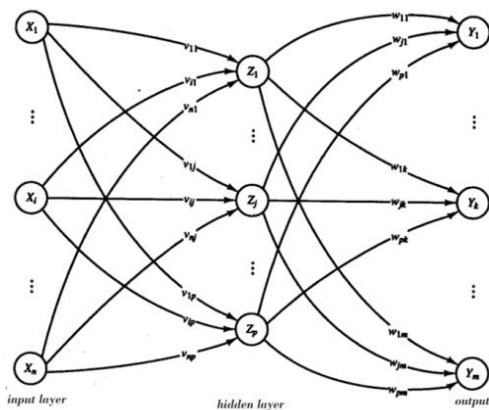
Hasil Penelitian

Neural Network

Fausett menuliskan, *NN* merupakan sistem komputasi yang operasinya terdiri dari sistem pemrosesan informasi (Fausett, 1994). *NN* memiliki cara kerja yang sama seperti jaringan saraf biologis manusia. Jaringan saraf biologis terdiri atas empat komponen yaitu dendrit, badan sel (*soma*), akson, dan sinapsis. Pada jaringan saraf biologis dendrit berfungsi untuk menerima sinyal. Selanjutnya, sinyal diteruskan menuju badan sel. Setelah selesai diproses dalam badan sel, sinyal akan diteruskan ke sinapsis melalui akson.

Struktur jaringan pada *NN* terdiri atas *input layer*, *hidden layer*, *output*

layer, dan bobot. Pada *NN input layer* bertugas menerima dan mengirimkan data yang selanjutnya diteruskan ke *hidden layer*. *Hidden layer* dapat terdiri dari beberapa *nodes* yang berfungsi untuk memproses data. Setelah mendapatkan hasil proses data dari *hidden layer*, data akan dikeluarkan dalam bentuk *output*. Sementara itu, fungsi bobot dalam *NN* digunakan untuk belajar dan mengingat suatu informasi (Fausett, 1994). Struktur *NN* ditunjukkan pada Gambar 1.

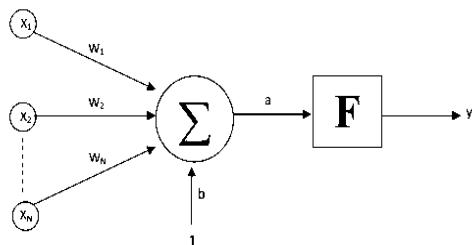


Gambar 1. Neural network

Dalam *NN* terdapat fungsi aktivasi yang berfungsi untuk mengaktifkan *neuron* sehingga *neuron* mengolah *N* input $(x_1, x_2, x_3, \dots, x_n)$ yang masing-masing memiliki bobot $(w_1, w_2, w_3, \dots, w_n)$ serta bobot bias b , dengan rumus

$$a = \sum_{i=1}^n x_i w_i$$

Setelah itu fungsi aktivasi F akan mengolah a menjadi *output* y , seperti yang ditunjukkan pada Gambar 2.



Gambar 2. Struktur fungsi aktivasi dalam *NN*

Backpropagation Neural Network

Backpropagation merupakan salah satu algoritme dalam *NN* dengan pelatihan mencari bobot pada setiap *neuron* untuk menghasilkan nilai eror terkecil. *Backpropagation* sering digunakan dalam penyelesaian masalah prediksi karena merupakan metode *supervise learning* (Sutikno et al., 2016). Artinya *backpropagation* merupakan *machine learning* yang mempelajari data berlabel dimana evaluasi model dilakukan berdasarkan target.

Tahap pelatihan *backpropagation* dibagi menjadi tiga bagian yaitu, *feedforward*, *backpropagation*, dan penyesuaian bobot. Tahapan-tahapan tersebut diulang hingga memenuhi iterasi dan toleransi kesalahan. Pelatihan *backpropagation* dapat diuraikan sebagai berikut (Sutikno et al., 2016).

- Langkah 0: Inisialisasi bobot dengan bilangan acak.
- Langkah 1: Cek kondisi penghentian, jika belum terpenuhi, maka lakukan langkah 2-9
- Langkah 2: Setiap pasangan data pelatihan, lakukan langkah 3-8.
- Langkah 3: Setiap *input* $(x_i, i = 1, 2, 3, \dots, n)$ menerima sinyal x_i kemudian meneruskannya ke *unit hidden layer*.

