

IMPLEMENTASI ALGORITMA BACKPROPAGATION (JST) PREDIKSI TINGKAT KELULUSAN SMK KOTA JAMBI

Mardan Junjung

AMIK Depati Parbo Kerinci

ABSTRAK

Penelitian ini menerapkan jaringan syaraf tiruan untuk memprediksi tingkat kelulusan siswa. Tujuan dari penelitian ini adalah melihat hasil prediksi, agar pengguna dapat melakukan perubahan terhadap sistem-sistem yang ada pada sekolah masing-masing, untuk meningkatkan tingkat kelulusan siswa untuk tahun-tahun yang akan datang berdasarkan kelulusan masa lalu yang telah diprediksi. Pengumpulan data dilakukan dengan observasi dan interview yang dilaksanakan di tempat melakukan penelitian, baik dengan para guru-guru maupun siswa yang bersangkutan. Data yang telah diperoleh dikumpulkan kemudian diolah, dianalisis dan dipelajari. Dirumuskan agar nantinya bisa melakukan penelitian lebih lanjut. Penelitian menggunakan teknik jaringan syaraf tiruan, dengan menggunakan software Matlab. Hasil penelitian ini menunjukkan bahwa jaringan syaraf tiruan dapat mengolah data yang berubah sewaktu-waktu atau bertahun-tahun.

Kata kunci : Artificial Intelligent, Jaringan syaraf Tiruan, Backpropagation, Prediksi Tingkat Kelulusan siswa SMK

PENDAHULUAN

Jaringan syaraf tiruan adalah merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba mensimulasikan proses pembelajaran pada otak manusia tersebut. Istilah tiruan disini digunakan karena jaringan syaraf ini diimplementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran. (Sri Kusumadewi : 2004).

Berdasarkan pada penjelasan di atas maka penulis mencoba menggunakan jaringan syaraf tiruan untuk melakukan prediksi terhadap tingkat kelulusan siswa untuk tahun – tahun yang akan datang, berdasarkan faktor – faktor yang mempengaruhi pada masing – masing sekolah. Penulis akan mengadakan penelitian terhadap 3 (tiga) sekolah, yaitu SMK Negeri 1 Kota Jambi, SMK Negeri 2 Kota Jambi, SMK Negeri 3 Kota Jambi.

Konsep Dasar

Setiap pola-pola informasi input dan output yang diberikan kedalam jaringan syaraf tiruan diproses dalam neuron. Neuron-neuron tersebut berkumpul

di dalam lapisan-lapisan yang disebut neuron layer. Lapisan-lapisan penyusun jaringan syaraf tiruan tersebut dapat di bagi menjadi tiga bagian :

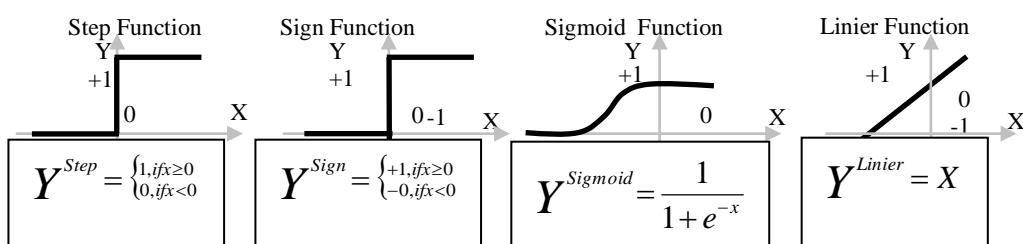
- Lapisan input, Node-node didalam lapisan input disebut unit-unit input. Unit-unit input tersebut menerima pola inputan data dari luar yang menggambarkan suatu permasalahan.
- Lapisan tersembunyi, Node-node didalam lapisan tersembunyi disebut unit-unit tersembunyi. Dimana outputnya tidak dapat secara langsung diamati.
- Lapisan output, Node-node didalam lapisan output disebut unit-unit output. Output dari lapisan ini merupakan solusi jaringan syaraf tiruan terhadap suatu permasalahan.

Pengertian Jaringan Saraf Tiruan

Jaringan saraf adalah sebuah prosesor yang terdistribusi paralel dan mempunyai kecenderungan untuk menyimpan pengetahuan yang didapatkannya dari pengalaman dan membuatnya tetap tersedia untuk digunakan

Mengaktifkan Jaringan Syaraf Tiruan

Mengaktifkan jaringan syaraf tiruan berarti mengaktifkan setiap neuron yang dipakai pada jaringan tersebut. Banyak fungsi yang dapat dipakai sebagai pengaktif, seperti fungsi-fungsi goniometri dan hiperboliknya, fungsi unit step, seumlse, segmod, dan lain sebagainya pada gambar dibawah ini tetapi yang lazim digunakan adalah fungsi sigmod, karena dianggap lebih mendeteksi kinerja sinyal pada otak manusia .



