ANALISIS LEARNING RATES PADA ALGORITMA BACKPROPAGATION UNTUK KLASIFIKASI PENYAKIT DIABETES

Thomas Brian
Universitas Trunojoyo Madura
Bangkalan, Indonesia
thomasbrian2112@yahoo.com

Abstrak

Salah satu algoritma klasifikasi pada jaringan syaraf tiruan adalah Backpropagation (BPN). Namun algoritma Backpropagation Standard masih memiliki kekurangan seperti waktu pelatihan yang lama dan terjebak dalam local minimum. Oleh karena itu, pada penelitian ini akan mencoba mengatasi permasalahan tersebut dengan mengajukan metode BPN dengan menggunakan learning rates yang berbeda. Sehingga akan mendapatkan hasil pengukuran yang dicapai berdasarkan waktu training, jumlah iterasi dan akurasi. Dari hasil ujicoba dengan nilai learning rate 0.2 menunjukkan berkurangnya jumlah iterasi 11 kali dibandingkan dengan BPN Standard pada K = 2 fold, waktu training menjadi 4.30 detik pada K = 3 fold dan akurasi masih di atas 80% di dataset Diabetes. Hasil tersebut membuktikan waktu pelatihan Backpropagation menjadi lebih cepat dalam klasifikasi penyakit diabetes dengan mengurangi jumlah iterasinya, tetapi tetap menghasilkan nilai akurasi yang baik.
Kata Kunci: Klasifikasi, Diabetes, Jaringan Syaraf Tiruan, Backpropagation, Learning Rate.

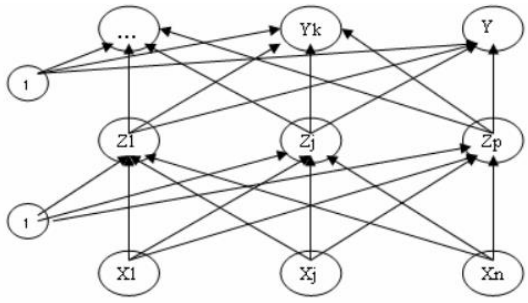
Abstract

One of the classification algorithm is Backpropagation neural network (BPN). But the Standard Backpropagation algorithm still has shortcomings as a long training time and get stuck in a local minimum. Therefore, in this study will try to overcome these problems by proposing a method BPN using different learning rates. So it will get the measurement results achieved by the time training, the number of iterations and accuracy. From the test results with the learning rate value of 0.2 indicates a reduced number of iterations 11 times compared with BPN Standard at K = 2 fold, training time becomes 4.30 seconds at K = 3 fold, and the accuracy is still above 80% in the dataset Diabetes. These results prove the Backpropagation training time to be faster in the classification of diabetes by reducing the number of iteration, but still produce good accuracy value.
Keywords: *Classification, Diabetes, Artificial Neural Networks, Backpropagation, Learning Rate*.

PENDAHULUAN

Di negara Indonesia, dalam bidang kesehatan menjadi hal yang penting. Karena itu diperlukan proses klasifikasi yang baik untuk menentukan seorang pasien terkena penyakit atau tidak. Salah satu penyakit kronis adalah diabetes. Banyak variabel yang perlu didapatkan oleh dokter seperti tekanan darah, sel darah, body mass index, umur, dll. Selanjutkan dilakukan klasifikasi penyakit diabetes.

Salah satu bidang teknik informatika adalah Jaringan Syaraf Tiruan (JST). Jaringan saraf tiruan terinspirasi oleh otak manusia di mana neuron saling interkoneksi secara non-linier. Neuron saling terhubung satu sama lain melalui suatu jaringan. Jaringan ini yang dilatih menggunakan algoritma backpropagation yang mengikuti Gradient Descent Method [1]. Dalam metode Backpropagation, biasanya, digunakan jaringan multilayer. Sebagai contoh, pada Gambar 1 dilustrasikan jaringan dengan sebuah hidden layer. Dalam jaringan, selain terdapat unit input, unit tersembunyi (hidden units) dan output juga terdapat bias yang diberikan pada unit-unit tersembunyi dan output.



