

ARSITEKTUR DAN ALGORITMA JARINGAN SARAF TIRUAN UNTUK Mendukung PREDIKSI GANGGUAN GEOMAGNET

John Maspupu
Pusat Sains Antariksa-LAPAN
e-mail: john_mspp@yahoo.com

Abstrak. Makalah ini membahas suatu arsitektur serta algoritma JST atau jaringan syaraf tiruan yang menggunakan umpan maju dan propagasi mundur (*feedforward and backpropagation*) dengan satu lapisan tersembunyi (*hidden layer*) untuk mendukung sistem prediksi gangguan geomagnet. Algoritma JST yang telah dirancang antara lain adalah : algoritma penentuan bobot awal, algoritma pelatihan dengan umpan maju (*feedforward*), algoritma perambatan mundur dari galat (*backpropagation of error*) serta algoritma perbaikan bobot dan bias.

Kata kunci : *Arsitektur , Algoritma , JST , Gangguan , Geomagnet.*

Abstract. This paper discusses an architecture and algorithm on artificial neural network (ANN) by using feedforward and backpropagation with one hidden layer for support the prediction system of geomagnetic disturbance. We design the algorithm of the artificial neural network which are algorithm of determination an initial weight, algorithm of training by using feedforward, backpropagation algorithm of error and algorithm of weight correction and biased.

Keywords : *Architecture, Algorithm, ANN, Geomagnetic disturbance*

1. Pendahuluan

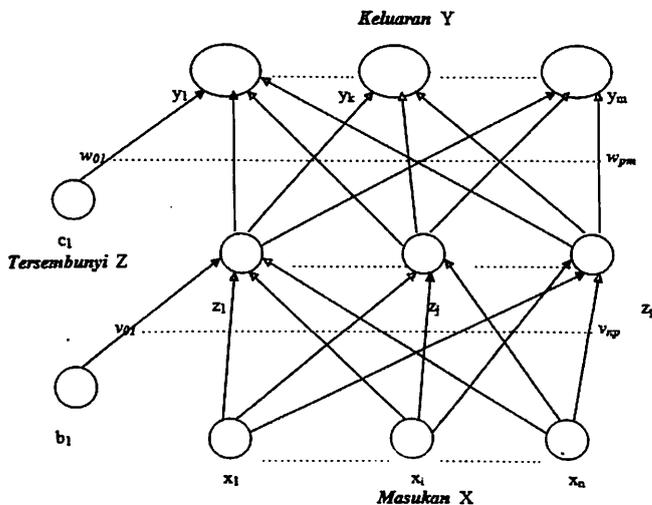
Jaringan syaraf tiruan atau singkatnya JST adalah terjemahan dari ANN (*artificial neural network*) dan merupakan model sistem komputasi yang dapat menirukan cara kerja jaringan syaraf biologis (Brunak dan Lautrup, 1990). JST juga telah dikembangkan sebagai generalisasi model matematika dari *human cognition* atau jaringan syaraf biologis dengan asumsi-asumsi tertentu (Feldman et al., 1988). Algoritma propagasi mundur ini pertama kali diperkenalkan dan dikembangkan oleh Rumelhart dan Mc Clelland (1988). Sedangkan beberapa aplikasi lainnya pada pengembangan model variasi harian geomagnet regional dan prediksi badai geomagnet dapat dilihat di Sutcliffe (2000) dan Kugblenu et al. (1999). Dari beberapa informasi pada referensi di atas, timbul pemikiran untuk membangun arsitek dan algoritma JST sebagai alat (*tools*) komputasi untuk memprediksi tingkat gangguan geomagnet. Dengan demikian tujuan pembahasan makalah ini adalah jelas untuk menentukan atau merancang algoritma JST sebagai alat komputasi prediksi gangguan geomagnet.

Selain itu, variasi harian geomagnet merupakan hasil penjumlahan dari variasi hari tenang dan gangguan geomagnet.

