

SISTEM KOMPUTASI JARINGAN SYARAF TIRUAN

BACKPROPAGATION

Subiyanto

Abstrak

Tulisan ini menyajikan model JST arsitektur umpan-maju (feedforward) dengan konsep belajar algoritma belajar perambatan-balik (backpropagation). Kaidah backpropagation dapat diterapkan pada jaringan lapis banyak, karena backpropagation mempunyai kemampuan mengajar sel-sel syaraf pada lapisan dalam. Kesalahan lokal setiap sel dilihat (diasumsikan) sebagai bagian yang berkontribusi dalam menghasilkan kesalahan total pada lapisan keluaran. Apabila kesalahan pada lapis keluaran dapat dipropagasikan kembali masuk ke lapisan dalam, maka kesalahan lokal sel-sel syaraf pada lapis tersebut dapat dihitung. Pelatihan jaringan backpropagation meliputi tiga tahap yaitu memasukan secara umpan maju (feedforward) pola-pola masukan, menghitung dan propagasi balikan kesalahan yang bersangkutan dan mengatur bobot-bobot koneksi. Setelah selesai pelatihan JST diterapkan untuk penyelesaian masalah (mode pengujian).

Kata kunci : Jaringan syaraf tiruan, backpropagation, pelatihan, pengujian.

1. PENDAHULUAN

Tubuh manusia merupakan aspirasi dalam pengembangan teknologi. Dalam tubuh manusia, sistem syaraf yang berpusat di otak memegang peranan yang sangat penting. Sistem syaraf merupakan komponen pengolah sinyal, dalam menentukan tanggapan yang harus diberikan apabila ditemui suatu permasalahan tertentu. Sampai saat ini sistem syaraf pusat yang diyakini sebagai elemen pokok yang menyebabkan manusia mampu belajar.

Jaringan Syaraf Tiruan (JST) adalah model sistem komputasi yang dapat bekerja seperti sistem syaraf biologis pada saat berhubungan dengan 'dunia luar', nama jaringan syaraf tiruan merupakan terjemahan dari "Artificial Neural Network" [Utama, 1995]. Terjemahan yang diambil bukan jaringan syaraf buatan seperti dalam menterjemahkan Artificial Intelligent (AI). Penggunaan kata buatan dapat memberikan konotasi, bahwa manusia berusaha membuat jaringan syaraf aslinya. Padahal maksud dari JST adalah membuat model sistem komputasi yang dapat menirukan cara kerja jaringan syaraf biologis.

Pada dasarnya jaringan syaraf terdiri dari banyak elemen pemroses sederhana yang disebut *neuron*, sel, unit atau simpul. Sebagai bahan perbandingan, jumlah sel syaraf otak seekor cacing diperkirakan ada 1000 buah dan pada otak manusia kurang lebih 100 milyar buah [Utama, 1995]. Setiap sel syaraf berhubungan dengan sel syaraf lainnya memakai saluran komunikasi (*communication link*) yang teratur dengan suatu bobot penghubung. Bobot-bobot tersebut digunakan oleh jaringan untuk menyelesaikan masalah. JST mempunyai beberapa sifat yang dimiliki otak manusia, yaitu [Arwin, 1995] :

- Kemampuan untuk belajar dari pengalaman.
- Kemampuan melakukan perumuman (*generalization*) terhadap input baru dari pengalaman yang dimilikinya.
- Kemampuan memisahkan (*abstraction*) karakteristik penting dari input yang mengandung data yang tidak penting.

JST merupakan bidang teknologi yang berkembang sangat pesat. Perkembangan terjadi antara lain dalam pemodelan sel dan jaringan, konsep belajar dan aplikasi jaringan.

Adapun model JST yang digunakan dalam paper ini adalah arsitektur umpan-maju (*feedforward*). Sedangkan konsep belajar yaitu algoritma belajar *momentum* perambatan-balik (*backpropagation*) yang merupakan perkembangan dari algoritma belajar perambatan-balik standar^[Fausett, 1994].

2. PEMODELAN JST

Jaringan Syaraf Tiruan (JST) adalah sebuah sistem pemroses informasi yang mempunyai karakteristik prestasi tertentu yang sama dengan jaringan syaraf biologis. JST secara umum dikembangkan pada model-model matematis dari cara berpikir manusia atau syaraf biologis tersebut, yang berdasar pada asumsi berikut^[Fausett, 1994].

