

PERBANDINGAN MODEL FEED FORWARD NEURAL NETWORK DAN GENERALIZED REGRESSION NEURAL NETWORK PADA DATA NILAI TUKAR YEN TERHADAP DOLAR AS

Budi Warsito

Program Studi Statistika, Jurusan Matematika, Universitas Diponegoro
Jl. Prof . Sudarto Kampus UNDIP Tembalang, Semarang

Abstrak: Beberapa tulisan yang berkaitan dengan pemodelan Neural Network (NN) pada time series lebih banyak difokuskan pada pemodelan Feed Forward Neural Network (FFNN). Dengan berbagai keunggulan yang dimilikinya model ini dalam berbagai aplikasi mampu membuat prediksi lebih baik daripada model linear ARIMA. Pada perkembangan pemodelan NN yang lain, berkembang model General Regression Neural Network (GRNN) yang desainnya diadopsi dari fungsi Gaussian multivariate yang telah diperluas. Model GRNN telah banyak dikembangkan untuk berbagai masalah statistika baik untuk output multivariate maupun univariat. Tulisan ini membahas prosedur pemodelan FFNN dan GRNN kemudian melakukan studi perbandingan keakuratan prediksi dari kedua model pada penerapan bidang financial yaitu nilai tukar mata uang Yen Jepang terhadap dolar AS. Hasil studi terhadap data ini menunjukkan bahwa untuk data ini model GRNN relatif lebih unggul daripada model FFNN.

Kata Kunci: FFNN, GRNN, time series, nilai tukar

PENDAHULUAN

Pemodelan Neural Network (NN) telah banyak diterapkan pada berbagai aplikasi. Dalam masalah statistika, salah satu yang banyak diterapkan adalah dalam bidang time series. Beberapa tulisan yang berkaitan dengan pemodelan NN pada time series lebih banyak difokuskan pada pemodelan Feed Forward Neural Network (FFNN). Dengan berbagai keunggulan yang dimilikinya model ini mampu membuat prediksi dengan baik, lebih baik daripada model linear ARIMA seperti telah dilakukan oleh Portugal[4], Kao and Huang [2], dan Chatfield and Faraday [1]. Pada perkembangan pemodelan NN yang lain, Specht dalam Leung, et al [3] mengusulkan dan mengembangkan model General Regression Neural Network (GRNN) yang desainnya diadopsi dari fungsi Gaussian multivariate yang telah diperluas. Model GRNN telah banyak dikembangkan untuk berbagai masalah statistika baik untuk output multivariate maupun univariat. Diantaranya adalah Leung, et al [3] yang melakukan prediksi nilai tukar beberapa mata uang internasional dan membandingkannya dengan model MLFN dan model fungsi transfer.

Sistematika tulisan ini disusun sebagai berikut. Pada bagian 2 dan 3 masing-masing dibahas prosedur pemodelan FFNN dan GRNN. Kemudian pada bagian 4 memuat studi perbandingan keakuratan prediksi dari kedua model pada penerapan bidang finansial yaitu nilai tukar mata uang yen Jepang terhadap dolar AS. Hasil perhitungan dan grafik disajikan pada bagian ini. Bagian 5 berisi kesimpulan.

FEED FORWARD NEURAL NETWORK (FFNN)

Pelatihan jaringan pada model FFNN menggunakan algoritma backpropagation yang meliputi tiga tahap yaitu umpan maju (*feedforward*) dari pola input, penghitungan dan propagasi balik dari error dan penyesuaian bobot. Pada tahap umpan maju setiap unit input menerima sinyal input (x_i) dan menyebarkannya ke unit tersembunyi z_1, \dots, z_p . Setiap unit tersembunyi menghitung aktivasinya dan jumlah terboboti dari input-inputnya dalam bentuk

$$z_{in_j} = \sum_i w_{ji} x_i + w_{bj} \quad (1)$$

dimana x_i adalah aktivasi dari unit input ke- i yang mengirimkan sinyal ke unit hidden ke j , w_j adalah bobot dari sinyal yang terkirim dan $j = 1, 2, \dots, q$ adalah jumlah unit hidden. Hasil penjumlahan ditransformasi dengan fungsi aktivasi nonlinear $f(\cdot)$

$$z_j = f\left(z_{in_j}\right) \quad (2)$$

Setelah semua unit tersembunyi menghitung aktivasinya kemudian mengirimkan sinyal (z_j) ke unit output. Kemudian unit output menghitung aktivasinya dalam bentuk

