

Implementasi Algoritme Genetika untuk Mengoptimasi Topologi Jaringan Saraf Tiruan

Muhamad Saad Nurul Ishlah

Diploma Sistem Informasi, Universitas Pakuan
Jl. Pakuan PO Box 452 Bogor 16143, +622518375547
e-mail: nurul.ishlah@unpak.ac.id

Abstrak

Tulisan ini menyajikan hasil pengimplementasian dari Genetic Algorithm (GA, Algoritme Genetika) untuk mengoptimasi topologi pada rancangan Neural Network (NN, Jaringan Saraf Tiruan). Pada penelitian ini, GA digunakan untuk menemukan jumlah neuron pada lapisan tersembunyi (*hidden layer*) untuk menyelesaikan masalah klasifikasi. Selain itu, GA juga digunakan untuk mengoptimasi *learning rate* dan parameter momentum dari rancangan NN. Untuk mengevaluasi performa dari hasil topologi baru yang terbentuk, dilakukan perbandingan dengan topologi NN yang belum dioptimasi menggunakan GA. Hasil simulasi menunjukkan bahwa arsitektur NN yang baru menghasilkan klasifikasi yang lebih baik.

Kata kunci: Optimasi, Topologi Jaringan Saraf Tiruan, Algoritme Genetika, Parametric Encoding

1. Pendahuluan

Selama bertahun-tahun, *artificial neural networks* (ANNs) telah diimplementasikan untuk membantu menyelesaikan berbagai macam masalah, seperti: *data clustering*, *pattern recognition*, *data fitting*, dan *classification* [1, 2]. Terinspirasi oleh mekanisme kerja otak manusia dalam berpikir, ANNs mengumpulkan informasi melalui proses belajar untuk selanjutnya dipropagasi kembali melalui hubungan jaringan internal [2, 3]. Walaupun demikian, terlepas dari kemampuannya untuk menyelesaikan berbagai masalah kecerdasan buatan, perancangan NN dalam kondisi tertentu sering dilakukan dengan proses *test and error* [4]. Oleh karenanya, unsur keoptimalan rancangan NN dalam kondisi tersebut tidak dapat digaransikan.

Di lain pihak, GA telah dikenal sebagai sebuah teknik optimasi yang efektif untuk keperluan optimasi beban (*weight optimization*) dan pemilihan topologi dari sebuah rancangan NN [3]. Dalam GA, karakteristik dari sebuah individu diwakili oleh sebuah genotip (*genotype*). Selanjutnya, untuk keperluan pemodelan keberlangsungan hidup dari *best fitness*, terdapat fitur dalam GA yang dapat digunakan, yaitu proses seleksi. Selain itu, GA juga digunakan untuk memodelkan reproduksi melalui fitur operasi *crossover* yang dimilikinya [5]. Prosesnya akan diulang secara iteratif sampai kondisi pemberhentian tertentu terpenuhi [5].

Dalam domain NN, GA dapat diimplementasikan sebagai konsep dasar untuk mengubah secara perlahan arsitektur jaringan sampai ditemukan arsitektur jaringan yang cocok untuk masalah tertentu [3]. Ide dasarnya adalah dengan melakukan pencarian secara genetika dalam sebuah populasi arsitektur jaringan [3].

Tulisan ini menyajikan hasil pengimplementasian GA untuk menemukan topologi yang optimal sebagai rancangan NN. Lebih khusus, penelitian ini difokuskan pada implementasi NN untuk menyelesaikan masalah klasifikasi. Topologi NN yang dimaksud pada tulisan ini merujuk pada *fixed multilayer feed-forward network* yang memiliki tiga lapisan: sebuah lapisan input, sebuah *hidden layer*, dan lapisan keluaran. Fokus dari penelitian ini ditujukan untuk menemukan jumlah neuron yang optimal pada *hidden layer*. Lebih lanjut, karena teknik *gradient descent* digunakan untuk melatih jaringan, pada penelitian ini juga dilakukan proses optimasi pada *initial learning rate* dan nilai momentum.