Keuntungan Menggunakan Komputasi dengan JST :

Menurut Jong Jek Siang (2005), Kemampuan dan proses komputasi pada JST memberikan keuntungan-keuntungan sebagai berikut :

-
- JST bersifat adaptif terhadap perubahan parameter yang mempengaruhi karakteristik sistem, sehingga pada proses belajar, JST mampu belajar secara adaptif dan melaksanakan tugas berbasis pada data yang diberikan saat pelatihan.
 - JST memiliki kekebalan atau toleran terhadap kesalahan. Artinya, JST tetap berfungsi walaupun ada ketidak-lengkapan data yang dimasukkan.
 - JST dapat dilatih memberikan keputusan dengan memberikan set pelatihan sebelumnya untuk mencapai target tertentu, sehingga JST mampu membangun dan memberikan jawaban sesuai dengan informasi yang diterima pada proses pelatihan.
 - JST mempunyai struktur paralel dan terdistribusi. Artinya, komputasi dapat dilakukan oleh lebih dari satu elemen pemroses yang bekerja secara simultan.
 - JST mampu mengklasifikasi pola masukan dan pola keluaran. Melalui proses penyesuaian, pola keluaran dihubungkan dengan masukan yang diberikan oleh JST.
 - JST mengurangi derau, sehingga dihasilkan keluaran yang lebih bersih.
 - JST dapat dimanfaatkan pada proses optimisasi penyelesaian suatu masalah.
 - JST dapat digunakan pada proses pengendalian sistem agar masukan memperoleh tanggapan yang diinginkan.

Algoritma Umum Jaringan Saraf Tiruan

Algoritma pembelajaran/pelatihan jaringan saraf tiruan. Dimasukkan n contoh pelatihan ke dalam jaringan saraf tiruan. Lakukan :

- Inisialisasi bobot-bobot jaringan. Set $i = 1$.
- Masukkan contoh ke- i (dari sekumpulan contoh pembelajaran yang terdapat dalam set pelatihan) ke dalam jaringan pada lapisan input.
- Cari tingkat aktivasi unit-unit output menggunakan algoritma aplikasi.
If kinerja jaringan memenuhi standar yang ditentukan sebelumnya (memenuhi syarat berhenti) *Then exit*.
- Update bobot-bobot dengan menggunakan aturan pembelajaran jaringan.
If $i=n$, then reset $i = 1$.

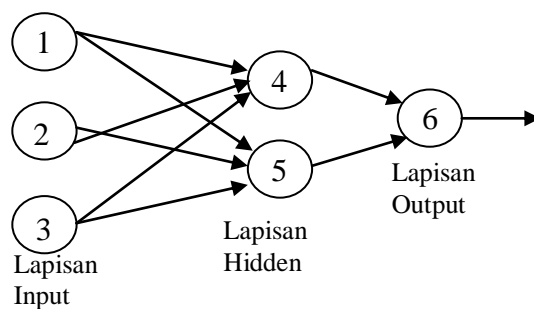
Else $i = i - 1$. Ke langkah 2

Arsitektur Jaringan

Menurut Jong Jek Siang (2005), Beberapa arsitektur jaringan yang sering dipakai dalam jaringan syaraf tiruan antara lain :

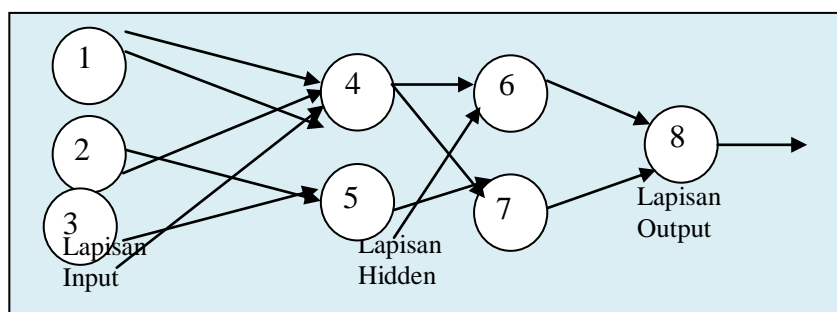
- Jaringan Layer Tunggal (*single layer network*)

Dalam jaringan ini, sekumpulan input neuron dihubungkan langsung dengan sekumpulan outputnya. Dalam beberapa model (misal *perceptron*), hanya ada sebuah *unit neuron output*.



