BAB III

JARINGAN SYARAF TIRUAN

3.1. Definisi Jaringan Syaraf Tiruan (JST)

Jaringan Syaraf Tiruan (JST) merupakan cabang ilmu multidisiplin yang relatif masih baru. Pada dasarnya, JST mencoba meniru cara kerja otak makhluk hidup, yaitu manusia. Salah satu struktur yang ingin ditiru adalah bentuk neuron (sel syaraf).

Secara harfiah, jaringan syaraf tiruan (JST) adalah salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia tersebut (Jang, 1997). Istilah buatan mengandung arti bahwa jaringan syaraf ini diimplementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran.

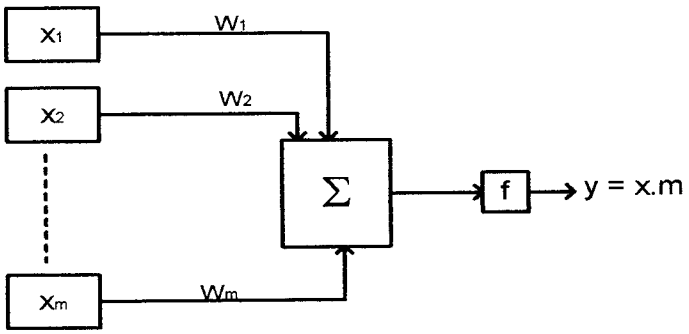
JST memiliki sifat mendasar yaitu kecerdasannya analog dengan kecerdasan manusia. JST dapat menyelesaikan persoalan yang sulit jika diselesaikan dengan menggunakan komputasi secara konvensional. Manusia memiliki kemampuan mengingat informasi pola secara menyeluruh dan mengadaptasi pemrosesan pola dengan baik. Jaringan syaraf tiruan dirancang dan dilatih untuk memiliki kemampuan seperti yang dimiliki manusia. Jadi, salah satu motivasi untuk mempelajari pola yang terkait dengan JST ini adalah untuk memahami kemampuan manusia dan diharapkan setelah dilatih maka kecerdasan dari jaringan syaraf tiruan dapat mempunyai kecerdasan yang dimiliki oleh manusia.

Model jaringan syaraf ditunjukkan dengan kemampuannya dalam analisa dan prediksi. Oleh karena itu, JST dapat digunakan untuk belajar dan menghasilkan operasi dari beberapa input untuk menghasilkan output tertentu dan juga dapat memprediksi kemungkinan output yang akan muncul jika diberikan sejumlah input kepadanya.

3.2. Model-Model Jaringan Syaraf Tiruan (JST)

Model JST adalah kondisi topologi koneksi-koneksi pada JST dan aturan-aturan tertentu yang ada pada JST. Pada dasarnya, model atau pola jaringan syaraf tiruan meniru jaringan syaraf manusia. (Fausett, 1986)

Gambar di bawah ini menunjukkan model sel syaraf (neuron) tiruan yang disederhanakan. Ini merupakan dasar dari jaringan syaraf tiruan.



Gambar 3.1 Model matematis jaringan syaraf tiruan

Pada gambar tersebut dijelaskan bahwa input pada jaringan tersebut adalah x_1, x_2, ..., x_m yang beranalogi dengan tingkat rangsangan yang datang dan kumpulan nilai bobot koneksi (*weight*) w_1, w_2, ..., w_m yang secara biologis memiliki analogi dengan kekuatan sinapsis (*synaptic strengths*) yang dimiliki neuron.

Pada dasarnya sebuah neuron mempunyai 2 buah hubungan, yaitu :

1. Hubungan *inhibitory*, yaitu hubungan yang dapat memberitahukan kepada neuron untuk tidak membangkitkan sinyal input,
2. Hubungan *excicatory*, yaitu hubungan yang dapat memberitahukan kepada neuron untuk membangkitkan sinyal input.

Berdasarkan model JST yang digambarkan di atas, maka JST dapat dinyatakan dalam hubungan matematis yang menyatakan perkalian antara input dan nilai bobot koneksi. Hasil perkalian tersebut dijumlahkan dan disimpan dalam neuron. Setelah itu, hasilnya

disubstitusikan dalam fungsi nonlinear *f*. Pada batas ambang (*threshold*) tertentu, fungsi *f* akan menghasilkan output dan apabila di bawah ambang batas maka fungsi *f* tidak akan menghasilkan output.

Apabila dinyatakan dalam notasi matematika maka persamaannnya adalah sebagai berikut : (Fausset, 1986)

$y = f(x_1 * w_1 + x_2 * w_2 + \ldots + x_m w_m)$

atau dalam notasi vektor :

$y = f(x*w)$

dengan :

x = vektor baris yang terdiri dari m elemen,

w = vektor kolom yang terdiri dari m elemen,

y = besaran skalar,

f = fungsi nonlinear.

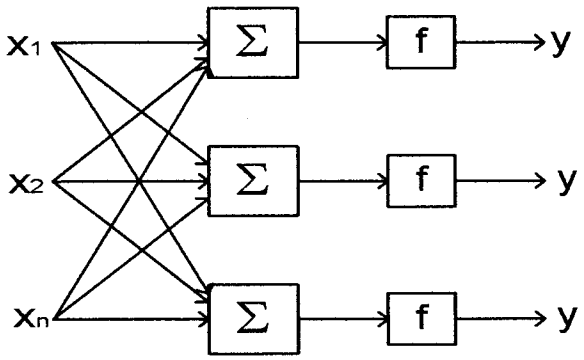
Fungsi nonlinear f dapat juga disebut dengan fungsi aktivasi.

