

Pemodelan Sistem dengan Metode Neural Network Back Propagation

Modeling System Using Neural Network Backpropagation

Dwi Sudarno Putra ¹⁾, Toto Sugiarto ¹⁾ & Meri Azmi ²⁾

¹⁾Jurusan Teknik Otomotif Universitas Negeri Padang Email: dwisudarnoputra@gmail.com, totosugiarto5526@yahoo.co.id

²⁾Jurusan Teknologi Informasi Politeknik Negeri Padang Kampus Limau Manis Padang Telp. 0751-72590 Fax. 0751-72576 Email: meriazmi@gmail.com

Abstract

A system can be modeled using mathematical formulation analysis method. Analysis of the system starts from the value of inputs, processes, noise, and output processes. then sought mathematical approach. This method is very complex, especially for some of the processes that have a higher order. In this article the author tries to discuss a method to perform modeling on a system by using Back Propagation Neural Network method. In the neural network method we require the initial data for network training process. Network training process is intended to gain weight. The weight is then identified with a model system. This paper has demonstrated that the modeling system can be done by the method of back propagation neural network

Keywords: modeling system, neural network application, back propagation.

PENDAHULUAN

Pemodelan sebuah sistem pada dasarnya adalah menggantikan proses sebuah sistem dengan sesuatu yang lain secara teoritis sehingga jika diberi masukan tertentu akan menghasilkan sebuah keluaran yang sesuai dengan sistem sebenarnya. (Benyamin 2009)

Pemodelan matematika menjadi salahsatu pendekatan yang sering digunakan dalam menganalisa proses sebuah sistem. Setelah model matematika diperoleh seorang analis dapat dengan mudah melakukan langkah-langkah strategis berikutnya berkenaan dengan sistem yang sedang dianalisanya. Pemodelan matematika memerlukan kemampuan analisis matematik dan penguasaan konsep sistem yang komprehensif dan mendalam. Jika terjadi salah persepsi konsep matematis dan konsep sistem maka hasil pemodelan juga tidak akan maksimal.

Back Propagation merupakan salah satu metode dari neural network (jaringan syaraf tiruan) yang banyak digunakan. Lihat gambar 1. Metode ini masuk dalam

kategori terbimbing. Metode ini menggunakan beberapa layer.

Seperti halnya model neural network lain, Backpropagation melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola yang digunakan selama pelatihan serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukan yang serupa (tapi tidak sama) dengan pola yang dipakai selama pelatihan.

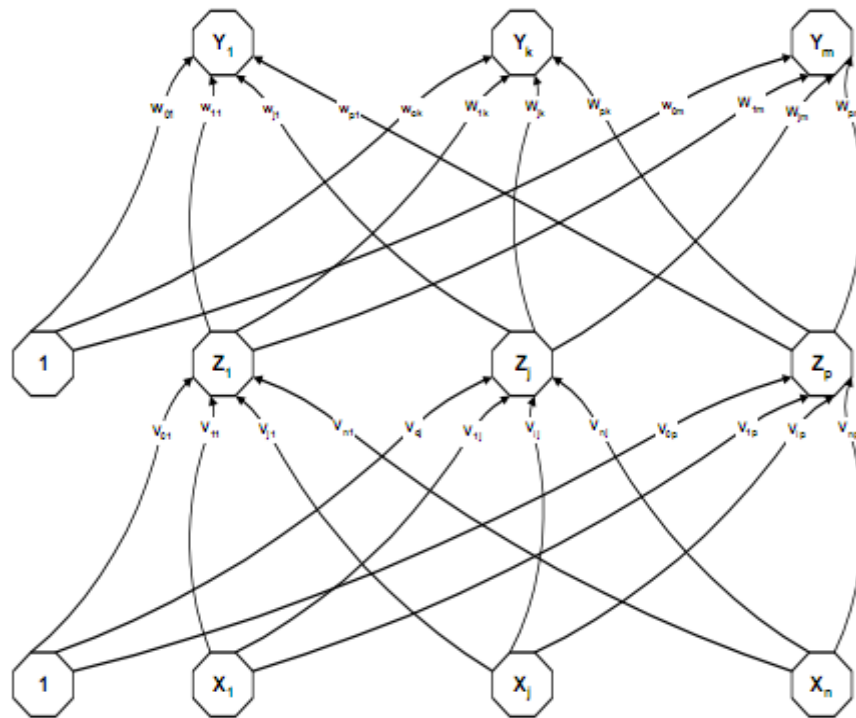
Kemampuan Neural Network Backpropagation dalam mengenali pola setelah proses pelatihan ini akan coba dijadikan dasar sebagai langkah pemodelan sebuah sistem. Lihat gambar 2, Parameter masukan dan keluaran proses plant digunakan untuk proses pelatihan. Untuk melakukan hal tersebut maka idealnya dibutuhkan data-data eksperimen yang diambil langsung dari plant.