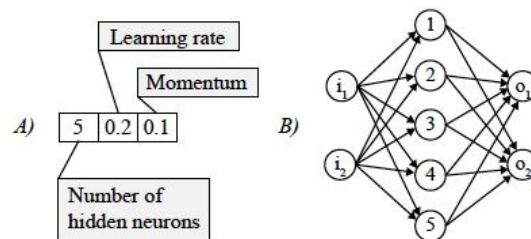
Selama proses penemuan dan pengaplikasian topologi yang telah dioptimasi, data yang digunakan untuk klasifikasi adalah kumpulan data tumbuhan Iris dari *UCI Machine Learning Repository* [6]. Dataset ini terdiri dari tiga tipe bunga Iris yang dapat diklasifikasikan ke dalam empat fitur, yaitu: *sepal length*, *sepal width*, *petal length*, dan *petal width* [1, 6]. Hasil dari simulasi menunjukkan bahwa topologi yang telah dioptimasi menggunakan GA, dapat meningkatkan akurasi pengenalan tumbuhan iris jika dibandingkan dengan topologi yang dirancang menggunakan pendekatan heuristik.

Penyusunan tulisan ini dilakukan sebagai berikut: Sesi 2 membahas tentang metode pengimplementasian GA ke dalam proses perancangan NN. Sesi 3 menyajikan konfigurasi yang digunakan dalam simulasi. Selanjutnya dalam Sesi 4, dibahas mengenai hasil dan analisis hasil simulasi. Terakhir, kesimpulan dari penelitian yang dikerjakan dihimpun dalam Sesi 5.

2. Metode Penelitian

Pada bagian sesi kedua ini, dijelaskan bagaimana GA dan NN diimplementasikan untuk menemukan topologi rancangan jaringan NN sebagai metode untuk menyelesaikan masalah klasifikasi pada dataset Iris. Untuk mengukur performa dari topologi yang terbentuk, digunakan metode pengukuran *mean square error* (MSE).

Hal pertama yang dilakukan pada penelitian ini adalah membangkitkan rancangan-rancangan jaringan sebagai populasi kandidat solusi. Populasi ini disebut juga sebagai populasi kromosom awal pada algoritme genetika. Penulis menggunakan teknik *parametric encoding* untuk merepresentasikan jaringan-jaringan tersebut sebagai kromosom-kromosom dengan tiga gen yang selanjutnya akan dilakukan optimasi terhadap kromosom-kromosom tersebut. Teknik ini biasanya digunakan ketika algoritme pembelajaran *back-propagation* digunakan [7]. Contoh dari pengimplementasian teknik ini dapat dilihat pada Gambar 1.



Gambar 1. Sebuah contoh pengkodean *feed-forward neural network* sederhana dengan satu *hidden layer* dan algoritme pembelajaran *back-propagation*. A) Sebuah kromosom yang berisikan jumlah neuron di dalam *hidden layer* dan parameter-parameter dari algoritme pembelajaran. B). Jaringan akhir yang mempunyai lima neuron dalam *hidden layer* [7].

Setelah populasi awal terbentuk, selanjutnya NN akan dijalankan untuk mendapatkan nilai *fitness* dari masing-masing kromosom dengan nilai *epoch* tertentu dan perhitungan MSE. Untuk melakukannya, setiap individu kromosom tersebut harus terlebih dahulu diuraikan (*decode*). Proses ini akan terus dilakukan hingga semua individu kromosom dalam populasi dijalankan dan mendapatkan nilai *fitness*.

Selanjutnya, setelah nilai dari semua individu didapatkan, proses yang dilakukan selanjutnya adalah memilih dua orang tua (*parents*) untuk dilakukan persilangan dan dijalankanlah proses operasi genetika untuk menghasilkan jaringan baru sebagai keturunan (*offspring*). Dalam penelitian ini digunakan strategi persilangan (*crossover*) dalam proses mereproduksi keturunan tersebut. Karena digunakan metode penyandian sederhana dimana terdapat *floating values*, strategi penyilangan ini direpresentasikan dalam persamaan matematika berikut:

$$\alpha \cdot x_1 + (1 - \alpha) \cdot x_2 \quad (1)$$

dimana x_1 merupakan orang tua pertama dan x_2 adalah orang tua kedua dengan nilai $\alpha = 5$ [8].

Jika jaringan baru yang terbentuk lebih baik dari orang tuanya, maka jaringan tersebut akan ditempatkan di dalam populasi baru. Langkah ini akan terus dilakukan sampai terbentuk populasi baru sebagai generasi baru. Semua proses ini akan diulang hingga sejumlah generasi tertentu tercapai.