Ada beberapa model yang digunakan dalam jaringan syaraf tiruan, yaitu : model JST satu lapisan, Model JST banyak lapisan, Model JST dua lapisan dengan umpan balik, dan model JST lapisan kompetitif.

3.2.1 Model JST satu lapisan

Model ini pertama kali diperkenalkan oleh Widrow dan Hoff (Fausset,1986). Penggunaan Model JST satu lapisan sangat terbatas, tetapi konsep dan gagasannya banyak dipakai oleh para pakar untuk membangun model JST lainnya, khususnya model JST banyak lapisan.

Pada model JST satu lapisan, sel-sel syaraf tiruan digabungkan menjadi satu lapisan. Hal ini dapat dilihat pada gambar di bawah ini.



Gambar 3.2 Model jaringan syaraf tiruan satu lapisan

Pada gambar di atas, kita dapat melihat bahwa sel-sel syaraf tiruan digabungkan menjadi satu lapisan. Semua input dihubungkan dengan kumpulan sel syaraf dan outputnya dapat dihitung dengan persamaan atau model matematika seperti di bawah ini : (Fausset, 1986)

$y_1 = f(x_1*w_{11} + x_2*w_{21} + \ldots + x_m*w_{m1})$

$y_2 = f(x_1*w_{12} + x_2*w_{22} + \ldots + x_m*w_{m2})$

$y_3 = f(x_1*w_{13} + x_2*w_{23} + \ldots + x_m*w_{m3})$

.....

.....

$y_n = f(x_1*w_{1n} + x_2*w_{2n} + \ldots + x_m*w_{mn})$

atau dengan notasi vektor :

$y = f(x*w)$

dengan :

x = vektor baris yang terdiri dari m elemen,

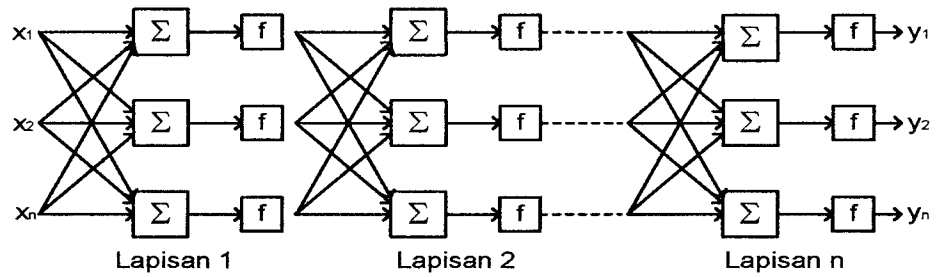
w = vektor kolom yang terdiri dari m elemen,

y = vektor kolom yang terdiri dari n elemen,

f = fungsi nonlinear atau biasa disebut sebagai fungsi aktivasi.

3.2.2 Model JST banyak lapisan

Pada dasarnya, model JST banyak lapisan merupakan gabungan dari beberapa model JST satu lapisan yang disusun berurutan. Prinsip kerja model JST ini sama dengan model JST satu lapisan. Output dari tiap lapisan sebelumnya merupakan input untuk lapisan berikutnya. Bentuk dari model JST banyak lapisan dapat dilihat pada gambar di bawah ini.



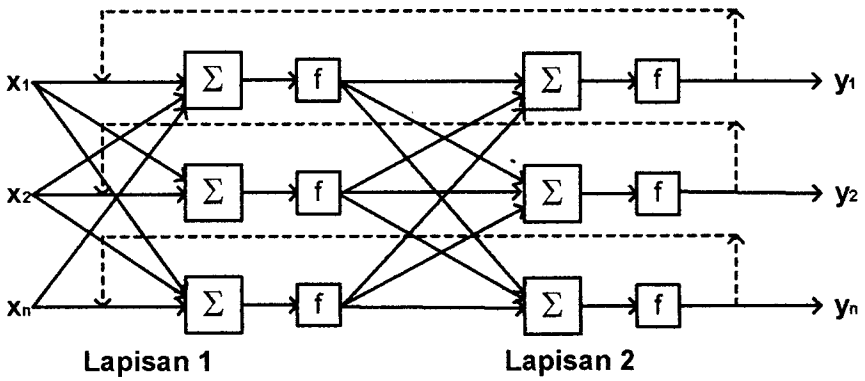
Gambar 3.3 Model jaringan syaraf tiruan banyak lapisan

3.2.3 Model JST dua lapisan dengan umpan balik

Model JST dengan umpan balik pertama kali diperkenalkan oleh John Hopfield dari California Institute of Technology pada tahun 1982. Hopfield berpendapat bahwa kumpulan neuron tiruan dalam jumlah yang sangat banyak dapat melakukan tugas-tugas tertentu.

Dengan jumlah neuron yang sangat banyak, JST akan mempunyai sifat *fault tolerance*. Sifat ini mengandung maksud bahwa sebagian dari sel-sel dalam jaringan syaraf tidak akan mempengaruhi output yang dikeluarkan atau dengan kata lain jika ada kerusakan sejumlah kecil sel-sel dalam JST, maka output akan tetap keluar sebagaimana mestinya.