Data yang akan digunakan untuk proses pelatihan tidak diambil langsung dari eksperimen terhadap plant, melainkan dengan melakukan simulasi pada model matematis dari plant. Masukan sistem

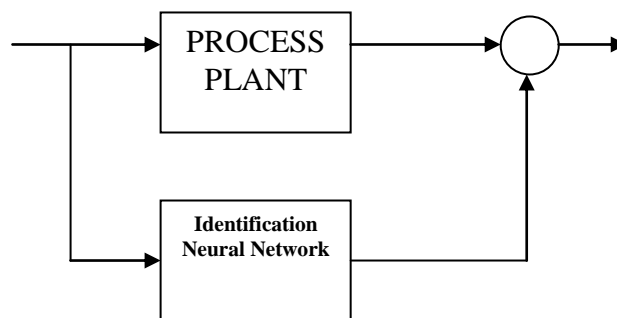
berupa bilangan acak sebanyak 100 buah dengan rentang nilai antara 1 s/d 100, Adapun persamaan model matematis dari

plant yang digunakan dalam tulisan kali ini seperti pada persamaan (1)

$$y(k) = \frac{1}{1+y(k-1)^2} + 0.25x(k) - 0.3x(k-1) \tag{1}$$



Gambar 1. Arsitektur Backpropagation (Benyamin 2009)



Gambar 2. Konsep Pemodelan Proses (Benyamin 2009)

METODOLOGI

Persiapan Data Training

Transformasi Z Score. Untuk memperoleh Data training yang ideal bagi proses backpropagation nilai

masukan perlu kita transformasi terlebih dahulu sedemikian rupa sehingga data masukan tersebut memiliki standar deviasi baru senilai 1. Untuk melakukan hal itu kita bisa gunakan transformasi Z-

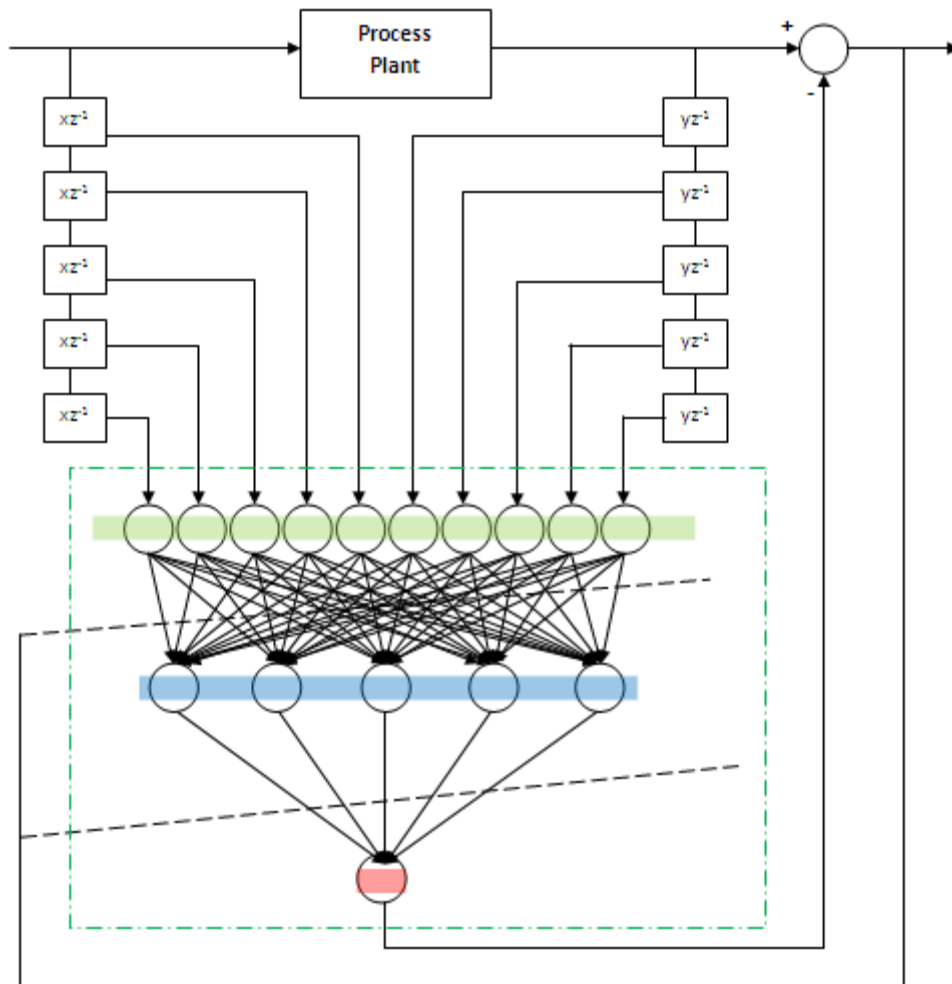
score. Rumusan transformasi ini adalah seperti pada persamaan (2)

$$Z = \frac{x-\mu}{\sigma} \quad (2)$$

Penghitungan nilai y(k).