Gambar 1. Jaringan saraf backpropagation dengan satu hidden layer

Algoritma backpropagation adalah meto- de pelatihan terawasi dengan meminimalkan error pada outputnya. Backpropagation mem-punyai tahapan yaitu saat umpan maju (feedforward), backpropagasi dan update untuk bobotnya [2]. Kelebihan yang dimiliki backpropagation adalah mengatasi permasala- han pelatihan klasifikasi dengan skala data yang luas dan robust terhadap missing data. Tetapi masih memiliki kekurangan dalam kecepatan konvergen yang buruk dan terjebak pada local minimum. Konvergen yang buruk ini dikarena- kan parameter dalam menentukan topologi jaringan, menentukan nilai bobot awal secara acak, penentuan nilai learning rate dan fungsi aktivasi yang tepat sebelum proses training.

Beberapa penelitian dalam mempercepat proses pelatihan backpropagation dengan menggunakan nilai learning rate yang adaptif di layer input dan outputnya untuk setiap iterasi. Dalam menentukan nilai learning rate yang adaptif dengan menghitung differential error linear dan non linear dari input dan output layer secara terpisah. Sehingga dapat mempercepat konvergen dan meminimalkan error [3]. Improvisasi backpropagation dengan penerapan adaptive learning rate memiliki pengaruh cukup baik namun peningkatan kecepatan tidaklah signifikan sehingga diperlukan pengembangan lainnya yang terfokus pada mempercepat proses iterasi jaringan sehingga cepat menuju ke nilai global optima. Untuk mempercepat proses iterasi pada jaringan backpropagation, banyak teknik yang dapat digunakan yang salah satunya adalah teknik Parallel Training dimana proses pembelajaran dilakukan secara parallel [4]. Pada penelitian lainnya untuk mengurangi waktu training dari backpropagation dengan metode Back-Propagation with Adaptive Learning rate and Momentum term (BPALM), perbedaannya dengan menambahkan faktor momentum yang disesuaikan di setiap iterasinya. Hasil yang diperoleh mendapatkan kecepatan konvergen yang lebih baik [5].

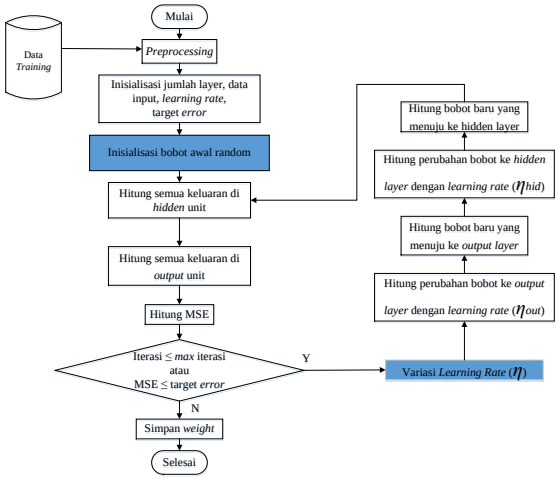
Pada penelitian ini penulis akan menggunakan learning rates yang berbeda

dalam proses klasifikasi Backpropagation. Tujuannya adalah untuk mendapatkan learning rate yang sesuai dalam klasifikasi penyakit Diabetes. Sebagai pembanding dalam menentukan learning rate adalah jumlah iterasi, waktu training dan akurasi dengan metode K-fold cross validation.

METODE PENELITIAN

Pada tahapan penelitian Backpropagatin terdiri dari tahap pembacaan dataset, preprocessing, cross validation, training dan testing. Hasil yang diperoleh adalah jumlah iterasi, waktu training dan akurasi.

Pada tahap preprocessing dilakukan proses penskalaan data dan normalisasi dari missing value sehingga datanya siap untuk dibagi dengan metode K-Fold Cross Validation. Dataset yang digunakan adalah Diabetes.arff yang di download di http://storm.cis.fordham. edu/~gweiss/data-mining/wekadata/diabetes .arff.