Tabel 1. Parameter-parameter optimasi

	Parameter	Nilai
<i>Genetic Algorithm</i>	Ukuran populasi	20
	Jumlah awal rentang neuron	rand[1,10]
	Rentang awal <i>learning rate</i>	rand[0,1]
	Rentang awal momentum	rand[0,1]
	Pembangkitan	100
<i>Neural Network</i>	<i>Training function</i>	traingdm
	Jumlah maksimum epochs	3000

3. Simulasi

Untuk mensimulasikan pengimplementasian GA dan NN ini, digunakan Matlab Global Optimization *toolbox* dan Matlab Neural Network *toolbox*. Dalam simulasi ini, digunakan dataset pengenalan pola Iris [6] dari UCI Machine Learning Repository. Tabel 1 menunjukkan parameter-parameter yang digunakan untuk menjalankan simulasi.

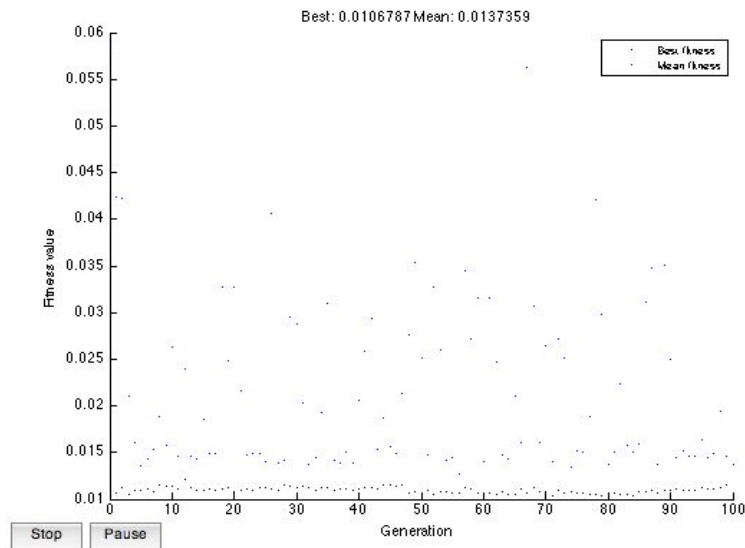
Proses evaluasi pengintegrasian NN dengan GA dilakukan untuk mengetahui seberapa baik pengaruh dari GA ini terhadap pemilihan topologi NN untuk masalah klasifikasi. Untuk keperluan tersebut, topologi baru yang dihasilkan dibandingkan dengan topologi sebelumnya yang menggunakan pendekatan heuristik pada penelitian sebelumnya di [9]. Topologi heuristik ini dijalankan dengan parameter-parameter yang tertera pada Tabel 2. Kehandalan dari kedua topologi yang dibandingkan di atas, diukur dengan menghitung rata-rata jumlah epoch, waktu CPU/epoch, dan tingkat akurasi.

Tabel 2. Parameter-parameter untuk topologi heuristik

Parameter	Nilai
<i>Initial weights</i>	rand[0,1]
<i>Learning rate</i>	0.3
<i>Momentum value</i>	0.7
<i>Target error</i>	0.05

4. Hasil dan Diskusi

Gambar 2 menunjukkan hasil dari proses optimasi arsitektur jaringan saraf tiruan yang dilatih menggunakan fungsi pembelajaran *traingdm* dalam 100 generasi. Dari gambar tersebut, dapat terlihat bahwa nilai *fitness* akhir yang didapatkan adalah 0.0106 dan rata-rata dicapai pada 0.0137. Berdasarkan hasil simulasi tersebut, GA menunjukkan bahwa kromosom individu terbaik didapat dengan kombinasi nilai genetik 6, 0.4701, and 0.5878. Nilai-nilai tersebut merepresentasikan jumlah neuron, learning rate, dan momentum secara berurutan.



Gambar. 2. Grafik hasil optimasi dari topologi jaringan dengan menggunakan fungsi *traingdm*.