$$y(k) = \frac{1}{1+y(k-1)^2} + 0.25xz(k) - 0.3xz(k-1) \quad (3)$$

Setelah didapatkan data-data x dalam kawasan Z-score (xz(k)) selanjutnya nilai xz(k) inilah yang kita operasikan ke persamaan y(k) menjadi persamaan (3)



- : input layer
- : hidden layer
- : output layer

Gambar 3. Ilustrasi Penyusunan data training

Setelah nilai y(k) kita dapatkan untuk masing-masing k, selanjutnya kita linearisasi nilai y(k) ini ke dalam range antara +1 dan -1. Hal ini dilakukan karena proses backpropagation akan bekerja optimal pada range +1 dan -1. Selanjutnya nilai hasil linearisasi kita notasikan sebagai yz(k). Sekali lagi, idealnya nilai y(k)

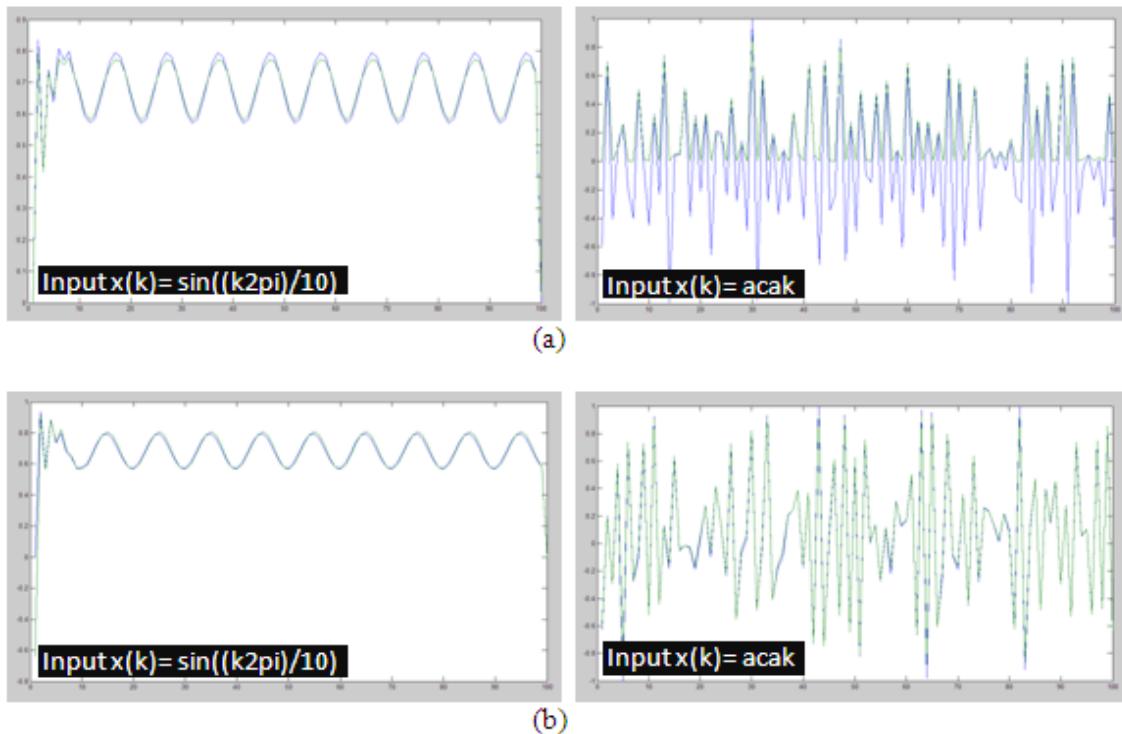
hendaknya diperoleh dengan eksperimen plant.

Penyusunan data training

Selanjutnya data yang sudah diperoleh yakni xz(k) dan yz(k) kita gunakan sebagai data training sesuai dengan konsep konfigurasi back propagation. Dalam tulisan

kali ini digunakan sebanyak 10 neuron masukan (5 neuron dari data masukan xz dan 5 neuron dari data keluaran yz). Kemudian sebuah layer pada hidden layer dengan 5 buah

neuron hidden layer. Dan untuk outputnya adalah sebuah neuron output yang nilainya akan dibandingkan dengan keluaran sistem guna mendapatkan eror.