Dengan perkataan lain gangguan geomagnet adalah selisih antara variasi harian geomagnet dengan variasi hari tenang. Untuk mendapatkan tingkat gangguan geomagnet kita perlu mengetahui variasi hari tenang geomagnet. Dengan demikian untuk memprediksi tingkat gangguan geomagnet, cukup diprediksi variasi hari tenangnya dan kemudian dibandingkan dengan hasil-hasil pemantauan variasi harian geomagnet di stasiun pengamatan tertentu. Komponen masukan yang digunakan dalam sistem JST ini adalah waktu lokal (*local time*), posisi lintang, urutan bilangan variasi hari tenang (*quiet time daily variations number*) dan bilangan bintang matahari (*sunspot number*). Sedangkan yang menjadi target adalah data variasi hari tenang dari stasiun pengamatan Lapan. Namun yang menjadi masalah adalah membangun atau membentuk arsitektur propagasi mundur maupun prosedur penentuan bobot awal dari JST tersebut. Kemudian menyusun algoritma pelatihan umpan maju (*feedforward*) dan perambatan mundur dari galat (*backpropagation of error*) serta menyusun algoritma untuk memperbaiki bobot dan bias sistem tersebut. Manfaat dari hasil-hasil kajian maupun konstruksi algoritma JST ini adalah sebagai suatu perangkat lunak (*software*) yang dapat diterapkan pada sistem prediksi tingkat gangguan geomagnet.

2. Data dan Metode

Data yang digunakan adalah data simulasi dari JST atau jaringan syaraf tiruan yang menggunakan fungsi propagasi mundur (*back propagation*) dengan satu lapisan tersembunyi (*hidden layer*) algoritma penentuan bobot awal, algoritma pelatihan dengan umpan maju (*feedforward*), algoritma perambatan mundur dari galat (*backpropagation of error*) serta algoritma perbaiki bobot dan bias.



Gambar 2-1. JST propagasi mundur dengan satu lapisan tersembunyi.

Nomenklatur

$X = (x_1, \dots, x_i, \dots, x_n)$ masukan sebagai vektor pelatihan.

$T = (t_1, \dots, t_i, \dots, t_m)$ keluaran sebagai vektor target.

δ_k adalah galat (*error*) pada unit y_k sebagai koreksi bobot w_{jk} ($z_j \xrightarrow{w_{jk}} y_k$).

δ_j adalah galat (*error*) pada unit z_j sebagai koreksi bobot v_{ij} ($x_i \xrightarrow{v_{ij}} z_j$).

α adalah tingkat pembelajaran ($0 < \alpha \leq 1$).

X_i adalah unit masukan ke i .

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

Z_j adalah unit tersembunyi ke j .

$z_{-in j}$ artinya jaringan dari masukan sinyal x_i ke unit Z_j .

z_j artinya sinyal keluaran (aktivasi) dari unit Z_j , jadi $z_j = f(z_{-in j})$.

w_{0k} adalah bias pada unit keluaran ke k .

Y_k adalah unit keluaran ke k .

$y_{-in k}$ artinya jaringan dari masukan sinyal z_j ke unit Y_k .

y_k artinya sinyal keluaran (aktivasi) dari unit Y_k , jadi $y_k = f(y_{-in k})$.

3. Hasil dan Pembahasan

Lebih dahulu akan diperkenalkan suatu relasi empiris antara faktor skala (*scale factor*), banyaknya unit masukan (*input unit*) dan banyaknya unit tersembunyi (*hidden unit*). Relasi tersebut dinyatakan dalam bentuk persamaan matematis berikut:

$\beta = 0,7\sqrt{p} = 0,7(p)^{\frac{1}{n}}$ yang dikenal sebagai formulasi *Nguyen-Widrow* (Nguyen dan Widrow, 1989; Fausett, 1994).