Gambar 2. Flowchart Training Backpropagation

Ada tiga fase pelatihan backpropa-gation diantaranya (Siang, 2009) :

A. Propagasi Maju (Fase 1)

Dalam propagasi maju, setiap sinyal masukan (xi) dipropagasikan ke layer tersembunyi menggunakan fungsi aktivasi yang ditentukan. Keluaran dari setiap unit layer tersembunyi (zj) tersebut selanjutnya dipropagasikan maju ke layer tersembunyi diatasnya menggunakan fungsi aktivasi, demikian seterusnya hingga menghasilkan keluaran jaringan (yk).

Berikutnya, keluaran jaringan (yk) dibandingkan dengan target yang harus dicapai (tk). Selisih tk – yk adalah kesalahan yang terjadi. Jika kesalahan ini lebih kecil dari batas toleransi yang ditentukan, maka iterasi dihentikan. Akan tetapi, apabila kesalahan masih lebih besar dari batas toleransinya, maka bobot setiap garis dalam jaringan akan dimodifikasi untuk mengurangi kesalahan yang terjadi.

B. Propagasi Mundur (Fase 2)

Berdasarkan kesalahan tk – yk, dihitung faktor δ_k (k = 1, …, m) yang merupakan unit kesalahan yang akan dipakai dalam perubahan bobot antara output layer (yk) dengan semua hidden layer (zj). Dengan cara yang sama dihitung faktor δ_j di hidden layer sebagai dasar perubahan bobot antara hidden layer (zj) dengan input layer (xi).

C. Perubahan Bobot (Fase 3)

Setelah semua faktor δ dihitung, bobot semua garis dimodifikasi bersamaan. Perubahan bobot suatu garis didasarkan atas faktor δ neuron di layer atasnya.

Sebagai contoh, perubahan bobot garis yang menuju ke layer keluaran didasarkan atas δ_k yang ada di unit keluaran. Ketiga fase tersebut diulang-ulang terus hingga kondisi penghentian dipenuhi. Umumnya, kondisi penghentian yang sering dipakai adalah jumlah iterasi atau kesalahan. Iterasi akan dihentikan jika jumlah iterasi yang dilakukan sudah melebihi jumlah maksimum iterasi yang ditetapkan, atau jika kesalahan yang terjadi sudah lebih kecil dari batas toleransi yang diizinkan.

Algoritma pelatihan untuk backpropaga tion dengan satu hidden layer dan fungsi sigmoid biner adalah sebagai berikut (Siang, 2009):

Langkah 0 : Inisialisasi semua bobot dengan bilangan acak kecil.

Langkah 1 : Jika kondisi penghentian belum terpenuhi, lakukan langkah 2-9.
Langkah 2 : Untuk setiap pasang data pelatihan, lakukan langkah 3-8.
Fase I : Propagasi Maju
Langkah 3 : Tiap unit masukan menerima sinyal dan meneruskannya ke unit tersembunyi diatasnya.
Langkah 4 : Hitung semua keluaran di unit tersembunyi zj (j = 1,2, … ,p).

$$\boldsymbol{z\_net_j = v_{j0} + \sum_{i=1}^{n} x_i v_{ji}}$$

(1.1)

$$\boldsymbol{z_j = f(z\_net_j) = \frac{1}{1 + e^{-z\_net_j}}}$$

(1.2)

Langkah 5 : Hitung semua keluaran jaringan di unit yk (k = 1,2, … ,m).

$$\boldsymbol{y\_net_k = w_{k0} + \sum_{j=1}^{p} z_j w_{kj}}$$

(1.3)

$$\boldsymbol{y_k = f(y\_net_k) = \frac{1}{1 + e^{-y\_net_k}}}$$

(1.4)

Fase II: Propagasi Mundur
Langkah 6 : Hitung faktor δ unit keluaran berdasarkan kesalahan di setiap unit keluaran yk (k = 1,2, … ,m)

$$\boldsymbol{\delta_k = (t_k - y_k) f'(y_{net_k}) = (t_y - y_k) y_k (1 - y_k)}$$