Keterangan Gambar : garis biru (tegas) output sistem dan garis hijau (samar) output model

Gambar 4. Hasil pengujian aktivasi

- a) Dengan Fungsi Aktivasi Sigmoid Biner
- b) Dengan Fungsi Aktivasi Sigmoid Bipolar

Setelah didapatkan, eror selanjutnya digunakan pada proses backward untuk mengoreksi nilai-nilai dari bobot neural network. Untuk lebih jelasnya lihat Gambar 3.

Pelatihan Neural Network Backpropagation

Neural Network Backpropagation yang digunakan di tulisan ini menggunakan fungsi aktivasi berupa fungsi sigmoid bipolar, karena data yang akan diproses adalah data antara +1 dan -1.

$$f(x) = \frac{2}{1+e^{-x}} - 1 \quad (4)$$

dengan turunan

$$f'(x) = \frac{(1+f(x))(1-f(x))}{2} \quad (5)$$

Penggunaan Neural Network.

Algoritma Neural Network Backpropagation yang digunakan di tulisan ini sebelumnya telah berhasil diujicoba pada proses pengenalan pola data.

Proses Training Neural Network. Proses training data dilakukan dengan data acak seperti yang telah disinggung di atas. Seperti terlihat di gambar 3 output sistem juga digunakan sebagai masukan untuk unit input Neural Network nya. Hal ini dilakukan karena tujuan dari implementasi adalah untuk memperoleh

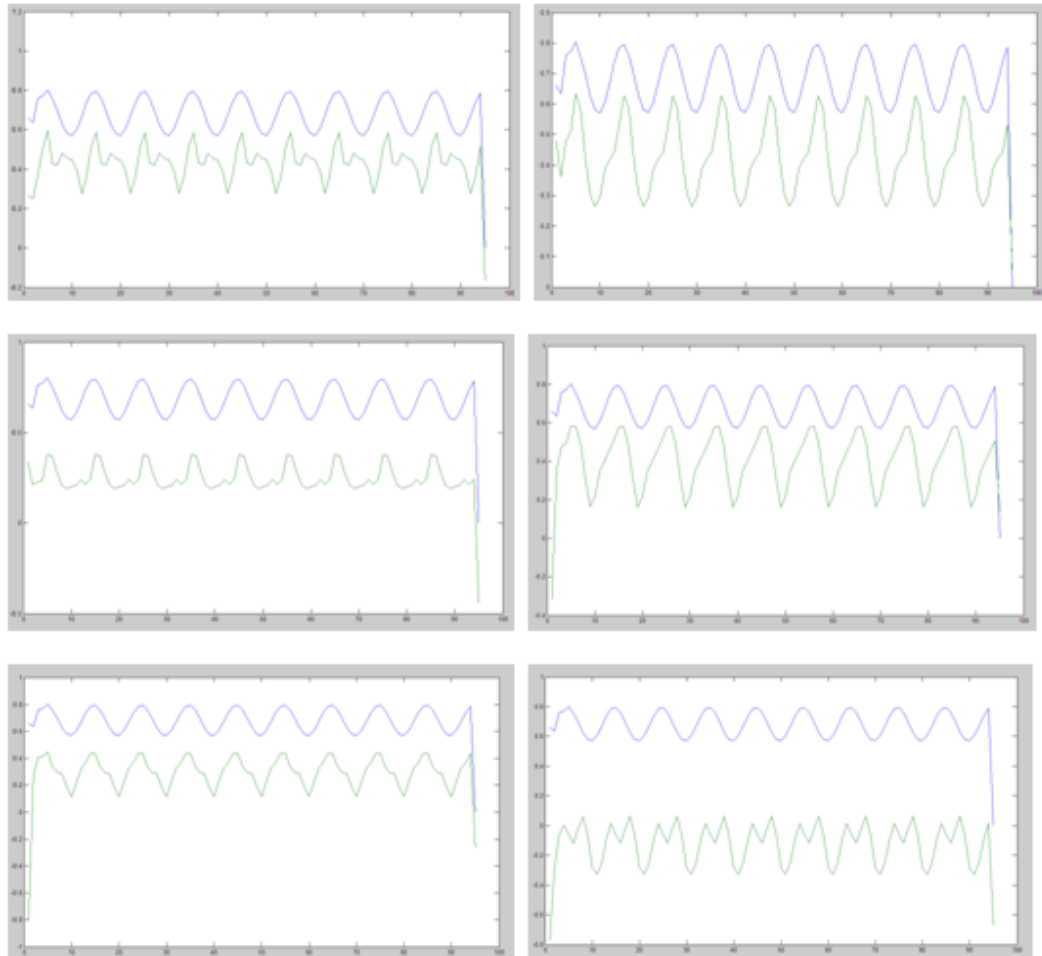
pemodelan sistem, sehingga selain memanfaatkan masukan sistem, keluaran sistem juga digunakan sebagai acuan.