1. Pemrosesan informasi terjadi pada beberapa elemen sederhana yang disebut sel atau sel syaraf (*neuron*).
2. Sinyal-sinyal dilewatkan diantara sel dengan *connection links*.
3. Setiap *connection links* mempunyai sebuah bobot (*weight*), yang mengalikan sinyal yang disalurkan dalam suatu JST tipikal.
4. Masing-masing sel menerapkan sebuah fungsi *activation* terhadap jumlah dari sinyal-sinyal masukan terbobot (*weighted*) jaringannya untuk menentukan sinyal keluaran.

Sebuah Jaringan Syaraf Tiruan mempunyai tiga karakteristik sebagai berikut^[Fausett, 1994] :

1. Pola hubungan antar sel (disebut arsitektur).
2. Metode penentuan bobot-bobot koneksi (disebut belajar atau pelatihan dan algoritma).
3. Fungsi *activation*-nya.

Fungsi Activation ^[Fausett, 1994]

Suatu fungsi activation untuk setiap sel sebuah Jaringan Syaraf Tiruan perambatan-balik mempunyai beberapa karakteristik penting sebagai berikut :

- Harus kontinyu, dapat diturunkan.

- Tidak naik atau turun secara monoton (*monotonically nondecreasing*).

Selanjutnya untuk mendapatkan efisiensi perhitungan, turunannya mudah dihitung. Pada umumnya nilai turunan fungsi *activation* dapat diperoleh dari rentang nilai tertentu. Fungsi diharapkan jenuh, yaitu mendekati batas minimum dan maksimum (dari asimtotnya) atau fungsi pemampat (*squashing function*) ^[Sandi S, 1993].

Suatu fungsi activation tipikal adalah fungsi sigmoid biner yang mempunyai rentang (0, 1) dan didefinisikan sebagai

$$f_1(x) = \frac{1}{1 + e^{-x}} \quad (1)$$

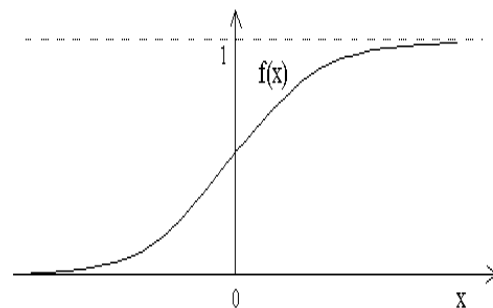
turunannya adalah,

$$f_1'(x) = \frac{e^{-x}}{(1 + e^{-x})^2}$$

atau,

$$f_1'(x) = f_1(x) \{1 - f_1(x)\} \quad (2)$$

Fungsi ini digambarkan dalam gambar 1.



Gambar 1 Kurva Fungsi Sigmoid Biner, Rentang (0, 1) ^[Fausett, 1994].

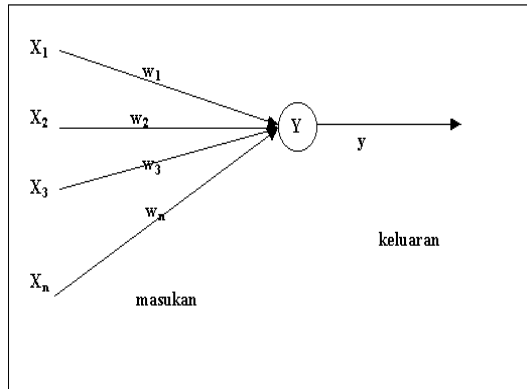
Model Matematis Jaringan Syaraf Tiruan ^[Fausett, 1994]

Sebagai contoh, sebuah sel syaraf Y dalam gambar 2 yang menerima masukan dari sel-sel $X_1, X_2, X_3, \dots, X_n$. Activation (sinyal keluaran) masing-masing sel adalah $x_1, x_2, x_3, \dots, x_n$. Bobot-bobot koneksi dari $X_1, X_2, X_3, \dots, X_n$ ke sel syaraf Y adalah $w_1, w_2, w_3, \dots, w_n$. Masukan jaringan, y_{in} pada sel syaraf Y adalah jumlah dari sinyal-sinyal terbobot dari sel-sel $X_1, X_2, X_3, \dots, X_n$ yaitu