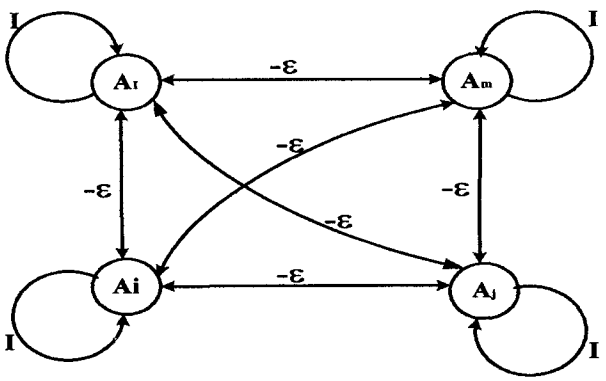
Model JST dua lapisan dengan umpan balik mengandung arti bahwa output yang dihasilkan akan mempengaruhi input. Dengan kata lain umpan balik berfungsi untuk memperbaiki *error* pada output. Model JST dua lapisan dengan umpan balik dapat dilihat pada gambar di bawah ini.



Gambar 3.4 Model jaringan syaraf tiruan dua lapisan dengan umpan balik

3.2.4 Model JST lapisan kompetitif

Model JST lapisan kompetitif merupakan bagian jumlah yang besar pada jaringan syaraf. Pada dasarnya, hubungan antara satu neuron dengan neuron lainnya pada lapisan kompetitif , arsitekturnya tidak dapat ditunjukkan pada beberapa jaringan syaraf. Contoh dari model JST lapisan kompetitif dapat dilihat pada gambar di bawah ini.



Gambar 3.5 Model jaringan syaraf tiruan lapisan kompetitif

Pada gambar tersebut, kita dapat melihat bahwa koneksi dari lapisan tersebut mempunyai bobot –e.

3.3. Prinsip Dasar Pelatihan JST

Suatu karakteristik dari jaringan syaraf tiruan adalah kemampuannya untuk belajar. Cara belajar dari latihan yang diberikan pada JST menunjukkan kesamaan dengan

perkembangan intelektual manusia. Akan tetapi, kemampuan belajar jaringan syaraf tiruan bersifat terbatas sehingga jaringan ini tidak dapat melakukan segalanya (Fausset, 1986).

Pada dasarnya, proses pelatihan dan pembelajaran pada JST diklasifikasikan menjadi 2 (Kosko, 1990), yaitu bagaimana JST ini menyimpan pengetahuan (*encode*) dan bagaimana JST ini menanggapi dan memproses data yang masuk (*decode*).

Berdasarkan encode, proses pelatihan dan pembelajaran pada JST diklasifikasikan menjadi 2 bagian, yaitu :

1. *Supervised,* yaitu jaringan diberi masukan tertentu dan keluarannya ditentukan oleh algoritma yang kita buat atau dengan kata lain proses belajarnya harus dibimbing. Pada proses belajar tersebut, jaringan syaraf tiruan akan menyesuaikan bobot sinaptiknya,
2. *Unsupervised,* yaitu jaringan syaraf tiruan secara mandiri akan mengatur keluarannya sesuai dengan aturan yang dimilikinya.

Berdasarkan decode, proses pelatihan dan pembelajaran pada JST diklasifikasikan menjadi 2 bagian, yaitu :

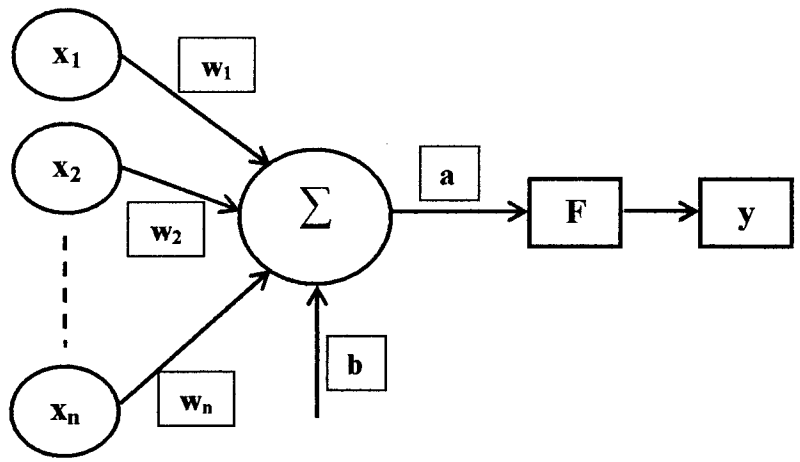
1. *feedforward* atau lurus, yaitu jika hasil outputnya sudah dapat diketahui sebelumnya tanpa adanya umpan balik,
2. *feedback* atau umpan balik, yaitu jika pada hasil outputnya dilakukan umpan balik.

Pada JST, neuron-neuron dikumpulkan dalam lapisan-lapisan (*layer*) yang disebut dengan lapisan neuron (*neuron layers*). Biasanya neuron-neuron pada satu lapisan akan dihubungkan dengan lapisan-lapisan sebelum dan sesudahnya (kecuali lapisan input dan lapisan output). Informasi yang diberikan pada jaringan syaraf akan dirambatkan mulai dari lapisan input sampai ke lapisan output melalui lapisan lainnya, yang sering disebut dengan lapisan tersembunyi (*hidden layer*). Tergantung algoritma pembelajarannya, bisa jadi informasi tersebut akan dirambatkan secara mundur pada jaringan.