Tujuan dari proses training adalah untuk mendapatkan bobot-bobot konvergen yang bisa mewakili karakteristik sistem.

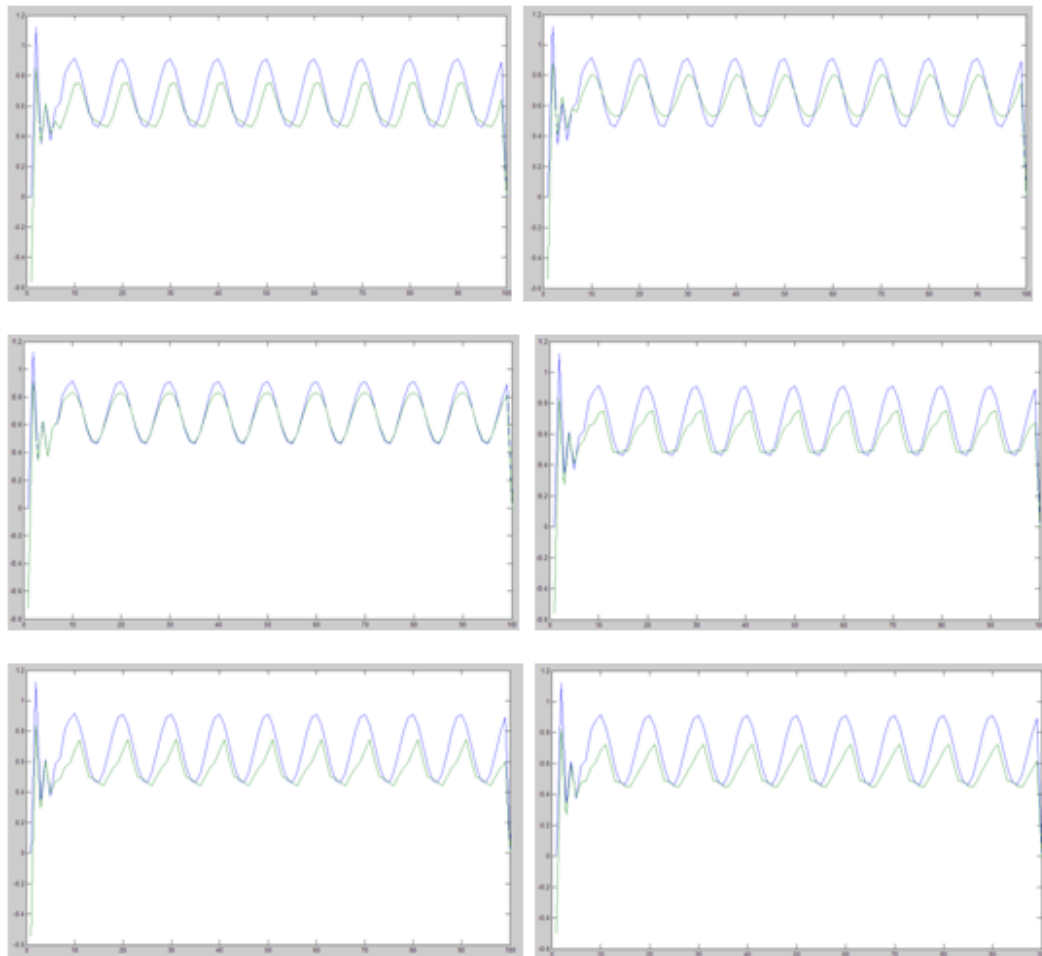
HASIL DAN PEMBAHASAN

Pengujian Fungsi Aktivasi

Pengaruh fungsi aktivasi tidak begitu terlihat jika masukan sinyal ada di range 0 s/d 1. Karena itu uji fungsi aktivasi