$$y_{in} = w_1x_1 + w_2x_2 + w_3x_3 + \dots + w_nx_n \quad (3)$$

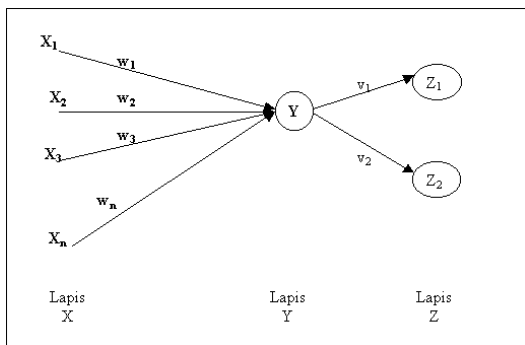
Activation sel Y yaitu y diberikan pada jumlah masukan terbobot, misalnya $y = f(y_{in})$ adalah fungsi logistic sigmoid (sebuah kurva berbentuk huruf S)

$$f(y_{in}) = \frac{1}{1 + \exp(-y_{in})} \quad (4)$$



Gambar 2 Sel Syaraf (Neuron) Tiruan Sederhana [Fausett, 1994].

Selanjutnya dimisalkan bahwa sel syaraf Y dihubungkan dengan sel-sel syaraf Z1 dan Z2 masing-masing dengan bobot v_1 dan v_2 seperti dalam gambar 3. Sel Y mengirimkan sinyal y -nya ke setiap unit ini. Tetapi pada umumnya nilai-nilai yang diterima oleh sel-sel syaraf Z1 dan Z2 adalah berbeda, karena tiap-tiap sinyal diskala dengan bobot-bobot v_1 dan v_2 . Dalam sebuah jaringan tipikal activation z_1 dan z_2 dari sel-sel Z1 dan Z2 akan tergantung pada masukan dari beberapa sel syaraf tidak hanya satu seperti dalam kasus sederhana ini.



Gambar 3 Sebuah Jaringan Syaraf Tiruan Sederhana [Fausett, 1994].

Meskipun jaringan dalam gambar 3 sangat sederhana, jaringan ini menghadirkan sebuah “unit dalam” bersama dengan sebuah activation nonlinier, memberikan kemampuan untuk menyelesaikan masalah lebih banyak daripada sebuah jaringan tanpa unit dalam.

Konsep Belajar JST [Syahirul, 1991]

Ciri utama yang dipunyai oleh sistem JST adalah kemampuan untuk belajar. Agar berfungsi seperti yang diinginkan, jaringan tidak diprogram seperti yang dilakukan pada sistem komputer masa lalu (program ad hoc), tetapi harus diajari.

Berdasarkan fungsi masukan keluarannya, dapat dilihat bahwa fungsi JST ditentukan oleh parameternya (bobot-bobot koneksi). Untuk kasus yang diketahui fungsi pemetaannya, bobot-bobot tersebut dapat berharga tetap dan ditentukan pada waktu perancangan. Tetapi pada kebanyakan kasus, parameter jaringan yang cocok belum diketahui, dan jaringan harus mencari sendiri besarnya bobot tersebut.

Suatu proses penyesuaian parameter secara berurutan dilakukan, dengan tujuan mendekati fungsi yang diinginkan. Proses penyesuaian parameter inilah yang disebut dengan proses belajar dalam sistem JST.

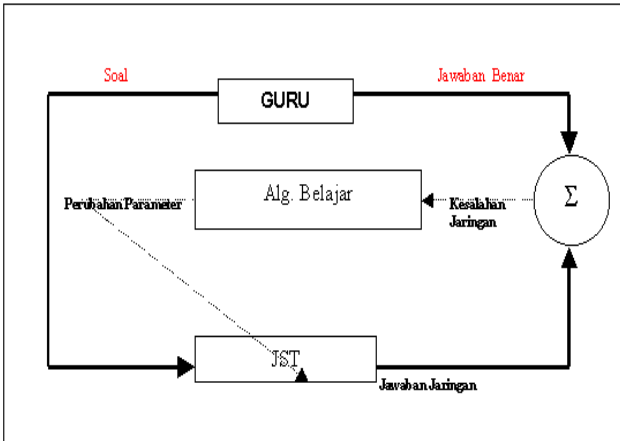
Poses belajar dapat dikategorikan dalam dua jenis :

1. Dengan pengawasan (supervised learning)
2. Tanpa Pengawasan (unsupervised learning).

Proses belajar dengan pengawasan memerlukan keluaran/jawaban yang diinginkan, sebagai dasar pengubahan bobot. Pasangan masukan (soal) - keluaran target (jawaban benar) yang diperlukan dalam proses belajar. Jaringan diajar untuk menyelesaikan persoalan-persoalan yang terdapat dalam paket belajarnya.