3.4. Fungsi Aktivasi pada JST

Input JST akan diproses oleh suatu fungsi perambatan yang akan menjumlahkan nilai-nilai semua bobot yang datang. Hasil penjumlahan ini kemudian akan menjumlahkan nilai-nilai semua bobot yang akan datang. Hasil penjumlahan ini kemudian akan dibandingkan dengan suatu nilai ambang (*threshold*) tertentu melalui fungsi aktivasi setiap neuron. Apabila input tersebut melewati suatu nilai ambang tertentu, maka neuron tersebut akan diaktifkan. Akan tetapi, jika tidak maka neuron tersebut tidak akan diaktifkan. Apabila neuron tersebut diaktifkan, maka neuron tersebut akan mengirimkan output melalui bobot-bobot outputnya ke semua neuron yang berhubungan dengannnya.

Gambar di bawah ini menunjukkan jaringan syaraf tiruan sederhana dengan fungsi aktivasi F.



Gambar 3.6 Jaringan syaraf sederhana dengan fungsi aktivasi F

Pada gambar di atas, neuron akan mengolah m input (x_1, x_2, ..., x_m) yang masing-masing memiliki bobot w_1, w_2, ..., w_m dan bobot bias b, dengan rumus :

$$a = \sum_{i=1}^{m} x_i . w_i$$

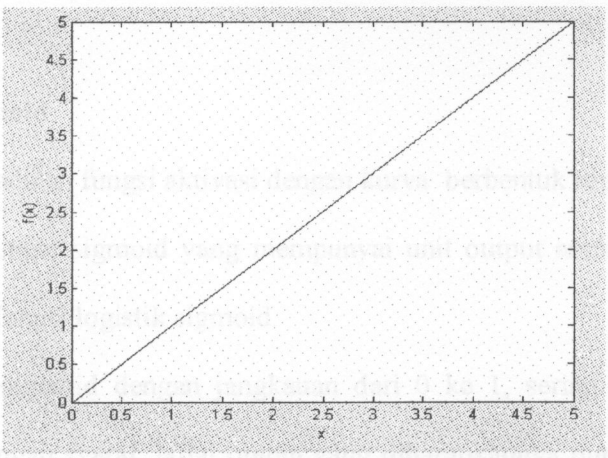
Setelah itu, fungsi aktivasi F akan mengaktivasi a menjadi output jaringan y.

Ada beberapa fungsi aktivasi yang sering digunakan dalam JST, yaitu fungsi identitas, fungsi biner dengan threshold θ, fungsi biner sigmoid, dan fungsi bipolar sigmoid.

3.4.1 Fungsi identitas (linear)

Fungsi identitas atau linear memiliki nilai output yang sama dengan nilai inputnya. Fungsi ini sering digunakan untuk unit input. Fungsi ini disebut juga *purelin*. Adapun persamaan dari fungsi identitas adalah sebagai berikut :

$y = F(x) = x$, untuk semua nilai x



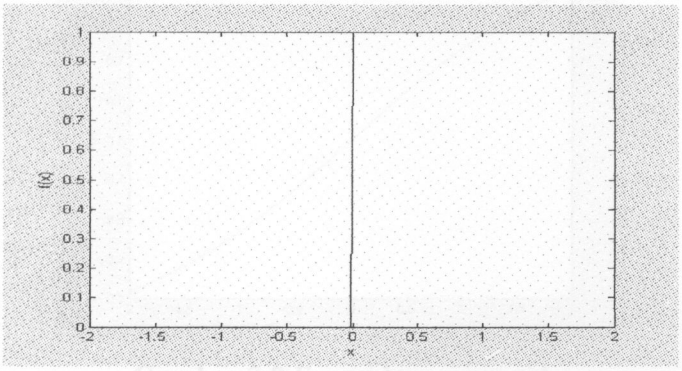
Gambar 3.7 Fungsi linear/identitas

Fungsi ini biasa dipakai pada lapisan tertentu pada model JST banyak lapisan

3.4.2 Fungsi biner dengan threshold θ

JST satu lapisan sering menggunakan fungsi ini untuk mengubah input yang masuk ke dalam jaringan, dimana nilai variabel terhadap sebuah unit output berupa angka biner (1 atau 0) atau tanda bipolar (1 atau -1). Persamaan dari fungsi biner dengan threshold adalah sebagai berikut :

$$f(x) = \begin{cases} 1 & \text{jika } x \geq \theta \\ 0 & \text{jika } x < \theta \end{cases}$$



Gambar 3.8 Fungsi biner

Fungsi ini sering digunakan pada model JST satu lapisan karena nilai output yang dihasilkannya dapat mengaktivasi model matematik untuk JST satu lapisan dengan baik.