(1.5)

δ_k merupakan unit kesalahan yang akan dipakai dalam perubahan bobot layer di bawahnya (langkah 7).
Hitung suku perubahan bobot w_kj (yang dipakai nanti untuk merubah bobot w_kj) dengan learning rate α.

$$\boldsymbol{\Delta w_{kj} = \alpha \delta_k z_j \ ; \ k = 1, 2, \dots, m \ ; \ j = 0, 1, \dots, p}$$

(1.6)

Langkah 7 : Hitung faktor δ unit tersembunyi (hidden layer) berdasarkan kesalahan di setiap unit tersembunyi zj (j = 1,2, … ,p)

$$\boldsymbol{\delta\_net_j = \sum_{k=1}^{m} \delta_k w_{kj}}$$

(1.7)

faktor $\boldsymbol{\delta}$ hidden layer

$$\boldsymbol{\Delta v_{ji} = \alpha \delta_j x_i \ ; \ j = 1, 2, \dots, p \ ; i = 0, 1, \dots, n}$$

(1.8)

Hitung suku perubahan bobot v_ji (yang dipakai nanti untuk merubah bobot v_ji) dengan learning rate α.

$$\boldsymbol{\Delta v_{ji} = \alpha \delta_j x_i \ ; \ j = 1, 2, \dots, p \ ; i = 0, 1, \dots, n}$$

(1.9)

Fase III: Perubahan Bobot
Langkah 8 : Hitung semua perubahan bobot
Perubahan bobot garis yang menuju ke unit keluaran

$$\boldsymbol{w_{kj}(baru) = w_{kj}(lama) + \Delta w_{kj}, \quad k = 1, 2, \dots, m \ ; \ j = 0, 1, \dots, p}$$

(1.10)

Perubahan bobot garis yang menuju ke unit tersembunyi

$$\boldsymbol{v_{ji}(baru) = v_{ji}(lama) + \Delta v_{ji} \ (j = 1, \dots, p \ ; \ i = 1, 2, \dots, n).}$$

(1.11)

Setelah pelatihan selesai dilakukan, jaringan dapat dipakai untuk klasifikasi dengan dataset yang telah ditentukan. Dalam hal ini, hanya propagasi maju (langkah 4 dan 5) saja yang dipakai untuk menentukan keluaran jaringan. Apabila fungsi aktivasi yang dipakai bukan sigmoid biner, maka langkah 4 dan

5 harus disesuaikan. Demikian juga turunannya pada langkah 6 dan 7.

Dalam beberapa kasus pelatihan yang dilakukan memerlukan iterasi yang banyak sehingga membuat proses pelatihan menjadi lama. Untuk mempercepat iterasi dapat dilakukan mengatur parameter α atau learning rate. Nilai α terletak antara 0 dan 1 ($0 \leq \alpha \leq 1$). Jika nilai α semakin besar, maka iterasi yang dipakai semakin sedikit tetapi menyebabkan pola yang sudah benar menjadi rusak sehingga pemahaman menjadi lambat. Proses pelatihan yang baik dipengaruhi pada pemilihan bobot awal karena bobot awal sangat mempengaruhi apakah jaringan mencapai titik minimum lokal atau global, dan seberapa cepat konvergensinya. Oleh karena itu, dalam backpropagation yang standart, bobot dan bias diisi dengan bilangan acak kecil dan biasanya bobot awal diinisialisasi secara random dengan nilai antara -0,5 sampai 0,5 (atau -1 sampai 1 atau interval yang lainnya) (Amin, 2012).