Gambar 5. Hasil uji Variasi I



Gambar 6. Hasil uji Variasi II

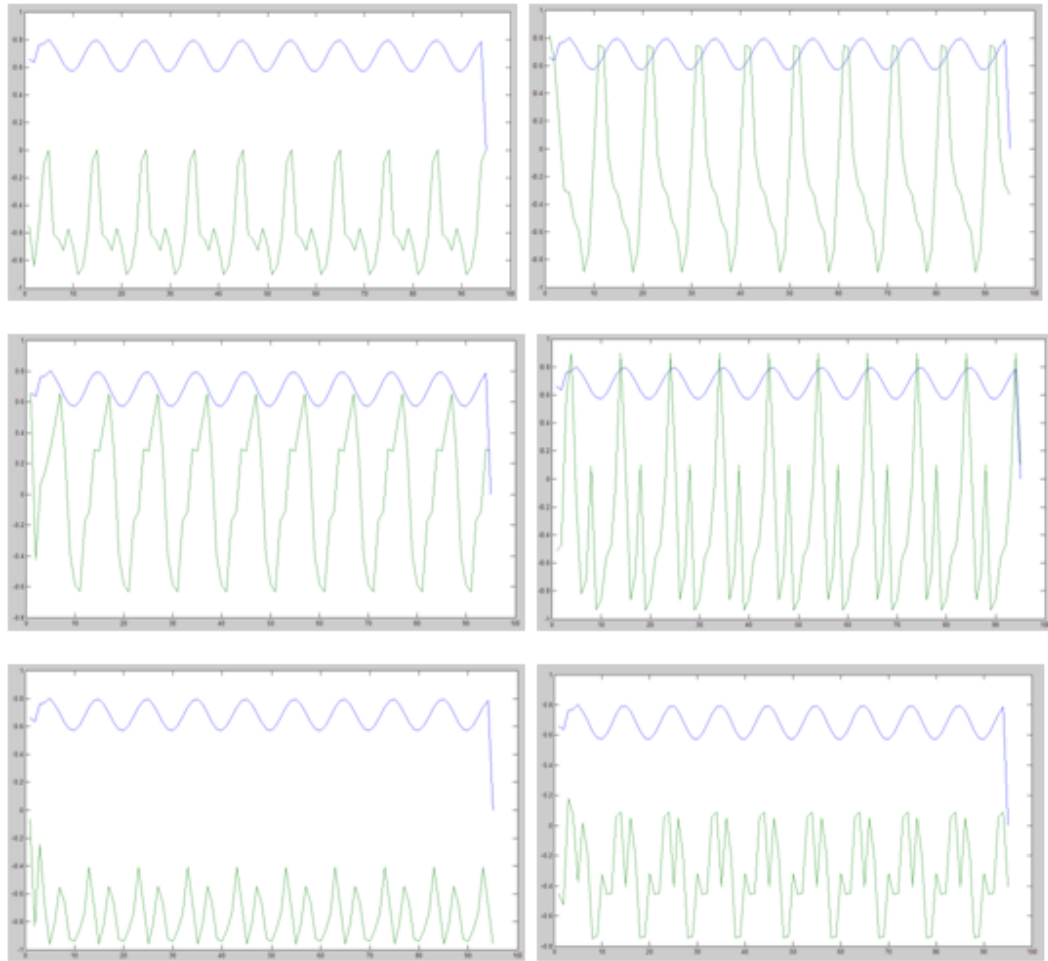
ini akan terlihat jelas bedanya pada saat kita melakukan pengujian dengan menggunakan fungsi masukan acak yang memiliki range -1 s/d $+1$. Lihat Gambar 4.

Dari hasil ujicoba tersebut jelas terlihat bila kita menggunakan fungsi aktivasi Sigmoid biner maka sinyal output model akan terpankaskan di batas bawah 0 dan batas atas 1 . Sedangkan saat kita gunakan fungsi aktivasi Sigmoid Bipolar kita akan mendapati bahwa output model bisa mencapai -1 s/d $+1$.