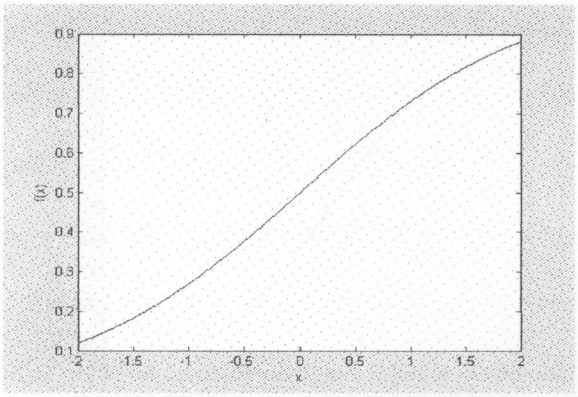
3.4.3 Fungsi biner sigmoid

Fungsi sigmoid adalah fungsi aktivasi dengan kurva berbentuk seperti huruf s. Fungsi biner sigmoid adalah fungsi sigmoid yang mempunyai unit output angka biner (1 atau 0). Fungsi ini disebut juga fungsi logistik sigmoid.

Fungsi logistik sigmoid dengan jangkauan dari 0 ke 1, sering digunakan sebagai fungsi aktivasi dari jaringan syaraf. Nilai output yang diharapkannya pun bernilai biner dan nilai tersebut berada di antara 0 dan 1. Persamaan dari fungsi biner sigmoid atau logistik sigmoid adalah sebagai berikut : (Fausett, 1986)

$$f(x) = \frac{1}{1+\exp(-\sigma x)}$$

$$f'(x) = \sigma f(x)[1-f(x)]$$



Gambar 3.9 Fungsi biner sigmoid

Fungsi ini secara khusus sangat baik jika kita menggunakan JST yang dilatih dengan algoritma *backpropagation*, karena hubungan yang sederhana di antara nilai fungsi pada suatu titik dan nilai turunan pada titik itu mengurangi beban perhitungan selama pelatihan.

3.4.4 Fungsi bipolar sigmoid

Fungsi bipolar sigmoid mempunyai jangkauan paling umum dari -1 sampai 1. Persamaan dari fungsi bipolar sigmoid adalah sebagai berikut : (Fausett, 1986)

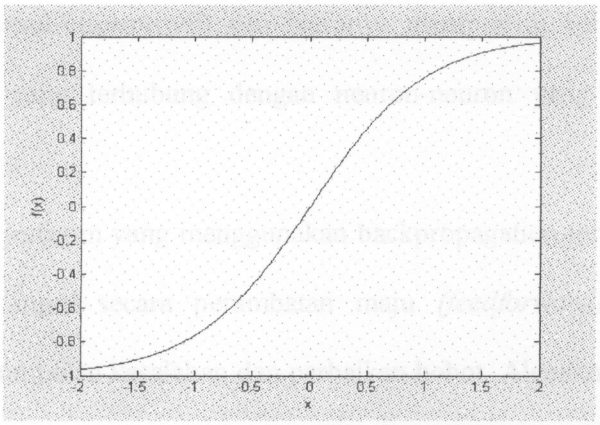
$$f(x) = \frac{1 - e^{-\sigma x}}{1 + e^{-\sigma x}}$$

dengan : $f'(x) = \frac{\sigma}{2}[1 + f(x)][1 - f(x)]$

Fungsi ini sangat dekat dengan fungsi tangen hiperbolik yang sama-sama mempunyai jangkauan antara -1 sampai 1. Fungsi tangen hiperbolik disebut juga *tansig*. Fungsi tangen hiperbolik dapat dirumuskan sebagai : (Fausett, 1986)

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

dengan : $f'(x) = [1 + f(x)][1 - f(x)]$



Gambar 3.10 Fungsi bipolar sigmoid

Penggunaan fungsi ini hampir sama dengan fungsi biner sigmoid (logsig). Yang membedakannya adalah nilai jangkauan output yang dihasilkanya, yaitu terletak antara -1 dan 1

3.5. Jaringan Syaraf Tiruan Backprogation (Jalar Balik)

Pada jaringan syaraf tiruan, kita mengenal banyak konfigurasi neuron dan algoritma pelatihan/pembelajaran. Saat ini, banyak sekali nama model algoritma pembelajaran pada JST, antara lain :

- *Perceptron*,
- *Backpropagation* (*Jalar balik*),
- *Hopfield*,
- *Linear filters* (*filter linear*)
- *Self organizing*,
- *Learning Vector Quantification (LVQ)*,
- dan masih banyak lagi.

Pada tugas akhir ini, penulis akan membangun jaringan syaraf tiruan dengan algoritma pembelajaran Backpropagation (Jalar Balik). Backpropragation adalah algoritma

pembelajaran yang terawasi (*supervised*) dan biasanya mempunyai banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyinya.

Pelatihan sebuah jaringan yang menggunakan backpropagation terdiri dari 2 langkah, yaitu : pelatihan pola input secara perambatan maju *(feedforward)*, dan jalar balik (backpropagation) dari kumpulan kesalahan dan perbaikan bobot. Algoritma backpropagation menggunakan error output untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*). Untuk mendapatkan error ini, tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu.

Pada saat perambatan maju, neuron-neuron diaktifkan dengan menggunakan fungsi aktivasi yang dapat didiferensialkan, seperti :

1. Sigmoid

   $$y = f(x) = \frac{1}{1+e^{-\sigma x}}$$

   dengan : $f'(x) = \sigma f(x)[1-f(x)]$,

   Fungsi ini memiliki nilai pada range 0 sampai 1. Oleh karena itu , fungsi ini sering digunakan untuk jaringan syaraf yang membutuhkan nilai output yang terletak pada interval $0 \leq$ nilai output ≤ 1.