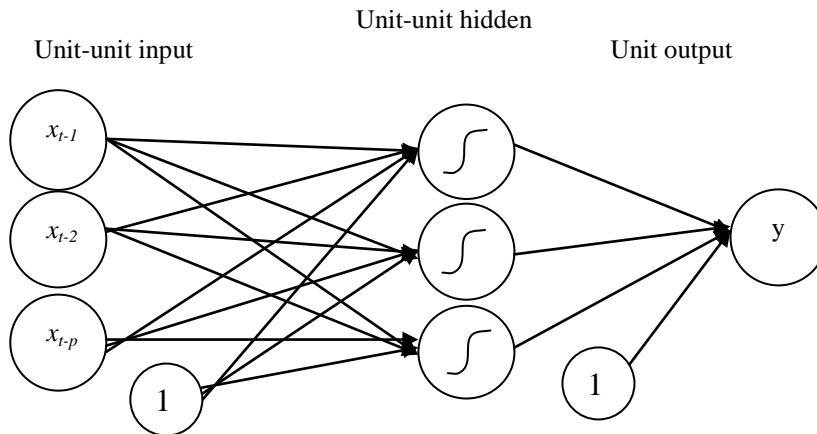
$$g(w, z) = \sum_j w_{jo} z_j + w_{bo} \quad (3)$$

Fungsi pada (3) merupakan nilai output dari jaringan yaitu

$$y = \sum_j w_{jo} f(a_j) + w_{bo} \quad (4)$$

dimana w_{bo} adalah bobot dari bias ke unit output.

Arsitektur model FFNN dengan unit input lag 1 sampai p dan unit konstan, satu *hidden layer* dengan 3 neuron dan 1 unit output diilustrasikan pada gambar berikut



Gambar 1. Arsitektur model FFNN untuk prediksi data time series dengan satu hidden layer yang terdiri 3 neuron dan variabel input nilai pada lag 1, 2, ..., p

Model FFNN dengan satu hidden dan input x_{t-1}, \dots, x_{t-p} ditulis dalam bentuk

$$\hat{x}_t = \psi_0 \left\{ w_{co} + \sum_n w_{no} \psi_n \left(w_{cn} + \sum_i w_{in} x_{t-j_i} \right) \right\} \quad (5)$$

dimana $\{w_{cn}\}$ adalah bobot antara unit konstan dan neuron dan w_{co} adalah bobot antara unit konstan dan output. $\{w_{in}\}$ dan $\{w_{no}\}$ masing-masing menyatakan bobot koneksi input dengan neuron dan antara neuron dengan output. Kedua fungsi ψ_n dan ψ_o masing-masing fungsi aktivasi yang digunakan pada neuron dan output. Notasi untuk model FFNN adalah $NN(j_1, \dots, j_k, n)$ yang menyatakan NN dengan input lag j_1, \dots, j_k dan n neuron.

GENERAL REGRESSION NEURAL NETWORK (GRNN)

GRNN pada awalnya diusulkan dan dikembangkan oleh Specht dalam Leung, et al [3]. Dasar dari operasi GRNN secara esensial didasarkan pada teori regresi nonlinear (kernel) dimana estimasi dari nilai harapan output ditentukan oleh himpunan input-inputnya. Walaupun GRNN menghasilkan output berupa vektor multivariat, dengan tidak mengurangi keumuman deskripsi dari logika operasi GRNN pada tulisan ini disederhanakan untuk kasus output univariat. Persamaan (1) meringkas logika GRNN dalam formula regresi nonlinear :

$$E[y|X] = \frac{\int_{-\infty}^{\infty} y f(X, y) dy}{\int_{-\infty}^{\infty} f(X, y) dy} \quad (6)$$

Dimana y adalah output yang diprediksi oleh GRNN, X adalah vektor input (x_1, x_2, \dots, x_n) yang terdiri dari n variabel prediktor, $E[y|X]$ harga harapan dari output y jika diberikan vektor input X dan $f(X, y)$ fungsi densitas probabilitas bersama dari X dan y.