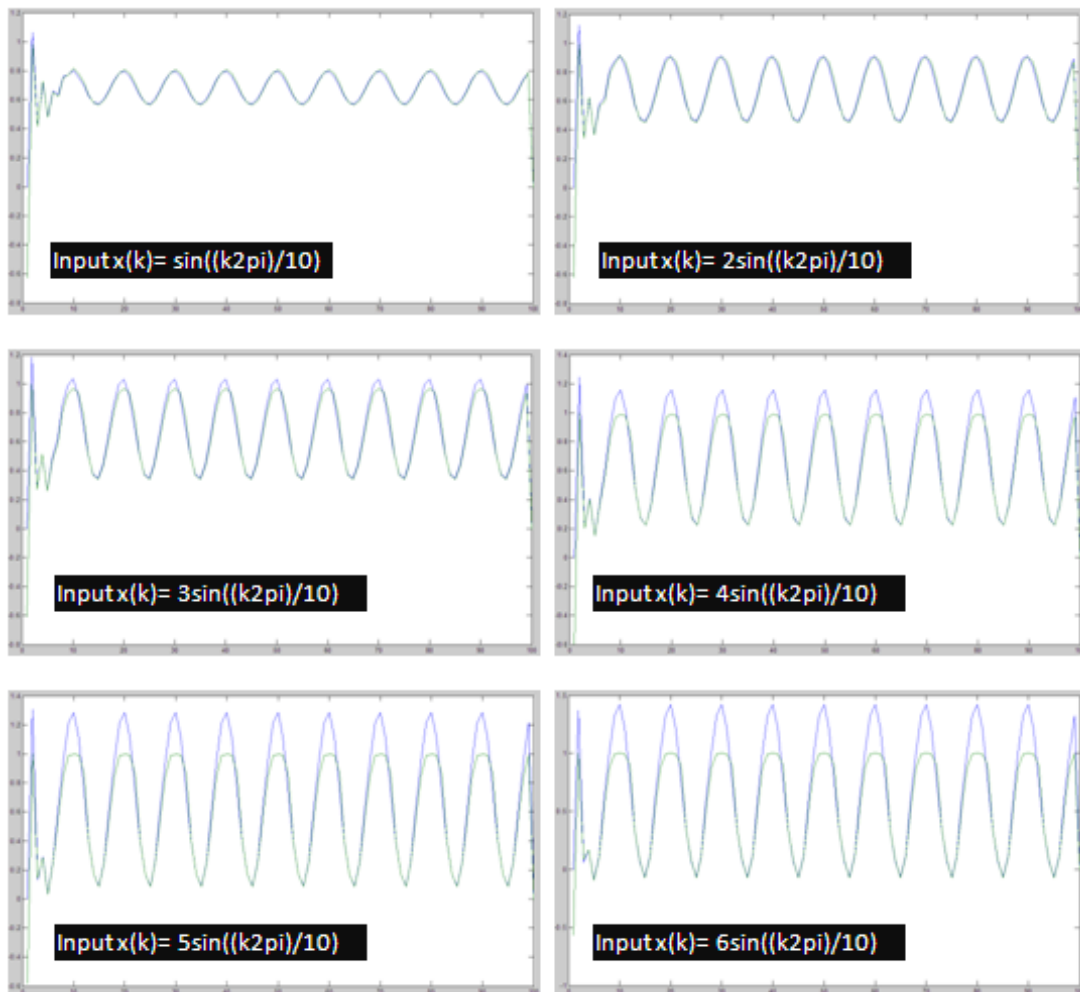
Dari kedua hal tersebut dapat diambil kesimpulan bahwa pemilihan fungsi aktivasi hendaknya disesuaikan dengan karakteristik sistem dan juga respon sistem yang kita harapkan.

Pengujian Pola Data Masukan

Tujuan dari pengujian ini adalah untuk mendapatkan pola masukan yang paling sesuai. Untuk itu pola masukan (di input layer) divariasikan. Pola masukan yang dimaksud di sini adalah nilai –nilai



Gambar 7. Hasil uji Variasi III



Gambar 8. Hasil pengujian dengan fungsi masukan yang bervariasi

yang diberikan pada neural network melalui input layernya.

Variasi I, dengan target $yz(k)$, input adalah data sekarang dan 4 data sebelumnya. Sehingga susunan data inputannya menjadi: $[xz(k) \ xz(k-1) \ xz(k-2) \ xz(k-3) \ xz(k-4) \ yz(k) \ yz(k-1) \ yz(k-2) \ yz(k-3) \ yz(k-4)]$. Gambar 5 menunjukkan hasil 6 kali simulasinya dengan data testing $x(k) = \sin((k2\pi)/10)$.

Variasi II, dengan target $yz(k)$ input adalah data sekarang yang disusun sejajar. Sehingga susunan data inputannya menjadi $[xz(k) \ xz(k) \ xz(k) \ xz(k) \ xz(k) \ yz(k) \ yz(k) \ yz(k) \ yz(k) \ yz(k)]$. Gambar 6 menunjukan hasil 6 kali simulasinya dengan data testing $x(k) = \sin((k2\pi)/10)$.

Variasi III, dengan target $yz(k)$ input adalah 5 data sebelumnya yang disusun sejajar. Sehingga susunan data inputannya menjadi $[xz(k-1) \ xz(k-2) \ xz(k-3) \ xz(k-4) \ xz(k-5) \ yz(k-1) \ yz(k-2) \ yz(k-3) \ yz(k-4) \ yz(k-5)]$. Gambar 7 menunjukkan hasil 6 kali simulasinya dengan data testing $x(k) = \sin((k2\pi)/10)$.

Dari ketiga masukan tersebut diperoleh kalau variasi ke II lah yang memiliki model respon yang relative sama/sesuai dengan target keluaran (output sistem). Seperti diketahui bahwa dalam hal ini output juga digunakan sebagai referensi masukan pada Neural Network sehingga respon Neural Network tentunya akan semakin baik

ketika semakin banyak referensi yang memuat parameter yang sama.