2. Tansig

   $$y = f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

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

   dengan : $f'(x) = [1+f(x)][1-f(x)]$,

Oleh karena fungsi ini memiliki nilai pada range -1 sampai 1, maka fungsi ini sangat cocok digunakan untuk jaringan syaraf tiruan yang mempunyai nilai output pada range -1 ≤ nilai output ≤ 1.

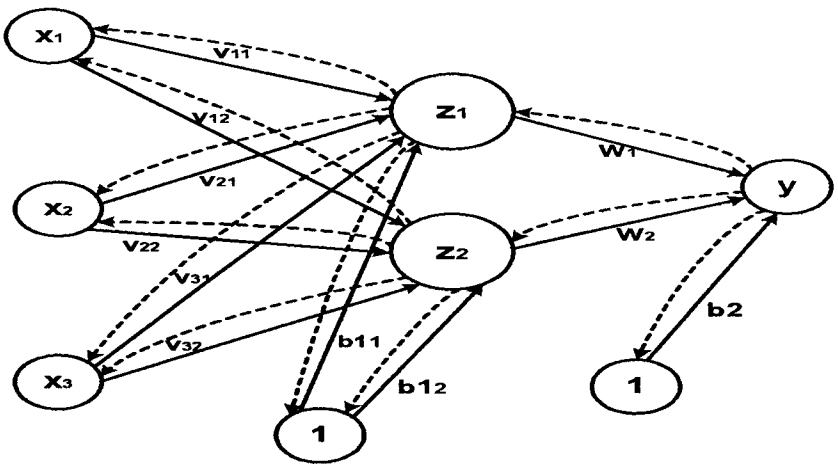
3. Purelin

$y = f(x) = x$

dengan : $f'(x) = 1$

Fungsi ini mempunyai nilai input sama dengan nilai outputnya dan biasa dipakai pada model JST banyak lapisan. Oleh karena nilai inputnya sama dengan nolai outputnya, maka fungsi ini dapat digunakan pada JST filter linear.

Contoh dari arsitektur jaringan backpropagation dapat kita lihat pada gambar berikut ini.



Gambar 3.11 Contoh arsitektur JST backpropagation (Fausett, 1986)

Pada gambar di atas, jaringan terdiri atas 3 unit neuron pada lapisan input, yaitu : x_1, x_2, dan x_3; 1 lapisan tersembunyi dengan 2 neuron, yaitu z_1 dan z_2; serta 1 neuron pada lapisan output, yaitu y. Bobot yang menghubungkan x_1, x_2, dan x_3 dengan z_1 pada lapisan tersembunyi adalah v_{11}, v_{21}, dan v_{31} dan bobot yang menghubungkan x_1, x_2, dan x_3 dengan z_2 pada lapisan tersembunyi adalah v_{12}, v_{22}, dan v_{32}. Bobot yang menghubungkan z_1 dan z_2

neuron pada lapisan output adalah w_1 dan w_2. Sementara itu, $b1_1$ dan $b1_2$ adalah bobot bias yang menuju ke neuron pertama dan kedua pada lapisan tersembunyi dan bobot bias b2 menghubungkan lapisan tersembunyi dengan lapisan output. Fungsi aktivasi akan mengaktivasi sinyal output tiap lapisan menjadi output lapisan tersebut.

3.6. Algoritma backpropagation

Seperti telah dijelaskan pada bagian sebelumnya bahwa pelatihan JST dengan menggunakan algoritma backpropagation terdiri atas 2 langkah, yaitu :

1. Pelatihan input secara feedforward dan perhitungannya,
2. Backpropagation (jalar balik) dari kumpulan kesalahan dan perbaikan bobot dan biasnya,

Untuk lengkapnya, algoritma backpropagation dapat kita lihat seperti di bawah ini.

Algoritma backpropagation :

- Inisialisasi bobot (ambil bobot awal dengan nilai random yang cukup kecil),
- Tetapkan : Maksimum Epoh, *Target Error*, dan *Learning Rate*(α),
- Inisialisasi : Epoh = 0, MSE = 1,
- Kerjakan langkah-langkah berikut selama (Epoh < Maksimum Epoh) dan (MSE > Target Error) :

1. Epoh = Epoh + 1,
2. Untuk tiap-tiap pasangan elemen yang akan dilakukan pembelajaran, kerjakan :

Feedforward :

a. Tiap-tiap unit input (x_i, i = 1, 2, 3, ...,n) menerima sinyal x_i dan meneruskan sinyal tersebut ke semua unit pada lapisan yang ada di atasnya (lapisan tersembunyi).

b. Tiap-tiap unit pada suatu lapisan tersembunyi (z_j, j =1, 2, 3, ...,p) menjumlahkan sinyal-sinyal input terbobot :

$$z_in_j = b_{1j} + \sum_{i=1}^{n} x_i . v_{ij}$$

gunakan fungsi aktivasi untuk menghitung sinyal outputnya :

$z_j = f(z_in_j)$

dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit output). Langkah (b) ini dilakukan sebanyak jumlah lapisan tersembunyi.

c. Tiap-tiap unit output (y_k, k = 1, 2, 3,...,m) menjumlahkan sinyal-sinyal berbobot :

$$y_in_k = b_{2k} + \sum_{i=1}^{p} z_i . w_{jk}$$

gunakan fungsi aktivasi untuk menghitung sinyal outputnya :

$y_k = f(y_in_k)$

dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit output).