Diagramatik dari konsep belajar dengan pengawasan ditunjukkan pada gambar 4. Selama belajar apabila jaringan mengeluarkan jawaban yang salah, maka besar kesalahan dapat dicari, yaitu beda keluaran aktual dan acuannya.

Sedangkan dalam belajar tanpa pengawasan, jaringan akan mengubah bobot-bobotnya, sebagai tanggapan terhadap masukan, tanpa memerlukan keluaran acuan. Dalam paper ini proses belajar yang digunakan adalah proses belajar dengan pengawasan.



Gambar 4 Diagram Blok Proses Belajar Dengan Pengawasan [Syahirul, 1991].

Aturan Delta (Delta Rule)

Aturan delta [Syahirul, 1991] [Fausett, 1994] adalah proses belajar yang diperkenalkan oleh Widrow dan Hoff (1960) digunakan untuk pemetaan pola. Dasar aturan ini adalah merubah bobot koneksi antar sel sedemikian hingga meminimasi perbedaan keluaran aktual dan acuan (target).

Notasi Untuk Aturan Delta

- y_{inj} jumlah masukan sel Y ke-j.
- y_j keluaran sel Y ke-j.
- β laju belajar.
- x_i vektor keluaran sel X ke-i, masukan sel Y ke-j.
- w_{ij} bobot koneksi antara sel i ke sel j.
- t_j keluaran target dari sel Y ke-j.
- δ_k porsi koreksi kesalahan untuk pengaturan bobot w_{ij}.

Persamaan aturan delta

Dari gambar 2 dan 3 diperoleh :

$$y_{inj} = \left(\sum_{i=1}^n x_i w_{i,j} \right) \tag{5}$$

$$y_j = f(y_{inj}) \tag{6}$$

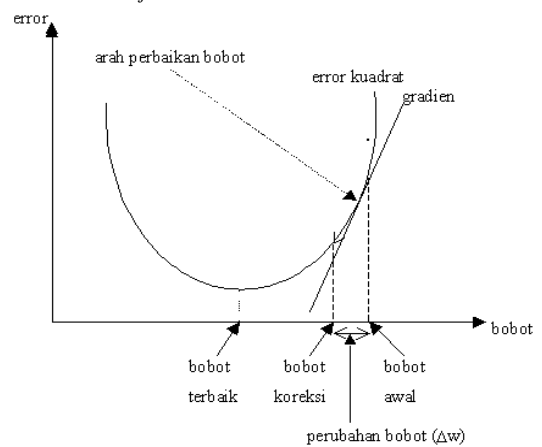
Misalkan error untuk sebuah pola pelatihan e.

$$e = \sum_{j=1}^p (t_j - y_j) \tag{7}$$

error kuadratnya adalah E.

$$E = \sum_{j=1}^p (t_j - y_j)^2$$

Maka E merupakan fungsi dari bobot w_{ij}, i = 1, 2, ..., n dan j = 1, 2, ..., p. Gradien E terdiri dari vektor turunan parsial dari E terhadap bobot-bobot (w_{ij}). Gradien dalam arah positif (naik) akan menambah E, gradien dalam arah negatif (turun) akan memperkecil E. Error dapat dikurangi dengan pengaturan bobot w_{ij} sebanding dengan $-\frac{\partial E}{\partial w_{ij}}$ (seperti pada gambar 5).



Gambar 5 Kurva Penurunan Gradien Aturan Delta.

$$\Delta w_{ij} \approx - \frac{\partial E}{\partial w_{ij}}$$

$$\Delta w_{ij} = - \beta \frac{\partial E}{\partial w_{ij}} \tag{8}$$

Sebelumnya diketahui

$$y_{inj} = \sum_{i=1}^n x_i w_{i,j} \text{ dan } y_j = f(y_{inj})$$

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial \sum_{j=1}^p (t_j - y_j)^2}{\partial w_{ij}}$$

bila bobot w_{ij} hanya dipengaruhi oleh error keluaran unit Y_j maka :