- Jaringan Layer Jamak (*multi layer network*)

Jaringan layer jamak merupakan perluasan dari layer tunggal. Dalam jaringan ini, selain *unit input* dan *output*, ada *unit-unit* lain (sering disebut layer tersembunyi atau *hidden layer*). Dimungkinkan pula ada beberapa layer tersembunyi. Sama seperti pada *unit input* dan *output*, *unit-unit* dalam satu layer tidak saling berhubungan.



Fungsi Aktivasi

Fungsi aktivasi adalah fungsi yang menggambarkan hubungan antara tingkat aktivasi internal (*summation function*) yang mungkin berbentuk linear atau

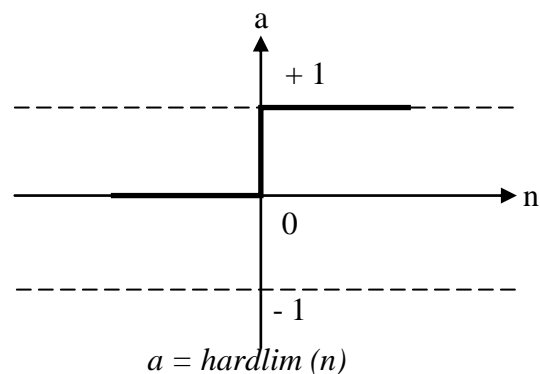
(*layer*) yang disebut dengan lapisan neuron (*neuron layer*) Beberapa aktivasi jaringan saraf tiruan yang umum digunakan adalah sebagai berikut :

- **Fungsi Undak Biner (Hardlim)**

Jaringan dengan lapisan tunggal sering menggunakan fungsi *undak biner* (*step function*) untuk mengkompersikan input dari suatu variabel yang bernilai kontinu kesuatu *output biner* (0 atau 1). Fungsi *undak biner* (*hardlimit*) dirumuskan sebagai berikut :

$$y = \text{hardlim}(a) \quad (2.1)$$

Fungsi undak biner (*hardlim*) dapat dilihat pada gambar 2.5 sebagai berikut:

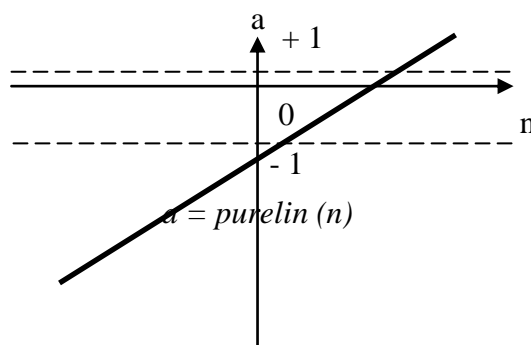


- **Fungsi Linear (Purelin)**

Fungsi *linear* memiliki nilai *output* yang sama dengan nilai inputnya. Fungsi *linear* dirumuskan sebagai berikut :

$$f(x) = x \quad (2.2)$$

Fungsi *linear* (*purelin*) dapat dilihat pada Gambar 2.7 berikut ini :



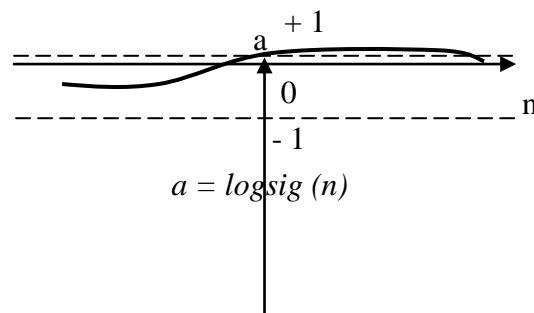
- **Fungsi Sigmoid Biner (logsig)**

Fungsi ini digunakan untuk jaringan saraf yang dilatih menggunakan metode *BackPropagation*. Fungsi *sigmoid biner* memiliki nilai pada range 0 sampai 1. Oleh karena itu, fungsi ini sering digunakan untuk jaringan saraf yang membutuhkan nilai *output* yang terletak pada interval 0 sampai 1. namun fungsi ini juga *output* 0 atau 1. fungsi *sigmoid biner* dirumuskan sebagai :