Backpropagation

a. Tiap-tiap unit output (y_k, k = 1,2,3, ...,m) menerima target pola yang berhubungan dengan pola input pembelajaran, hitung informasi errornya :

$\delta_{2k} = (t_k - y_k)\, f'(y_in_k)$

$\varphi_{2jk} = \delta_k . z_j$

$\beta_{2k} = \delta_k$

kemudian hitung koreksi bobot (yang digunakan untuk memperbaiki nilai w_{jk}) :

$\Delta w_{jk} = \alpha . \varphi_{2jk}$

dan akhirnya menghitung koreksi bias (yang digunakan untuk memperbaiki b_{2k}) :

$\Delta b_{2k} = \alpha . \beta_{2k}$

langkah (a) ini dilakukan sebanyak jumlah lapisan tersembunyi.

b. Tiap-tiap unit lapisan tersembunyi (z_j, j = 1,2,3,...,p) menjumlahkan delta inputnya :

$$\delta_ in_j = \sum_{k=1}^{m} \delta_{2k} . W_{jk}$$

kalikan nilai ini dengan turunan fungsi aktivasi untuk menghitung informasi kesalahan,

$\delta_{1j} = \delta_ in_j . f'(z_in_j)$

kemudian hitunglah koreksi bobot (yang digunakan untuk koreksi v_{ij}) :

$\Delta v_{jk} = \alpha . \varphi_{1ij}$

Hitung juga koreksi bias (yang digunakan untuk koreksi b_{1j}) :

$\Delta b_{1j} = \alpha . \beta_{1j}$

d. Masing-masing unit output (y_k, k = 1,2,3,...,m) memperbaiki bobot dan biasnya (j = 0,1,2,...,p) :

$w_{jk}(baru) = w_{jk}(lama) + \Delta w_{jk}$

$b_{2k}(baru) = b_{2k}(lama) + \Delta b_{2k}$

e. Masing-masing unit tersembunyi (z_j, j = 1,2,3,...,p) memperbaiki bobot dan biasnya (I = 0,1,2,...,n) :

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

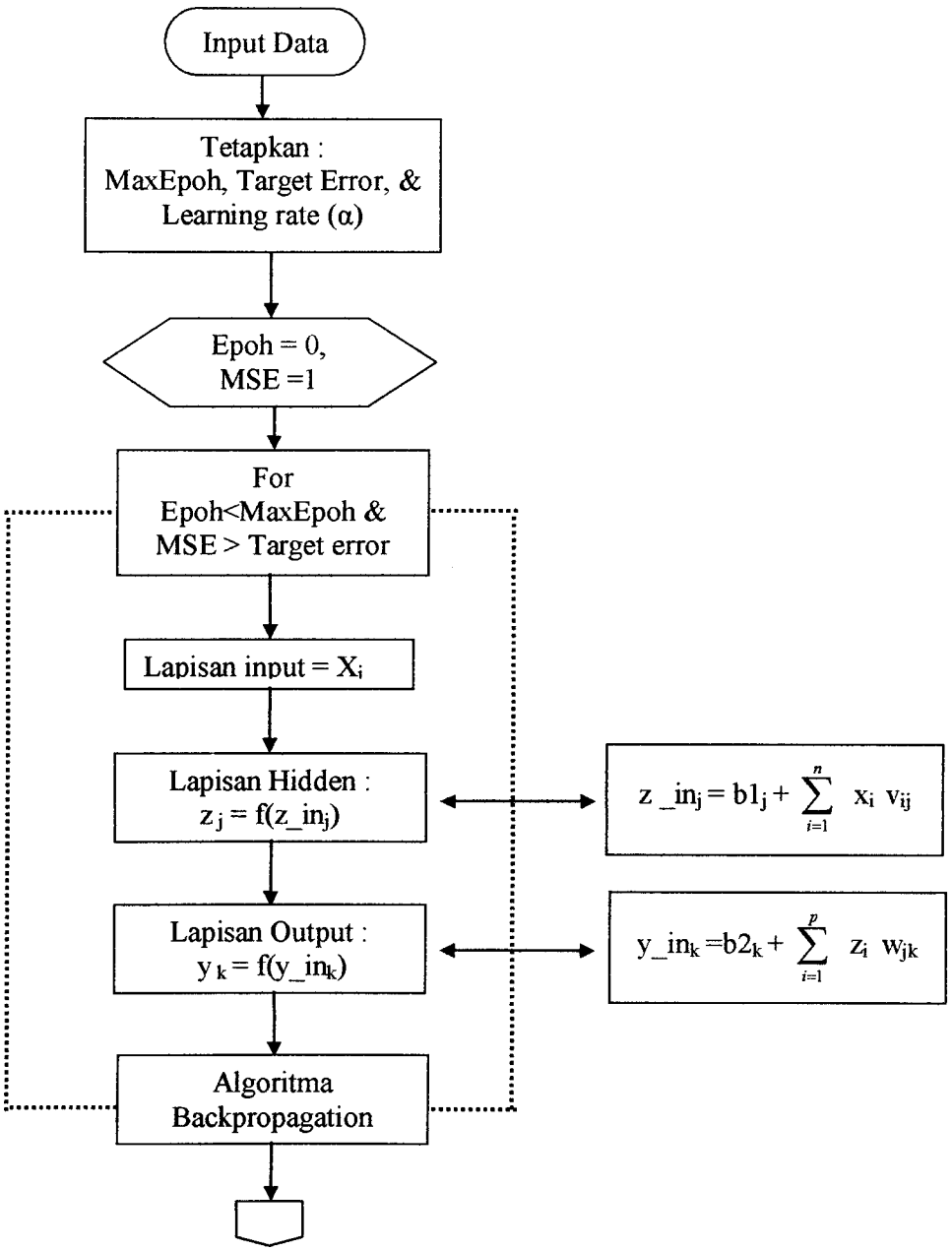
$b_{1j}(baru) = b_{1j}(lama) + \Delta b_{1j}$