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial (t_j - y_j)^2}{\partial w_{ij}}$$

$$\frac{\partial E}{\partial w_{ij}} = -2 (t_j - y_j) \cdot \frac{\partial y_j}{\partial w_{ij}}$$

$$\frac{\partial E}{\partial w_{ij}} = -2 (t_j - y_j) x_i \cdot f'(y_{in_j})$$

$$-\frac{\partial E}{\partial w_i} = 2 (t_j - y_j) x_i \cdot f'(y_{in_j})$$

Sehingga error lokal sel dapat dikurangi secara cepat dengan pengaturan bobot menurut aturan delta (didiberikan parameter belajar β)

$$\Delta w_i = \beta (t_j - y_j) x_i \cdot f'(y_{in_j})$$

atau

$$\Delta w_i = \beta \delta_j \cdot x_i \tag{9}$$

dimana,

$$\delta_j = (t_j - y_j) \cdot f'(y_{in_j})$$

3. JST PERAMBATAN-BALIK [Syahirul, 1991] [Fausett, 1994]

Jaringan Syaraf Tiruan Perambatan-Balik pertama kali diperkenalkan oleh Rumelhart, Hinton dan William pada tahun 1986, kemudian Rumelhart dan Mc Clelland mengembangkannya pada tahun 1988.

Jaringan Syaraf Tiruan perambatan-balik (PB) adalah jaringan yang biasanya jaringan umpan maju, yang menggunakan PB sebagai kaidah belajar. Jaringan ini merupakan salah satu jenis yang mudah dipahami. Konsep belajarnya relatif sederhana, yaitu belajar dari kesalahan.

Kaidah PB dapat diterapkan pada jaringan lapis banyak, karena PB mempunyai kemampuan mengajar sel-sel syaraf pada lapisan dalam. Kesalahan lokal setiap sel dilihat (diasumsikan) sebagai bagian yang berkontribusi dalam menghasilkan kesalahan total pada lapisan keluaran. Apabila kesalahan pada lapis keluaran dapat dipropagasikan kembali masuk ke lapisan dalam, maka kesalahan lokal sel-sel syaraf pada lapis tersebut dapat dihitung. Secara matematis masalah tersebut adalah turunan parsial.

Pelatihan jaringan PB meliputi tiga tahap :

- Memasukan secara umpan maju (*feedforward*) pola-pola masukan
- Menghitung dan propagasi balikan kesalahan

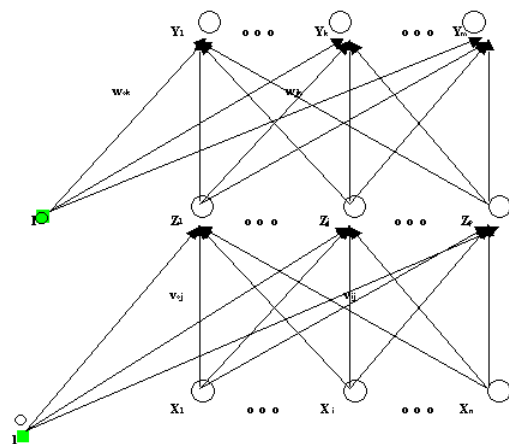
yang bersangkutan

- Mengatur bobot-bobot koneksi.

Setelah pelatihan selesai komputasi jaringan hanya pada fase umpan maju yaitu memasukan pola-pola masukan. Sekalipun pelatihan lambat cara kerja jaringan terlatih dapat menghasilkan keluaran sangat cepat.

Arsitektur JST PB

Sebuah jaringan banyak lapis dengan satu lapisan dalam (unit-unit Z) ditunjukkan dalam gambar 6. Unit-unit keluaran (Y) dan unit-unit dalam (Z) mempunyai *bias*. Bias pada unit keluaran tipikal Y_k dinotasikan dengan w_{ok} , bias pada unit keluaran tipikal Z_j dinotasikan dengan v_{oj} . Tindakan bias ini seperti pada bobot-bobot koneksi, dimana unit-unit bias selalu mengeluarkan nilai 1. Unit-unit ini bersifat optional pada penyajian lain mungkin dihilangkan.



Gambar 6. JST PB Dengan Satu Lapisan Dalam.

Gambar 6 tersebut menunjukkan arah sinyal pada fase umpan maju. Selama operasi fase pelatihan perambatan-balik, sinyal-sinyal error dikirim dengan arah yang sebaliknya.

Notasi Algoritma PB

Penentuan notasi (tata nama) yang digunakan dalam pelatihan dengan algoritma belajar perambatan-balik adalah sebagai berikut.

x vektor masukan pelatihan :

$$x = (x_1, \dots, x_i, \dots, x_n).$$

t vektor keluaran target :

$$t = (t_1, \dots, t_k, \dots, t_m).$$

δ_k porsi koreksi kesalahan untuk pengaturan bobot w_{jk} , juga merupakan informasi kesalahan pada unit Y_k .