Dalam hal ini, β adalah faktor skala, n adalah banyaknya unit masukan dan p adalah banyaknya unit tersembunyi. Bobot awal v_{0j} (dimulai dari unit masukan ke unit tersembunyi) merupakan bilangan acak (*random number*) yang dipilih antara faktor $-\beta$ dan β . Selanjutnya algoritma penentuan bobot awal adalah sebagai berikut :

L_1 : Tentukan v_{0j} (*lama*) yaitu pilihan bilangan acak antara $-\beta$ dan β (atau antar $-0,5$ dan $0,5$ atau antara -1 dan 1).

L_2 : Hitung norm v_j (*lama*) atau ditulis: $\|v_j(\textit{lama})\|$.

L_3 : Iterasi bobot awal $v_{ij} = \frac{\beta v_{ij}(\textit{lama})}{\|v_j(\textit{lama})\|}$ untuk $j = 1, \dots, p$ dan $i = 1, \dots, n$.

Selanjutnya diperkenalkan juga suatu relasi antara akurasi (e), banyaknya pola latihan (p) dan banyaknya bobot yang digunakan untuk pelatihan (w).

Relasi tersebut dinyatakan dalam bentuk persamaan matematis berikut: $e = \frac{w}{p}$ (3-1)

Bukti dari relasi (3-1) ini dapat dibaca pada Baum dan Haussler (1989). Jika $p = 800$ dengan akurasi yang diharapkan sebesar 0,1 maka diperlukan banyaknya bobot sebesar 80.

3.1. Algoritma Pelatihan dengan umpan maju (*feedforward*)

L₀ : Penentuan bobot awal (himpunkan nilai-nilai acak yang kecil dari -0,5 s/d 0,5).

L₁ : Setiap unit masukan (X_i , $i = 1, \dots, n$) menerima sinyal x_i dan meneruskan ke semua unit tersembunyi (Z_j , $j = 1, \dots, p$).

L₂ : Hitung $z_{-inj} = v_{0j} + \sum_{i=1}^n x_i v_{ij}$.

L₃ : Gunakan fungsi aktivasi untuk menghitung sinyal keluaran $z_j = f(z_{-inj})$.
Teruskan sinyal tersebut ke semua unit keluaran dalam lapisan keluaran.

L₄ : Setiap unit tersembunyi (Z_j , $j = 1, \dots, p$) menerima sinyal z_j dan meneruskan ke semua unit keluaran (Y_k , $k = 1, \dots, m$).

L₅ : Hitung $y_{-ink} = w_{0k} + \sum_{j=1}^p z_j w_{jk}$.

L₆ : Gunakan fungsi aktivasi untuk menghitung sinyal keluaran $y_k = f(y_{-ink})$.

3.2. Algoritma Perambatan Mundur dari galat (*backpropagation of error*)

L₇ : Setiap unit keluaran (Y_k , $k = 1, \dots, m$) menerima suatu pola target yang berkorespondensi dengan masukan pola pelatihan. Hitung galat informasinya dalam bentuk $\delta_k = (t_k - y_k) f'(y_{-ink})$.

L₈ : Hitung koreksi bobotnya (digunakan untuk memperbaiki w_{jk} terakhir).
Jadi $\Delta w_{jk} = \alpha \delta_k z_j$.

L₉ : Hitung koreksi biasnya (digunakan untuk memperbaiki w_{0k} terakhir).
Jadi $\Delta w_{0k} = \alpha \delta_k$, kemudian teruskan δ_k ke unit-unit di lapisan tersembunyi.

L₁₀ : Hitung $\delta_{-inj} = \sum_{k=1}^m \delta_k w_{jk}$, untuk setiap unit tersembunyi (Z_j , $j = 1, \dots, p$).

L₁₁ : Kemudian kalikan dengan turunan fungsi aktivasinya untuk menghitung galat informasinya dalam bentuk $\delta_j = \delta_{-inj} f'(z_{-inj})$.

L₁₂ : Hitung koreksi bobotnya (digunakan untuk memperbaiki v_{ij} terakhir).
Jadi $\Delta v_{ij} = \alpha \delta_j x_i$.