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

$$\text{dengan: } f'(x) = f(x)(1 - f(x)) \quad (2.3)$$

Fungsi *Sigmoid Biner (logsig)* dapat dilihat Pada gambar 2.8 berikut ini :



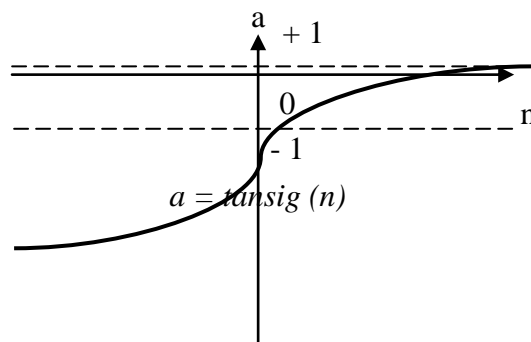
- **Fungsi Sigmoid Bipolar (tansig)**

Fungsi *sigmoid bipolar* hampir sama dengan fungsi *sigmoid biner*, hanya saja *output* dari fungsi ini memiliki range dari 1 sampai -1. Fungsi *sigmoid bipolar* dirumuskan sebagai berikut :

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

$$\text{dengan: } f'(x) = 1 - f^2(x) \quad (2.4)$$

Fungsi *sigmoid bipolar (tansig)* dilihat pada gambar 2.9 berikut ini :

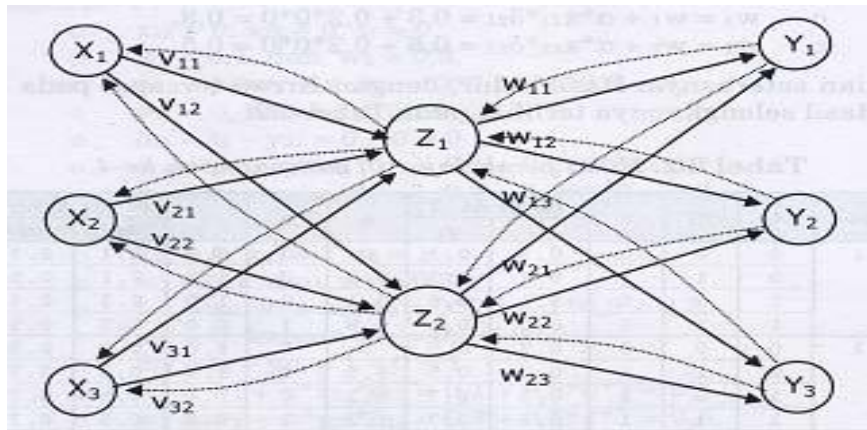


Gambar 2.9 Fungsi Sigmoid Bipolar (tansig)

Backproagation

Backproagation merupakan suatu teknik pembelajaran atau pelatihan *supervised learning* yang paling banyak digunakan. Metode ini merupakan salah satu metode yang sangat baik dalam menangani masalah pengenalan pola-pola kompleks. Metode Jaringan Syaraf Tiruan yang akan digunakan adalah metode jaringan syaraf tiruan propagasi balik. Kusumadewi (2004) menjelaskan, propagasi balik 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.

Arsitektur BackPropagation



Gambar Arsitektur *backpropagation* di atas merupakan jaringan *backpropagation* yang terdapat neuron pada lapisan *input*, lapisan tersembunyi dan lapisan *output*. Sehingga terdapat *edge* dari lapisan *input* ke lapisan tersembunyi dan *edge* dari lapisan tersembunyi ke lapisan *output*. Setiap *edge* memiliki bobot (*weight*). V_{ij} merupakan bobot garis dari unit masukan x_i pada lapisan *input* ke masing-masing unit Z pada lapisan tersembunyi. W merupakan bobot dari unit lapisan tersembunyi Z ke unit keluaran Y .

Nilai masukan pada lapisan *input* dinotasikan sebagai x_1 , x_2 , x_3 . Pada lapisan tersembunyi neuron-neuronnya dinotasikan dengan Z_{11} , Z_{22} . Begitujuga neuron-neuron pada lapisan *output* dinotasikan dengan Y_1 , Y_2 , Y_3 .

Fase-fase Algorithm Backpropagation

- Initialization, Pada fase pertama ini yang harus dilakukan adalah set : set nilai awal untuk variabel-variabel yang diperlukan (nilai input, weight, output yang diharapkan, learning rate α , θ , dan sebagainya).