f. Hitung MSE dan tes kondisi berhenti.

3.7. Diagram alir algoritma backpropagation

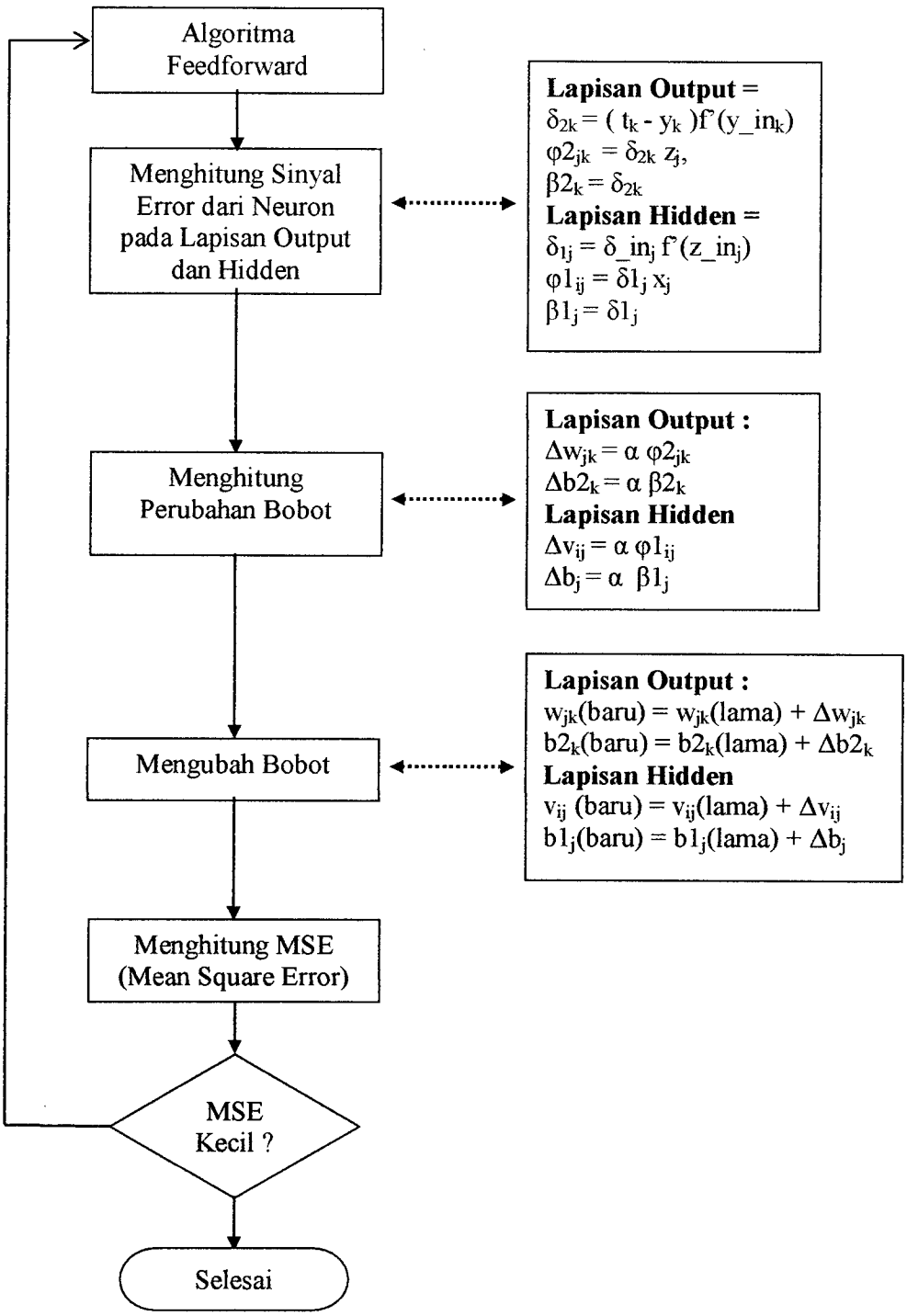
Berdasarkan algoritma di atas, maka kita dapat membuat diagram alir (*flow chart*) umum untuk algoritma backpropagation, yaitu diagram alir untuk algoritma feedforward dan diagram alir untuk algoritma backpropagation. Berikut ini adalah diagram alir dari algoritma yang dimaksud.

Diagram Alir untuk Algoritma Feedforward



Gambar 3.12 Diagram alir algoritma feedforward

Diagram Alir untuk Algoritma Backpropagation



Gambar 3.13 Diagram alir algoritma backpropagation