Langkah 4: Setiap *unit hidden layer* ($z_j, j = 1, 2, 3, \dots, p$)

menjumlahkan sinyal-sinyal *input* terbobot dengan rumus

$$z_{in_j} = v_{0j} + \sum_{i=1}^n x_i v_{ij}.$$

Dengan v_{0j} merupakan bias antara *input layer* dan *hidden layer*, serta v_{ij}

merupakan bobot antara *input layer* dan *hidden layer*.

Menghitung nilai *hidden layer* menggunakan fungsi aktivasi dengan rumus

$$z_j = f(z_{in_j}).$$

Selanjutnya z_j dikirim ke *output*.

Langkah 5: Setiap *unit output* ($y_k, k = 1, 2, 3, \dots, m$)

menjumlahkan sinyal-sinyal *input* terbobot dengan rumus

$$y_{in_k} = w_{0k} + \sum_{j=1}^m z_j w_{jk}.$$

Dengan w_{0j} merupakan bias antara *hidden layer* dan *output*, serta w_{jk} merupakan bobot antara *hidden layer* dan *output*. Menghitung nilai *output* menggunakan fungsi aktivasi dengan rumus

$$y_k = f(y_{in_k}).$$

Selanjutnya mengirim sinyal tersebut ke *unit output*.

Langkah 6: Setiap *unit output* ($Y_k, k = 1, 2, 3, \dots, m$)

menerima target pola yang berhubungan dengan pola *input* pelatihan kemudian, menghitung eror pada *output* yang dinotasikan δ_k , dengan rumus

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

Dengan t_k merupakan nilai data aktual dan y_k merupakan nilai data prediksi. Selanjutnya

menghitung koreksi bobot untuk memperbaiki nilai w_{jk}

dengan rumus

$$\Delta w_{jk} = \alpha \delta_k z_j.$$

α merupakan *learning rate*.

Kemudian, dihitung juga koreksi bias untuk memperbaiki nilai w_{0k} dengan rumus

$$\Delta w_{0k} = \alpha \delta_k.$$

Mengirim δ_k ke *unit* yang ada di *layer* di bawahnya (*hidden layer*).

Langkah 7: Setiap *unit hidden layer* ($z_j, j = 1, 2, 3, \dots, p$)

menjumlahkan delta *input*-nya dari *unit* yang berbeda pada *layer* di atasnya (*output*) dengan rumus

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk}.$$

Kemudian mengalikan δ_{in_j} dengan turunan dari fungsi aktivasinya untuk menghitung eror, didefinisikan sebagai

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

Selanjutnya menghitung koreksi bobot untuk memperbaiki nilai v_{ij} dengan rumus

$$\Delta v_{ij} = \alpha \delta_j x_i$$

Kemudian menghitung koreksi bias untuk memperbaiki nilai v_{0j} , dengan rumus

$$\Delta v_{0j} = \alpha \delta_j.$$

Langkah 8: Setiap *unit output* ($y_k, k = 1, 2, 3, \dots, m$)

memprediksi bias dan bobotnya

($j = 1, 2, 3, \dots, p$) dengan

$$w_{jk}^{baru} = w_{jk}^{lama} + \Delta w_{jk}.$$

Setiap *unit hidden layer* ($z_j, j = 1, 2, 3, \dots, p$)

memperbaiki bias dan bobotnya ($i = 1, 2, 3, \dots, n$) dengan

$$v_{ij}^{baru} = v_{ij}^{lama} + \Delta v_{ij}$$

Langkah Tes kondisi berhenti, jika 9: terpenuhi maka selesai apabila tidak maka kembali ke langkah 2.

Genetic Algorithm

GA merupakan algoritme heuristik yang berdasarkan pada mekanisme evolusi biologis (Chen et al., 2019; Kusumadewi & Purnomo, 2005). Algoritme ini menggunakan pendekatan evolusi Charles Darwin dalam bidang biologi seperti pewarisan sifat, mutasi, dan kombinasi. Algoritme ini merupakan solusi dari penyelesaian optimasi permasalahan kompleks yang sulit untuk diselesaikan oleh metode konvensional (Setiawan et al., 2018). Chen juga menuliskan bahwa, GA tidak memiliki persyaratan ataupun batasan khusus sehingga GA mudah digabungkan dengan algoritme lainnya (Chen et al., 2019). Tahapan-tahapan dalam GA dapat diuraikan sebagai berikut (Setiawan et al., 2018).