L₁₃ : Hitung koreksi biasnya (digunakan untuk memperbaiki v_{0j} terakhir).
Jadi $\Delta v_{0j} = \alpha \delta_j$.

3.3. Algoritma Perbaiki Bobot dan Bias

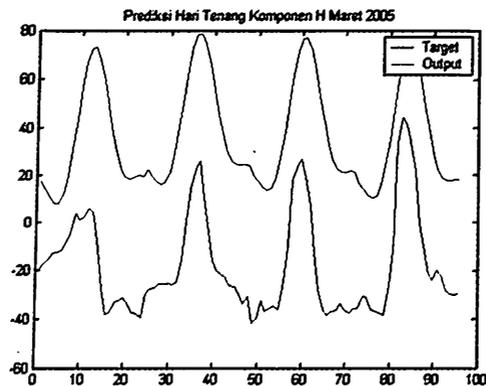
L₁₄ : Perbaiki bobot dan bias ($j = 0, \dots, p$) untuk setiap unit keluaran (Y_k , $k = 1, \dots, m$).
Jadi $w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk}$.

L₁₅ : Perbaiki bobot dan bias ($i = 0, \dots, n$) untuk setiap *hidden unit* (Z_j , $j = 1, \dots, p$).

Jadi $v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij}$.

L₁₆ : Selesai (Test Stopping Condition) !!!

Salah satu aplikasi dari RBF pada jaringan saraf tiruan dalam kegiatan prediksi pola hari tenang geomagnet. Contohnya seperti ditunjukkan pada Gambar 3-1.



Gambar 3-1. Plot antara target dan output untuk data prediksi (Maret 2005)

Pada gambar tersebut terlihat bahwa pola hasil prediksi mengikuti pola data target. Akan tetapi terdapat simpangan amplitudo sebesar 40-50 nT. Melihat hasil dari pelatihan dan pengujian untuk komponen D yang kurang baik, maka tidak dilakukan prediksi untuk komponen D.

4. Kesimpulan

Algoritma JST yang telah dirancang ini perlu diuji keandalannya dengan data variasi harian geomagnet terdahulu untuk mengetahui seberapa besar akurasi hasil prediksinya. Selain itu keempat algoritma yaitu: algoritma penentuan bobot awal, algoritma pelatihan dengan umpan maju (*feedforward*), algoritma perambatan mundur dari galat (*backpropagation of error*) serta algoritma perbaiki bobot dan bias yang telah dijelaskan dalam butir pembahasan (butir 3), ini perlu diterjemahkan ke dalam program komputer sehingga dapat diimplementasikan pada data masukan gangguan geomagnet yang siap dipakai (*real time data or near real time data*).

Daftar Rujukan

Baum, E.B. and Haussler, D., 1989, *What size net gives valid generalization?*, Neural Computation Journal 1, pp. 151 – 160.

- Brunak S. and Lautrup B., 1990, *Neural Network Computers with Instuition*, World Scientific, Singapore.
- Fausett L., 1994, *Fundamentals of Neural Networks : Architectures, Algorithms and Applications*, Prentice Hall, Englewood Cliffs, New Jersey.
- Feldman J., M. A. Fanty and N. H. Goddard, 1988, *Computing with Structured Neural Networks*, Journal of Computer, Vol. 21, No.3, pp.91-103.
- Kugblenu S., Taguchi S., and Okuzawa T., 1999, *Prediction of the geomagnetic storm associated Dst index using ANN-algorithm*, Journal EPS, 51, pp.303 – 313.
- Nguyen D. and Widrow B., 1989, *The truck backer-upper: An example of self-learning in neural networks*, International Joint Conference on neural Networks, Vol. 2, pp.357 – 363, Washington, DC.
- Rumelhart D.E. and Mc Clelland J.L., 1988, *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*, MIT Press, Cambridge, Mass.
- Sutcliffe P. R., 2000, *The development of a regional geomagnetic daily variation model using neural networks*, Ann. Geophys., 18, pp. 120 – 132.