Topologi GRNN

Topologi GRNN pada prinsipnya terdiri dari empat layer unit pemrosesan (*neuron*). Tiap-tiap layer unit pemrosesan ditandai dengan suatu fungsi komputasional yang spesifik. Layer pertama disebut neuron input (*input neurons*), bertanggung jawab untuk menerima informasi. Terdapat suatu neuron input tunggal untuk setiap variabel prediktor dalam vektor input X. Tidak ada pemrosesan data yang dilakukan pada neuron-neuron input tersebut. Neuron input kemudian mengirimkan data ke layer kedua dari unit pemrosesan yang disebut neuron pola (*pattern neurons*). Neuron pola digunakan untuk memproses data secara sistematis sedemikian hingga hubungan yang tepat antara input dan respon bersifat “memorized”. Dalam hal ini, jumlah neuron pola sama dengan jumlah kasus dalam himpunan training. Neuron pola i mendapatkan data dari neuron input dan menghitung output θ_i menggunakan fungsi transfer :

$$\theta_i = e^{-(X-U_i)(X-U_i)/2\sigma^2} \quad (7)$$

Dimana X adalah vektor input dari variabel prediktor untuk GRNN, U_i adalah vector training khusus yang direpresentasikan oleh neuron pola i dan σ adalah parameter smoothing.

Output dari neuron pola kemudian diteruskan ke layer ketiga dari unit pemrosesan yang disebut neuron jumlahan (*summation neurons*) dimana output dari semua neuron pola ditambahkan. Secara teknis ada dua tipe penjumlahan yang dibentuk dalam *summation neurons* yaitu penjumlahan aritmatik sederhana dan penjumlahan terboboti. Dalam topologi GRNN terdapat unit pemrosesan terpisah yang melakukan penjumlahan aritmatik sederhana dan penjumlahan terboboti. Persamaan (8a) dan (8b) masing-masing menyatakan operasi matematis yang dibentuk oleh penjumlahan sederhana dan penjumlahan terboboti.