δ_j porsi koreksi kesalahan untuk pengaturan bobot v_{ij} yaitu perambatan-balik informasi kesalahan dari lapisan keluaran ke unit dalam Z_j .

β laju belajar.

X_i unit masukan ke- i .

sinyal-sinyal masukan yang bersangkutan diberi nama x_i .

v_{0j} bias pada unit dalam ke- j .

Z_j unit dalam ke- j .

masukan jaringan pada Z_j dinotasikan dengan z_in_j :

$$z_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij}$$

sinyal keluaran (*activation*) dari Z_j dinotasikan z_j :

$$z_j = f(z_in_j)$$

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

Y_k unit keluaran ke- k :

masukan pada unit keluaran Y_k dinotasikan y_ink :

$$y_ink = w_{0k} + \sum_{j=1}^p z_j w_{jk}$$

sinyal keluaran (*activation*) dari Y_k dinotasikan dengan y_k :

$$y_k = f(y_ink)$$

Algoritma PB

Seperti telah ditunjukkan dimuka, pelatihan sebuah jaringan (Perambatan-Balik) PB meliputi tiga tahap : pelatihan umpan maju pola-pola masukan, mempropagasikan kesalahan yang bersangkutan dan pengaturan bobot. Pengaturan bobot berdasar aturan delta.

Selama umpan maju setiap unit masukan (X_i) menerima sebuah sinyal dan memancarkan sinyal ini ke unit-unit dalam Z_1, \dots, Z_p . Kemudian setiap unit dalam melakukan perhitungan dengan fungsi *activation* dan mengirimkan sinyal-sinyalnya (z_j) ke setiap unit keluaran. Selanjutnya setiap unit keluaran (Y_k) melakukan perhitungan dengan fungsi *activation* dan memberikan sinyal keluaran (y_k) yang merupakan respon jaringan tersebut untuk pola-pola masukan yang diberikan.

Selama pelatihan setiap unit keluaran membandingkan *activation* keluaran yang dihasilkan y_k dengan nilai targetnya t_k untuk menentukan kesalahan yang berhubungan dengan pola-pola pada unit tersebut. Berdasar pada kesalahan ini, besaran δ_k ($k=1,2,\dots,m$) dihitung. δ_k digunakan untuk mendistribusi-balikan kesalahan yang terdapat pada unit keluaran Y_k ke semua unit pada lapisan sebelumnya (unit-unit dalam yang berhubungan dengan unit keluaran Y_k). δ_k juga digunakan untuk memperbaiki bobot-bobot antara lapisan keluaran dan lapisan dalam. Dengan cara yang sama, besaran δ_j ($j=1,\dots,p$) dihitung pada setiap unit Z_j . Tetapi δ_j tidak dipergunakan untuk mempropagasi-balikkan kesalahan, δ_j dipergunakan untuk memperbaiki bobot-bobot antara lapisan hidden dan masukan.

Setelah semua besaran δ_j ditentukan, secara serentak bobot-bobot pada semua lapis diatur. Pada pengaturan bobot w_{jk} (dari unit

dalam Z_j ke unit keluaran Y_k) berdasarkan pada besaran δ_j dan *activation* z_j dari unit dalam Z_j . Pengaturan bobot-bobot v_{ij} (dari unit masukan X_i ke unit dalam Z_j) berdasar pada besaran δ_j dan *activation* x_i dari unit masukan X_i .

Fungsi *activation* yang didefinisikan didepan digunakan untuk pelatihan JST dengan algoritma perambatan-balik standar. Algoritma belajar Jaringan Syaraf Tiruan Perambatan-Balik adalah sebagai berikut :

- Langkah 0. Tentukan bobot-bobot awal (acak).
- Langkah 1. Sementara dalam kondisi salah (bobot tidak sesuai) kerjakan langkah 2-9.
- Langkah 2. Untuk setiap pasangan pelatihan, kerjakan langkah 3-8.
- Umpan maju (feedforward) :
- Langkah 3. Setiap unit masukan (X_i , $i = 1, \dots, n$) menerima sinyal-sinyal masukan x_i dan menyalurkan sinyal-sinyal ini ke unit-unit di atasnya (unit-unit dalam).
- Langkah 4. Pada setiap unit dalam (Z_j , $j = 1, \dots, p$) menjumlahkan sinyal-sinyal masukan terbobotnya,

$$z_{inj} = v_{oj} + \sum_{i=1}^n x_i v_{ij} \quad (10)$$

terapkan fungsi *activation* unuk menghitung sinyal keluaran,

$$z_j = f(z_{inj}) \quad (11)$$

dan kirimkan sinyal ini pada semua unit pada lapisan di atasnya (lapisan keluaran).