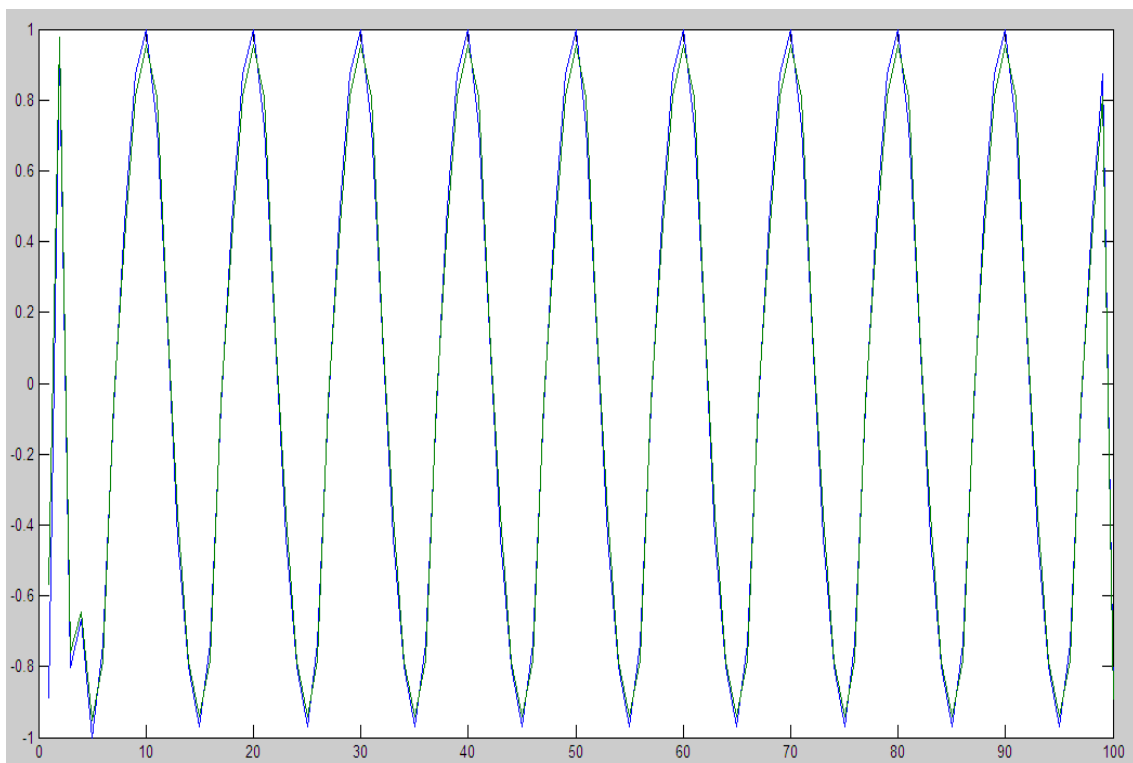
Pengujian Variasi Fungsi Masukan

Tujuan dari pengujian ini adalah untuk mengetahui parameter masukan yang ideal untuk sistem yang kita modelkan.

Dari ke enam hasil ujicoba pada gambar 8 dapat ditarik analisa bahwa ketika sinyal keluaran sistem yang masih dibawah level 1 maka output model masih bisa mengikutinya. Namun pada saat output sistem telah melebihi level 1 maka output model tidak mampu lagi

untuk mengikuti kendati secara umum bentuk output masih relative sama (fungsi sinus) namun jelas terlihat bahwa output model seakan terpankaskan di level 1.

Untuk mengatasi hal ini, yaitu agar output dari model bisa senantiasa mengikuti output sistem maka kita harus melakukan normalisasi terlebih dahulu pada output sistem sebelum kita masukkan ke neural network. Sebagai contoh, jika kita menormalisasi output sistem untuk input $6\sin((k2\pi)/10)$ maka akan didapatkan hasil simulasi seperti pada gambar 9.



Gambar 9. Hasil pengujian setelah adanya proses normalisasi

SIMPULAN

Neural Network Backpropagation terbukti dapat digunakan untuk memodelkan sebuah sistem. Untuk melakukan training, guna mendapatkan bobot yang mewakili model sistem, diperlukan data-data eksperimen dari sistem yang akan dimodelkan. Data eksperimen

tersebut berupa data masukan dan data keluaran dari sistem. Hasil output dari Neural Network kemudian dibandingkan dengan output sistem guna memperoleh nilai erornya, kemudian nilai eror inilah yang digunakan untuk mengkoreksi bobot neural network begitu seterusnya hingga kondisi konvergen tercapai.

SARAN

Penelitian yang telah dilakukan masih dapat dilanjutkan dengan variasi-variasi yang lain. Serta disarankan pula agar mencoba dengan metode neural network yang lain.

DAFTAR PUSTAKA

- Benyamin Kusumoputro,(2009), *Bahan Belajar Mahasiswa, Jaringan Neural Buatan*; Universitas Indonesia, Jakarta.
- Jong Jek Siang,(2009), *Jaringan Syaraf Tiruan & Pemrogramannya Menggunakan MATLAB*, Penerbit Andi, Yogyakarta.
- BrianR.Hunt,RonaldL.Lipsman and JonathanM.Rosenberg(2000), *A Guide to MATLAB For Beginners and Experienced Users*, Cambridge University Press, New York.