1. Representasi Kromosom.
Pada tahap ini kromosom akan direpresentasikan nilai bobot pada *backpropagation* dengan pengodean *real*
2. Inisialisasi Populasi.
Pada tahap ini dibentuk individu secara acak sesuai ukuran populasi yang telah ditetapkan
3. Reproduksi.
 - a. *Crossover*.
Crossover merupakan operator rekombinasi untuk mendapatkan individu. *Crossover* dilakukan untuk menentukan nilai rasio *offspring*.
 - b. Mutasi.
Mutasi merupakan operator yang berfungsi mengganti gen yang hilang akibat proses seleksi. Mutasi dilakukan untuk menentukan nilai rasio *offspring*.
4. Evaluasi.

Dalam tahap evaluasi nilai *fitness* kromosom akan diperiksa dengan melihat nilai *fitness* ($f(x)$).

5. Seleksi.
Seleksi merupakan tahap pemilihan calon induk. Seleksi dilakukan dengan menggunakan *Roulette wheel*.

Mean Squared Error (MSE)

Mean squared error (MSE) atau rata-rata kuadrat eror merupakan salah satu cara untuk mengukur eror prediksi keseluruhan. *MSE* adalah rata-rata selisih kuadrat antara nilai data aktual dan data prediksi. Secara matematis, *MSE* dapat dirumuskan sebagai

$$E = \sum_{k=1}^n \frac{(t_k - y_k)^2}{n}$$

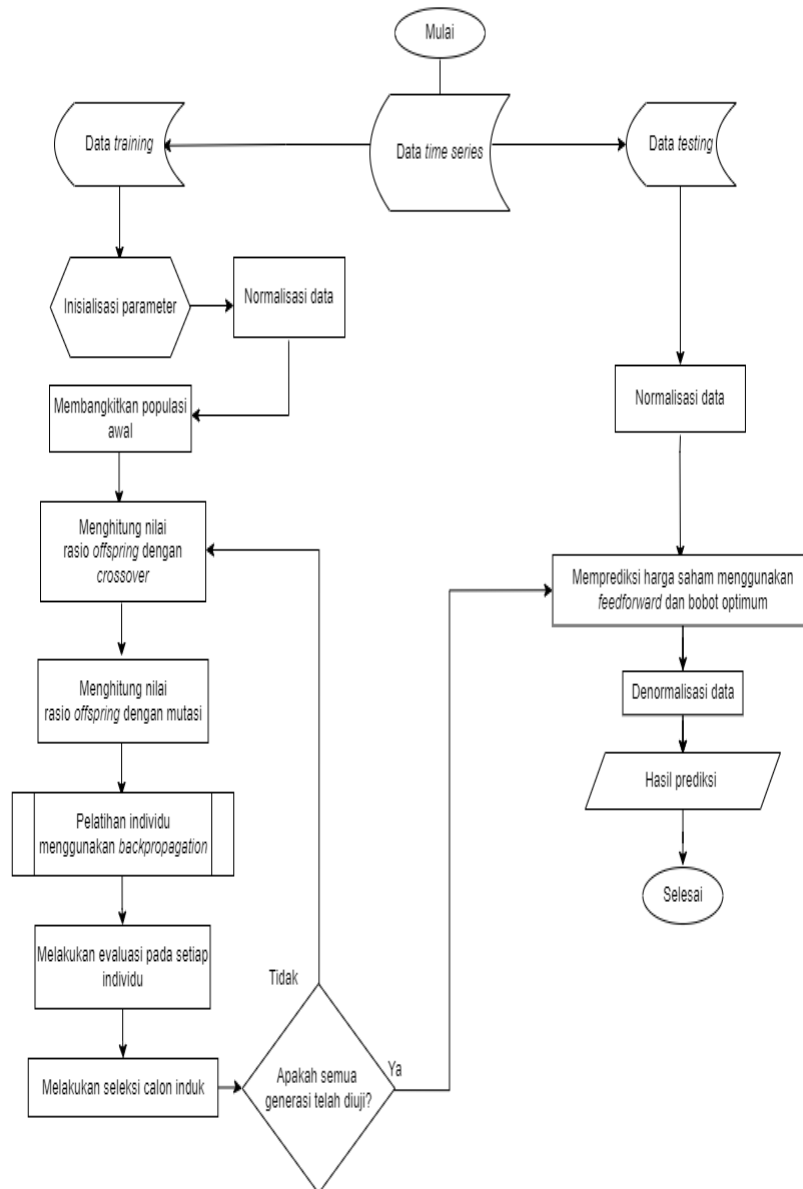
dengan E merupakan *MSE*, t_k merupakan data aktual, y_k merupakan data prediksi, dan n merupakan jumlah data. Semakin rendah nilai *MSE*, semakin mendekati perkiraan aktualnya.