- Langkah 5. Pada setiap unit keluaran (Y_k , $k = 1, \dots, m$) jumlahkan sinyal-sinyal masukan terbobotnya,

$$y_{ink} = w_{ok} + \sum_{j=1}^p z_j w_{jk} \quad (12)$$

dan terapkan fungsi aktivasi untuk menghitung sinyal keluarannya,

$$y_k = f(y_{ink}) \quad (13)$$

Perambatan-balik kesalahan:

- Langkah 6. Pada setiap unit keluaran (Y_k , $k = 1, \dots, m$) menerima sebuah pola keluaran target yang berhubungan dengan pola masukan pelatihan, untuk menghitung informasi kesalahannya,

$$\delta_k = (t_k - y_k) f'(y_{ink}) \quad (14)$$

hitung besar koreksi bobotnya (untuk memperbaiki w_{jk})

$$\Delta w_{jk} = \beta \delta_k z_j \quad (15)$$

hitung besar koreksi biasnya (yang nantinya untuk memperbaiki w_{ok})

$$\Delta w_{ok} = \beta \delta_k \quad (16)$$

dan kirimkan δ_k ke unit-unit lapisan dibawahnya.

- Langkah 7. Pada setiap unit dalam (Z_j , $j = 1, \dots, p$) jumlahkan masukan deltanya (dari unit-unit lapisan di atasnya/lapisan keluaran),

$$\delta_{inj} = w_{ok} + \sum_{k=1}^m \delta_k w_{jk} \quad (17)$$

kalikan dengan turunan fungsi *activation* untuk menghitung besar informasi kesalahannya,

$$\delta_j = \delta_{inj} f'(y_{inj}) \quad (18)$$

hitung besar koreksi bobotnya (untuk memperbaiki v_{ij})

$$\Delta v_{ij} = \beta \delta_j x_i \quad (19)$$

dan hitung koreksi biasanya (untuk memperbaiki v_{0j})

$$\Delta v_{0j} = \beta \delta_j \quad (20)$$

Perbaiki bobot-bobot dan bias :

Langkah 8. Masing-masing unit keluaran (Y_k , $k = 1, \dots, m$) diperbaiki bias dan bobotnya,

$$w_{jk} \text{ (baru)} = w_{jk} \text{ (lama)} + \Delta w_{jk} \quad (21)$$

dimana ($j = 1, \dots, p$),

dan pada setiap unit dalam (Z_j , $j = 1, \dots, p$) diperbaiki bias dan bobotnya,

$$v_{ij} \text{ (baru)} = v_{ij} \text{ (lama)} + \Delta v_{ij} \quad (22)$$

dimana ($i = 1, \dots, n$).

Langkah 9. Berhenti.

Suatu jangka waktu (disebut *epoch*) adalah satu set putaran vektor-vektor pelatihan. Beberapa *epoch* diperlukan untuk pelatihan sebuah JST PB. Dalam algoritma ini dilakukan perbaikan bobot setelah masing-masing pola pelatihan disajikan. Setelah pelatihan selesai bobot-bobot yang telah diperbaiki disimpan.

Prosedur Pengujian (Aplikasi JST)

Setelah pelatihan, sebuah JST PB hanya menggunakan fase umpan majunya (*feedforward*). Prosedur aplikasinya adalah sebagai berikut :

Langkah 0. Penentuan bobot awal (hasil dari pelatihan).

Langkah 1. Untuk setiap vektor masukan, kerjakan langkah 2-4.

Langkah 2. Untuk setiap unit masukan distribusikan masukan x_i ke setiap unit di atasnya (unit dalam).