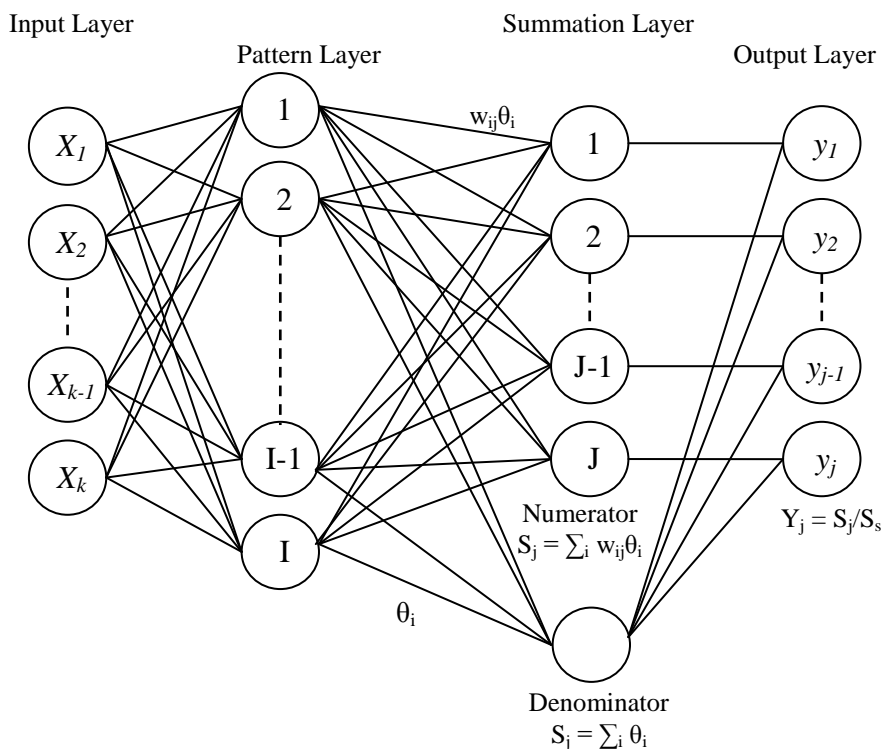
$$S_s = \sum_i \theta_i \quad (8a)$$

$$S_w = \sum_i w_i \theta_i \quad (8b)$$

Jumlahan yang dihasilkan oleh *summation neurons* secara berturut-turut dikirimkan ke layer ke empat dari unit pemrosesan yaitu neuron output. Neuron output kemudian membentuk pembagian berikut untuk mendapatkan

output regresi GRNN y :
$$y = \frac{S_w}{S_s} \quad (9)$$

Gambar 2 mengilustrasikan skema dari desain GRNN dan logika operasionalnya sebagaimana telah dijelaskan di atas. Perlu dicatat bahwa konstruksi jaringan yang digambarkan akan valid untuk berbagai model dengan output multivariat. Desain khusus untuk data time series hanya mempunyai satu (univariat) neuron output.



Gambar 2. Skema dari desain GRNN dan logika operasionalnya

Gambar 2. Konstruksi GRNN secara umum. Konstruksi GRNN terdiri dari empat layer unit pemrosesan yaitu neuron input, pattern, summation dan output. Input layer menerima vector input X dan mendistribusikan data ke pattern layer. Tiap-tiap neuron dalam pattern layer kemudian membangun output θ dan mengirimkan hasilnya ke summation layer. Neuron-neuron numerator dan denominator summation layer menghitung jumlahan aritmatika sederhana dan terboboti yang didasarkan pada nilai θ dan w_{ij} yang diperoleh berdasarkan pembelajaran melalui training dengan supervise. Neuron-neuron pada output layer kemudian melakukan pembagian terhadap jumlahan yang telah dihitung oleh neuron-neuron pada summation layer.

TERAPAN PADA DATA FINANSIAL

Dipunyai data harian nilai tukar (*exchange rate*) Yen Jepang terhadap dolar AS sebanyak 363 data kemudian dilakukan differencing sehingga tinggal diperoleh 362 data. Data dibagi menjadi dua sebagai training dan cecking. Dicobakan dua model yaitu lag 1 dan 2 atau AR(2) serta lag 1 saja sebagai input atau AR(1). Untuk model AR(2) akan digunakan 300 data sebagai training dan 60 data sebagai cecking (data yang digunakan mulai lag 3). Sedangkan untuk model AR(1) akan digunakan 301 data sebagai training dan 60 data sebagai cecking (data yang digunakan mulai lag 2).

Pengolahan data dilakukan menggunakan program Matlab 6.5 dengan perintah `newgrnn` dan `newff` masing-masing untuk GRNN dan FFNN. Prediksi dari peramalan nilai tukar untuk seleksi model atau periode penentuan dimulai setelah jaringan dilatih. Sebanyak 60 data peramalan kemudian dibandingkan dengan observasi actual dan parameter jaringan yang telah disesuaikan. Setelah itu model dipilih berdasarkan statistik performansi yang telah ditentukan. Kriteria evaluasi yang digunakan sebagai pembandingan dipilih mean absolute error (MAE), root mean square error (RMSE), mean absolute error (MSE) dan koefisien determinasi (R^2) dari regresi antara output dengan nilai actual.

Setelah dengan beberapa kali trial pada model FFFN hanya akan dipilih model dengan lima unit pada hidden layer. Fungsi aktivasi yang digunakan adalah logistic sigmoid, pelatihan menggunakan teknik gradient descent. Code program dibatasi sampai 1000 epoch dan $mse = 10^{-3}$.