Untuk mengetahui kapan proses training akan berhenti, dapat dilakukan dengan cara membatasi jumlah iterasi atau menentukan nilai Mean Square Error (MSE) antara target output yang harus dicapai (tk) dengan output keluaran jaringan (yk). Jika terdapat sebanyak m training data, maka untuk menghitung Mean Square Error digunakan persamaan berikut:

$$\boldsymbol{MSE = \frac{\sum_{i=1}^{m}(t_k - y_k)^2}{m}} \quad (1.12)$$

HASIL DAN PEMBAHASAN

Uji coba pada penelitian ini dilakukan pada PC dengan spesifikasi *CPU i7 2.20 GHz* dan *RAM 4GB* dengan *IDE Matlab 2015b*. Menggunakan dataset *Diabetes*, selanjutnya dibagi dengan *cross validation K = 2, 3, 5, 7* dan *10*. Hasil perbandingan yang dipakai adalah jumlah iterasi, waktu *training* dan akurasi. Jumlah iterasi didapatkan dari hasil pelatihan *backpropagation*. Satu kali iterasi dimulai dari inisialisasi bobot sampai dengan *backforward*. Untuk waktu training diperoleh dari selisih antara waktu awal sebelum proses training dan sesudah proses *training* selesai. Evaluasi waktu *training* mengikuti persamaan berikut

$$waktu\ training\ (s) = waktu\ selesai - waktu\ awal \quad (1.13)$$

Untuk akurasinya dengan cara menghitung jumlah hasil klasifikasi benar dibandingkan dengan jumlah data di setiap dataset. Evaluasi akurasi mengikuti persamaan

$$\%\ akurasi = \frac{jumlah\ klasifikasi\ benar}{jumlah\ data\ di\ dataset} \times 100\% \quad (1.14)$$

Berikut adalah parameter dalam menentukan Backpropagation (BPN) sebagai acuan untuk dibandingkan dengan metode yang diajukan dengan pemodelan jaringan pada Tabel 1.

Tabel 1. Optimasi Parameter BPN

	Optimasi
Dataset	Diabetes (768 data)
Pembagian Dataset	Metode *K-Fold Cross Validation* *K = (2,3,5,7,10)* dengan 5 kali *running*
Jumlah *layer*	*1 input layer* *1 hidden layer* *1 output layer*
Neuron	Jumlah neuron di *input* dan *hidden layer* sesuai dengan jumlah *input* dataset dengan satu neuron di *output layer*
Learning rate	0.1, 0.2, 0.3
Target error	0.17

Hasil yang diukur adalah jumlah iterasi, waktu training dan akurasi dengan 5 kali running.

Tabel 2. Hasil Ujicoba Jumlah Iterasi Dataset Diabetes

Fold	**Rata-rata Iterasi**

	BPN Std	BPN ALR		
		0.1	0.2	0.3
2	47.70	47.10	35.9	44.20
3	35.7	36.2	29.7	37.10
5	53.20	54.30	45.23	55.67
7	23.39	24.34	20.1	26.12
10	55.66	57.12	45.33	50.50

Tabel 3. Hasil Ujicoba Waktu Training Dataset Diabetes

Fold	**Rata-rata Waktu Training**			
	BPN Std	BPN ALR		
		0.1	0.2	0.3
2	4.39	2.67	2.14	3.10
3	5.57	5.12	4.30	6.11
5	6.89	2.13	2.58	2.45
7	5.67	6.11	4.15	5.12
10	3.02	3.45	2.11	4.45

Tabel 4. Hasil Ujicoba Akurasi Dataset Diabetes

Fold	**Rata-rata Akurasi**			
	BPN Std	BPN ALR		
		0.1	0.2	0.3
2	74.57	84.90	86.23	85.00
3	74.18	86.21	87.11	87.00
5	73.38	80.12	84.90	82.22
7	74.76	80.12	89.90	82.11
10	73.81	89.45	93.00	83.89

Pada pengujian *BPN ALR* untuk error setiap iterasi tidak terlalu besar dibandingkan target *error*. Maka penurunan *error rate* setiap iterasi akan cepat berkurang sampai di bawah target error. Jadi jumlah iterasi *BPN ALR* menjadi lebih sedikit dari *BPN Standard*. Waktu *training BPN ALR* bisa menjadi lebih lama setiap iterasi. Tetapi karena jumlah iterasi lebih sedikit, maka total waktu *training BPN ALR* lebih cepat daripada *BPN Standard*. Hasil akurasi *BPN ALR* masih dibawah *BPN Standard*. Karena pada metode ALR jumlah atribut input dan jumlah data pada dataset berpengaruh pada proses training *Backpropa gation* (BPN).