Penerapan GA pada Backpropagation NN

Sebagai solusi dari kelemahan *backpropagation* dilakukan penerapan metode optimasi GA. Langkah-langkah yang dilakukan untuk mendapatkan bobot awal yang optimum adalah sebagai berikut (**setiawan**)

1. Melakukan inisialisasi populasi.
2. Menghitung rasio *offspring* dengan *crossover* dan mutasi.
3. Melakukan pelatihan dengan menggunakan *backpropagation NN* pada seluruh individu.
4. Menghitung nilai *fitness* setiap individu.
5. Melakukan seleksi.

Berikut merupakan *flowchart* penelitian penerapan GA pada *backpropagation NN*, ditunjukkan pada Gambar 3.



Gambar 3. Flowchart Penelitian

KESIMPULAN

Berdasarkan permasalahan dalam pengolahan data dengan akurasi rendah yang disebabkan oleh terjebaknya *backpropagation NN* dalam lokal minimum. Tujuan penelitian ini adalah memberikan metode optimasi *GA*. *GA* dapat menentukan nilai bobot awal

optimum sehingga *backpropagation NN* tidak terjebak di lokal minimum. Kemudian berdasarkan pembahasan dapat disimpulkan bahwa penerapan metode optimasi *GA* pada *backpropagation NN* akurasi yang lebih baik.

REFERENSI

- Chen, N., Xiong, C., Du, W., Wang, C., Lin, X., & Chen, Z. (2019). An improved genetic algorithm coupling a back-propagation neural network model (IGA-BPNN) for water-level predictions. *Water (Switzerland)*, 11(9). <https://doi.org/10.3390/w11091795>
- Fausett, L. (1994). *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*. Prentice Hall.
- Haviluddin, & Alfred, R. (2016). A genetic-based backpropagation neural network for forecasting in time-series data. *Proceedings - 2015 International Conference on Science in Information Technology: Big Data Spectrum for Future Information Economy, ICSITech 2015*, 158–163. <https://doi.org/10.1109/ICSITech.2015.7407796>
- Herdianto, (Universitas Sumatra Utara). (2013). *Prediksi Kerusakan Motor Induksi Menggunakan Metode Jaringan Saraf Tiruan Backpropagation*.
- Jaya, H., Sabran, D., Pd, M., Ma, M., Djawad, Y. A., Sc, M., Ilham, A., Ahmar, A. S., Si, S., & Sc, M. (2018). Kecerdasan Buatan. In *Journal of Chemical Information and Modeling* (Vol. 53, Issue 9).
- Kusumadewi, S., & Purnomo, H. (2005). *Penyelesaian masalah optimasi dengan teknik-teknik heuristik*. Graha Ilmu.
- Rumelhart, D. E., & Chauvin, Y. (1995). *Backpropagation: Theory, Architectures, and Applications (illustrate)*. Psychology Press.
- Setiawan, D. N., Dewi, C., & Adinugroho, S. (2018). Prediksi Nilai Tukar Rupiah Terhadap Dolar Amerika Dengan Menggunakan Algoritme Genetika-Backpropagation. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2(11), 4794–4801.
- Siang, J. J. (2005). *Jaringan Saraf Tiruan dan Pemrogramannya Menggunakan Matlab*. ANDI.
- Suhari, Y. (2010). Jaringan Syaraf Tiruan : Aplikasi Pemilihan Merek. *Jurnal Teknologi Informasi DINAMIK*, XV(2), 90–95.
- Suhendra, C. D., & Wardoyo, R. (2015). Penentuan Arsitektur Jaringan Syaraf Tiruan Backpropagation (Bobot Awal dan Bias Awal) Menggunakan Algoritma Genetika. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 9(1), 77. <https://doi.org/10.22146/ijccs.6642>
- Sutikno, Indriyati, Priyo, S. S., Helmie, A. W., Indra, W., Nurdin, B., & K, T. W. (2016). *Backpropagation dan Aplikasinya. Ilmu Komputer: Studi Kasus Dan Aplikasinya*, 134–146.
- Willmot, P. (2019). *Mavhine Learning An Applied Mathematics Introduction*. Panda Ohana