Hasil perbandingan antara topologi baru yang didapat dari menjalankan GA dengan topologi jaringan saraf tiruan dengan rancangan heuristik ditunjukkan pada Tabel 3. Baris pertama merepresentasikan performa dari topologi baru yang telah dioptimasi. Hasilnya menyatakan bahwa, topologi jaringan baru memerlukan waktu epoch yang sedikit untuk *converge* daripada jaringan yang menggunakan pendekatan heuristik. Walaupun demikian, terdapat sedikit kenaikan dalam hal jumlah waktu CPU/Epoch.

Tabel 3. Hasil perbandingan topologi.

Algoritme Pembelajaran dan Parameter	Rataan waktu Epoch	Waktu CPU/Epoch	<i>Generalization Accuracy (%)</i>
traingdm (0.05, 6, 0.4701, 0.5878)	105	0.0020	91.78
traingdm (0.05, 5, 0.3, 0.7)	276	0.0014	90.39

5. Kesimpulan

Penelitian ini dilakukan untuk mengetahui pengaruh penggunaan GA sebagai metode optimasi untuk menemukan topologi jaringan saraf tiruan terbaik pada masalah klasifikasi. Topologi yang dimaksud lebih spesifik pada jumlah neuron yang optimal pada *hidden layer*. Selain itu, dilakukan juga optimasi terhadap learning rate dan parameter momentum. Selanjutnya dilakukan juga perbandingan hasil topologi yang dioptimasi GA dengan versi topologi sebelumnya yang menggunakan pendekatan heuristik.

Berdasarkan hasil simulasi, diindikasikan bahwa topologi yang telah dioptimasi menggunakan GA dan dilatih dengan menggunakan algoritme pembelajaran *traingdm*, secara umum telah dapat meningkatkan tingkat akurasi jaringan saraf tiruan untuk mengatasi masalah klasifikasi dataset tumbuhan Iris. Selain itu, hasil simulasi menunjukkan adanya penurunan rata-rata waktu epoch, walaupun terdapat kenaikan yang tidak begitu signifikan dalam hal waktu CPU/epoch.

Selama proses implementasi GA untuk mengoptimasi topologi jaringan saraf tiruan ini terdapat sebuah kendala yang muncul, yaitu bagaimana menentukan strategi terbaik untuk

menyandi jaringan menjadi kromosom. Solusi dari masalah tersebut adalah dengan menggunakan teknik parametric encoding yang terdapat pada [7].

References

1. OvuncPolat, TulayYildirim: FPGA implementation of a General Regression Neural Network: An embedded pattern classification system, Digital Signal Processing, Volume 20, Issue 3, May 2010, Pages 881-886, ISSN 1051-2004,10.1016/j.dsp.2009.10.013.2.
2. Zarei, A.; DuxingMou, "Artificial Neural Network for Prediction of Ethnicity Based on Iris Texture," Machine Learning and Applications (ICMLA), 201211th International Conference on vol.1, no., pp.514,519, 12-15 Dec. 2012 doi:10.1109/ICMLA.2012.94
3. Michael Negnevitsky. 2011. Artificial Intelligence: A Guide to Intelligent Systems(3st ed.). Pearson Education Limited., England.
4. Fiszlelew, A., Britos, P., Ochoa, A., Merlino, H., Fernández, E., García-Martínez, R.: Finding Optimal Neural Network Architecture Using Genetic Algorithms. Research in Computing Science 27, 2007, pp. 15-24
5. Engelbrecht, A. P., 2007. Computational Intelligence: An Introduction. 2nd Edition. John Wiley & Sons Ltd. England.
6. Frank, A., Asuncion, A. (2010). UCI Machine Learning Repository (<http://archive.ics.uci.edu/ml>). Irvine, CA: University of California, School of Information and Computer Science.
7. Feki, J., Zelinka, I., Burguillo, J. C.: A Review of Methods for Encoding Neural Network Topologies in Evolutionary Computation. Proceedings 25th European Conference on Modelling and Simulation ©ECMS. ISBN: 978-0-9564944-2-9 / ISBN: 978-0-9564944-3-6 (CD)
8. Eiben, A. E., Smith, J. E. Introduction to Evolutionary Computing Generic Algorithm. Power point slide.
9. Ishlah, Muhamad Saad Nurul, A Comparative Study of Neural Network Approaches using Back Propagation Technique on Classification Problem. 2013. Assignment Report of Bio-inspired Computing: Application and Interfaces.