Langkah 3 Untuk $j = 1, \dots, p$; terapkan persamaan (10 dan 3.11) :

$$z(in_p) = v_{0p} + \sum_{i=1}^n x_i v_{ip}$$

dan

$$z_p = f \{z(in_p)\}$$

Langkah 4. Untuk $k = 1, \dots, m$; terapkan persamaan (12 dan 3.13) :

$$y(in_m) = w_{0m} + \sum_{j=1}^p z_j w_{jm}$$

dan

$$y_m = f \{y(in_m)\}.$$

4. PERBAIKAN BOBOT MOMENTUM ^[Fausett, 1994]

Salah satu pengembangan dalam Jaringan Syaraf Tiruan Perambatan-Balik (JST PB) untuk memperbaiki bobot digunakan parameter *momentum* (α). JST PB dengan *momentum* tersebut mempunyai sifat konvergensi lebih cepat dari JST PB standar. Rumus untuk pengubahan bobot algoritma *Momentum* Perambatan-Balik adalah :

$$w_{jk}(t+1) = w_{jk}(t) + \beta \delta_k z_j + \alpha \{w_{jk}(t) - w_{jk}(t-1)\}, \quad (23)$$

atau

$$\Delta w_{jk}(t+1) = \beta \delta_k z_j + \alpha \Delta w_{jk}(t),$$

dan

$$v_{ij}(t+1) = v_{ij}(t) + \beta \delta_j x_i + \alpha \{v_{ij}(t) - v_{ij}(t-1)\}, \quad (24)$$

atau

$$\Delta v_{ij}(t+1) = \beta \delta_j x_i + \alpha \Delta v_{ij}(t),$$

Dalam proses belajar diatas ada satu hal yang perlu diperhatikan. Berapakah nilai yang optimal untuk laju belajar β dan *momentum* α ? Sebagian besar JST PB menggunakan β dan

α tetap. Rumelhart merekomendasikan sebuah kombinasi $\beta=0,25$ dan $\alpha=0,9$ dapat memberikan hasil yang terbaik pada banyak masalah. Tetapi belum ada konsensus besar nilai β dan α yang harus digunakan dalam proses belajar. Sebab dalam kenyataannya nilai β dan α tergantung pada permasalahan yang dihadapi.

5. KESIMPULAN

Dari ulasan diatas dapat diambil beberapa kesimpulan sebagai berikut :

1. Kelebihan JST terletak pada kemampuan belajar yang dimilikinya. Dengan kemampuan tersebut JST dapat digunakan terutama dalam bidang yang berhubungan dengan pola-pola. Bidang yang intinya masalah pemetaan. Untuk menyatakan pemetaan tersebut pengguna tidak perlu merumuskan kaidah atau fungsinya. JST akan belajar mencari sendiri kaidah atau fungsi tersebut. Dengan demikian JST mampu digunakan untuk menyelesaikan masalah yang rumit dan atau masalah yang terdapat kaidah atau fungsi yang tidak diketahui.
2. Jaringan Syaraf Tiruan Perambatan-Balik ini termasuk jenis *hetero-associative*, dimana vektor masukan dipetakan ke vektor keluaran target.
Merupakan jaringan yang sederhana dengan banyak lapis dan banyak sel. Meskipun jaringan ini tidak mempunyai umpan balik tetapi error dibalikan untuk memperbaiki bobot (proses belajar). Dari hasil belajarnya Jaringan mampu menyelesaikan masalah yang diberikannya dengan error yang kecil.
3. Jaringan Syaraf Tiruan Perambatan-Balik dalam proses belajarnya memiliki parameter konstanta belajar α dan β untuk mempercepat proses belajar. Selama ini belum ada konsensus mengenai nilai α dan β yang tepat.

DAFTAR PUSTAKA

- Arwin, D. , Wahyudi, S., 1995, **JST Dalam Komputer Cerdas, Bagian I, II dan III**, Mokrodata 9 seri 10, PT Gramedia, Jakarta.
- Fausett, L., 1994, **Fundamental Of Neural Network Architectures, Algorithms And**

Application, Prentice Hal, Englewood Cliffs, New Jersey.

Sandi Setiawan, 1993, **Artificial Intellegence**, Andi offset, Yogyakarta.

Syahirul Hakim AD Dairi, 1991, **Karakteristik Jaringan Syaraf Tiruan dengan Arsitektur Feed-Forward**, Tugas Akhir, ITB.

Utama, W., 1995, **Menuju Era Teknologi Jaringan Syaraf Tiruan**, Mikrodata 9 seri 10, PT Gramedia, Jakarta.

D. Biografi

Subiyanto lahir di Klaten, 23 Nopember 1974. Lulus Sarjana Teknik Elektro UNDIP pada tahun 1998 dengan mempunyai rangking I, lulus Pascasarjana S2 Teknik Elektro UGM 2003. Sejak desember 2004 menjadi staf pengajar di Teknik Elektro Universitas Negeri Semarang. Bidang yang diminati Penggunaan Komputer dalam Sistem Tenaga Listrik.