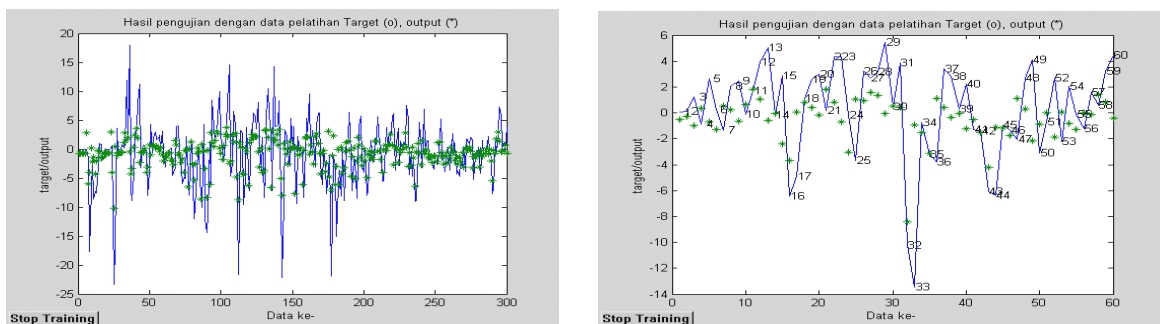
Pada pelatihan jaringan GRNN untuk model GRNN-AR(2) jaringan yang terbentuk terdiri dari dua unit input masing-masing dengan 300 data atau vector inputnya dapat ditulis sebagai $X_1 = (x_1, x_2, \dots, x_{300})$ dan $X_2 = (v_1, v_2, \dots, v_{300})$ serta terdapat $302 - 2 = 300$ neuron pada *pattern layer*, sama dengan jumlah data pada himpunan data training. Setelah pelatihan yang terawasi telah lengkap jaringan regresi akan menghitung peramalan nilai tukar pada periode t berdasarkan nilai-nilai pada variable predictor. Kemudian dilanjutkan pada model AR(1) sehingga jaringan yang terbentuk terdiri dari satu unit input dengan 301 data. Vektor inputnya dapat ditulis sebagai $X_1 = (x_1, x_2, \dots, x_{301})$ serta terdapat 301 neuron pada *pattern layer*.

Hasil prediksi baik training maupun cecking secara lengkap disajikan pada Tabel 1.

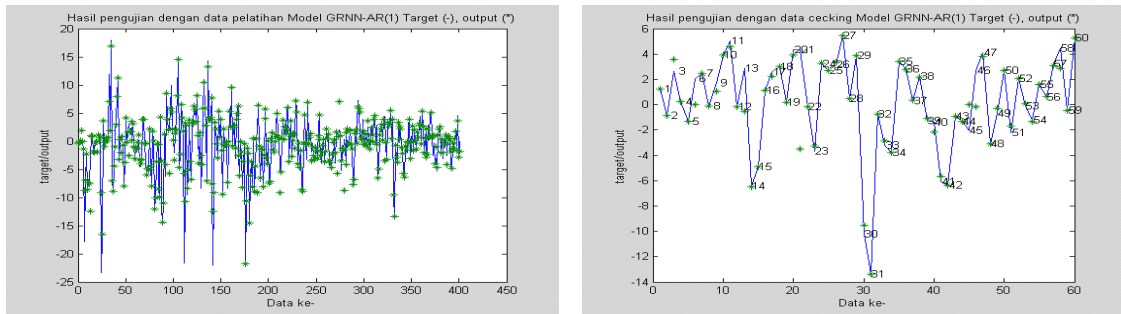
Tabel 1. Hasil penghitungan keakuratan prediksi nilai tukar mata uang yen Jepang terhadap dolar AS dengan model FFNN dan GRNN

Model	training				cecking			
	MSE	RMSE	MAE	R^2	MSE	RMSE	MAE	R^2
FFNN-AR(1)	0.0001	0.0070	0.0004	0.5128	34.4552	5.8699	0.7578	0.2509
FFNN-AR(2)	0.0175	0.1325	0.0076	0.4536	26.0772	5.1066	0.6593	0.4536
GRNN-AR(1)	0.0005	0.0226	0.0013	0.9213	3.4303	1.8521	0.2391	0.9405
GRNN-AR(2)	0.0045	0.0674	0.0039	0.9937	0.8448	0.9191	0.1187	0.9243