KESIMPULAN DAN SARAN

Berdasarkan hasil ujicoba dengan learning rate yang berbeda (0.1, 0.2, 0.3) dan analisis terhadap dataset, modifikasi *Backpropagation* (BPN) dilakukan metode *Adaptive Learning Rate* (ALR). Hasil tersebut ditunjukkan dengan menggunakan learning rate 0.2 hasilnya berkurangnya jumlah iterasi 11 kali dibandingkan dengan *BPN Standard* pada *K = 2 fold*, waktu *training* menjadi 4.30 detik pada *K = 3 fold* dan akurasi masih di atas 80% di dataset Diabetes. Jadi metode penggabungan yang diusulkan yaitu *BPN ALR* secara keseluruhan dari skenario ujicoba *1, 2* dan *3* dapat mengurangi jumlah iterasi, dengan jumlah data *training* yang banyak seperti ujicoba pada *10 Fold*. Sehingga waktu *training* menjadi lebih cepat dan akurasi yang baik dibandingkan dengan *BPN Standard* pada ujicoba *K-Fold Cross Validation* untuk semua *K= 2, 3, 5, 7* dan *10* di dataset Diabetes.

Beberapa saran yang diajukan untuk pengembangan lebih lanjut adalah mencoba metode penggabungan lain dalam modifikasi *backpropagation* untuk mempercepat *training* dan menambahkan beberapa dataset dalam skenario ujicoba.

.

DAFTAR PUSTAKA

[1] Ihwan, Andi, 2013, Metode Jaringan Saraf Tiruan Propagasi Balik untuk Estimasi Curah Hujan Bulanan di Ketapang Kalimantan Barat, Prosiding Semirata FMIPA Universitas Lampung.

[2] Hermawan, A. 2006. "Jaringan Syaraf Tiru an, Teori, dan Aplikasi". Yogyakarta : ANDI.

[3] Subavathi Jeyaseeli, S., Kathirvalavakumar T. 2011. "Adaptive Modified Backpro-pagation Algorithm Based On Diffe-

rential Errors". International Journal of Computer Science, Engineering and Applications (IJCSEA) Vol.1, No.5.

[4] Khairani, Mufida. 2014. "Improvisasi Back Propagation Menggunakan Penerapan Adaptive Learning Rate dan Parallel Training". TECHSI Vol 4. Nomor 1 2014 : Jurnal Penelitian Teknik Informatika Universitas Sumatera Utara.

[5] ChengYu, Chien., Bin-Da Liu. 2002. "A Backpropagation Algorithm with Adaptive Learning Rate and Momentum Coefficient". IEEE Journal. Department of Electrical Engineering. National Cheng Kung University, Taiwan.

[6] Siang, J.J (2009), "Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan MATLAB". Yogyakarta : ANDI.

[7] Subavathi Jeyaseeli, S., Kathirvalavakumar T. 2011. "Adaptive Modified Backpropagation Algorithm Based On Differential Errors". International Journal of Computer Science, Engineering and Applications (IJCSEA) Vol.1, No.5.

[8] Khairani, Mufidah. 2014. "Improvisasi Back Propagation Menggunakan Penerapan Adaptive Learning Rate dan Parallel Training". TECHSI Vol 4. Nomor 1 2014 : Jurnal Penelitian Teknik Informatika Universitas Sumatera Utara.

[9] ChengYu, Chien., Bin-Da Liu. 2002. "A Backpropagation Algorithm with Adaptive Learning Rate and Momentum Coefficient". IEEE Journal. Department of Electrical Engineering. National Cheng Kung University, Taiwan.

[10] Iranmanesh, Saeid. 2009. "Differential adaptive learning rate method for backpropagation neural networks," World Academy of Science, Engineering and Technology 50 285-288.

[11] http://storm.cis.fordham.edu/~gweiss/data-mining/weka-data/diabetes.arff diakses tanggal 16 September 2016