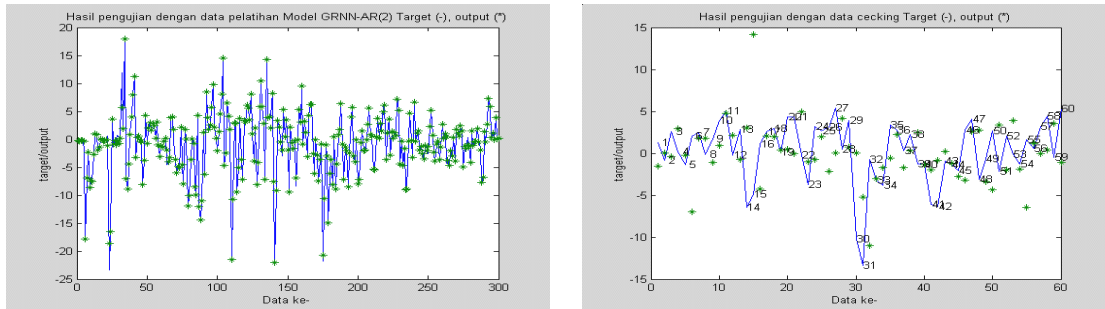
Dari Tabel 1 nampak bahwa pada kasus data nilai tukar mata uang yen Jepang terhadap dolar AS kedua model GRNN relatif lebih unggul daripada model FFNN berdasarkan pelatihan menggunakan data training dan data cecking.



Gambar 3. Plot hasil pengujian dengan data pelatihan dan data cecking model FFFN-AR(2)



Gambar 4. Plot hasil pengujian dengan data pelatihan dan data cecking model GRNN-AR(1)



Gambar 5. Plot hasil pengujian dengan data training dan data cecking model GRNN-AR(2)

Dari Gambar 3 sampai 5 menunjukkan prediksi berdasarkan data training dan data cecking model GRNN mempunyai presisi yang lebih baik dari model FFNN untuk data nilai tukar yen Jepang terhadap dolar AS. Untuk pengujian dengan data pelatihan bahkan model GRNN mampu menghasilkan prediksi yang hampir sempurna terutama untuk model GRNN-AR(2). Sedangkan model model GRNN-AR(1) unggul dalam prediksi dengan data cecking.

KESIMPULAN

Berdasarkan pembahasan dapat disimpulkan bahwa model GRNN mempunyai arsitektur yang cenderung lebih kompleks daripada model FFNN dengan jumlah unit pada layer tambahan yang relative besar. Namun demikian perkembangan komputasional tidak menjadikan hal ini masalah yang berarti. Pada aplikasi terhadap data nilai tukar yen Jepang terhadap dolar AS menunjukkan bahwa dengan input yang sama model GRNN relatif lebih unggul dalam ketepatan prediksi daripada model FFNN.

DAFTAR PUSTAKA

- [1]. Chatfield, C. and Faraway, J., Time Series Forecasting with Neural Networks: a Comparative Study Using the Airline Data, *Royal Statistical Society*, 47, Part 2, pp. 231-250, 1998
- [2]. Kao, J.J & Huang, S.S. 2000, Forecasts Using Neural Network versus Box-Jenkins Methodology for Ambient Air Quality Monitoring Data, *Journal of the Air and Waste Management Association*, 50, pp. 219-226, 2000.
- [3]. Leung, M.T., Chen, A.N., and Daouk, H., Forecasting Exchange Rates using General Regression Neural Networks, *Computers & Operations Research*, 27, pp. 1093-1110, 2000.
- [4]. Portugal, M.S., Neural Networks Versus Time Series Methods: a Forecasting Exercise, *14th International Symposium on Forecasting*, Stockholm School of Econometrics, Stockholm, Sweden, pp. 12-15 of June, 1995.