- Activation, Menghitung aktual output pada hidden layer dengan rumus :

$$Y_j(P) = \text{fungsi} \left[\sum_{i=1}^n x_{ij}(P) \cdot w_{ij}(P) - \theta \right] \quad \dots\dots (2.1)$$

Menghitung aktual output pada layer dengan rumus :

$$Y_k(P) = \text{fungsi} \left[\sum_{j=1}^n x_{jk}(P) \cdot w_{jk}(P) - \theta \right] \quad \dots\dots 2.2)$$

- Weight Training

Menghitung error griend pada output layer

Menghitung koreksi dari pada nilai pemberat (*weight correction*)

$$\Delta k(P) = e_k [1 - Y_k(P)] \quad \dots\dots (2.3)$$

Dimana :
 $\delta k(P)$ = Error griend pada output layer
 $Y_k(P)$ = Actual output pada output layer
 $E(k)$ = Error pada output layer

Menghitung koreksi weight

$$W_{jk}(P+1) = W_{jk}(P) + \Delta W_{jk}(P) \quad \dots\dots (2.4)$$

Dimana :

$$\Delta W_{jk}(P) = \alpha \cdot Y_j(P) \cdot \delta k(P)$$

α = Learning rate

Y_j = Actual output hidden

Menghitung error griend pada hidden layer

$$\Delta j(P) = Y_j(P) \cdot [1 - Y_j(P)] \cdot \sum_{k=1}^l \delta k(P) \cdot W_{jk}(P) \quad \dots\dots (2.5)$$

Menghitung nilai pemberat (*weight correction*)

$$W_{ij}(P+1) = W_{ij}(P) + \Delta W_{ij}(P)$$

Dimana :

$$\Delta w_{ij}(P) = \alpha \cdot X_i(P) \cdot \delta j(P)$$

α = Learning rate

X_i = Actual output hidden

Pengertian Prediksi

Prediksi adalah memperkirakan sesuatu yang akan terjadi di masa yang akan datang. Kejadian masa lampau dipelajari untuk menentukan kecondongan atau pola datanya.

Jenis Prediksi

Jenis prediksi berdasarkan horizon perencanaan adalah sebagai berikut:

- Prediksi jangka pendek, Prediksi jangka pendek merupakan waktu prediksi kurang dari tiga bulan
- Prediksi jangka menengah, Prediksi jangka menengah merupakan prediksinya tiga bulan hingga dua tahun.
- Prediksi jangka panjang, Prediksi jangka panjang merupakan jangka waktu prediksi lebih dari dua tahun.

Teknik Prediksi

Terdapat dua kategori umum teknik prediksi yaitu kuantitatif dan kualitatif. Antara lain adalah sebagai berikut :

- Teknik prediksi kuantitatif meliputi *regression analys, exponential smoothing, moving average, life cycle, box-jenskin, trend line analys, docomposition, stright-line, projection, life cycle analys, simulation, neural network*.
- Teknik prediksi kuantitatif, meliputi ; *neural network, jury of executiveopinion, sales force coposite, customer expectations (customer survey), delphi* dan *naive*.

Validasi Prediksi

Error pada prediksi merupakan selisih dari nilai data aktual ($Y(t)$) dengan nilai hasil prdiksi $Y^{\wedge}(t)$.

$$E(t) = Y(t) - Y^{\wedge}(t)$$

Beberapa pengukuran keefektifan suatu teknik prediksi adalah :

Mean Absolute Deviation (MAD) :

$$MAD = \frac{\sum_{t=1}^N [Y(t) - Y^{\wedge}(t)]}{N}$$

Mean Squared Error (MSE) :

$$MSE = \frac{\sum_{i=1}^N (Y(t) - \hat{Y}(t))^2}{N}$$

Mean Absolute Precen Error (MAP) :

$$MAP = \frac{100}{N} \sum_{t=1}^N \left\| \frac{Y(t) - \hat{Y}(t)}{Y(t)} \right\|$$

MAD dan MSE dapat digunakan untuk mengevaluasi beberapa teknik prediksi atau untuk kondisi parameter yang berbeda-beda pada satu teknik prediksi. Nilai yang lebih kecil merepresentasikan hasil prediksi yang lebih baik di antara hasil-hasil prediksi yang dibandingkan. MAP dapat memberikan evluasi tentang seberapa baik suatu prediksi.

Implementasi

Berikut diberikan contoh perhitungan pelatihan/peramalan dengan Jaringan Syaraf Tiruan pada prediksi tingkat kelulusan siswa, sebagai contoh dilakukan pelatihan menggunakan 2 buah polal input vektor input x , yaitu :

X_1 = Banyak Guru

X_2 = Banyak Lokal

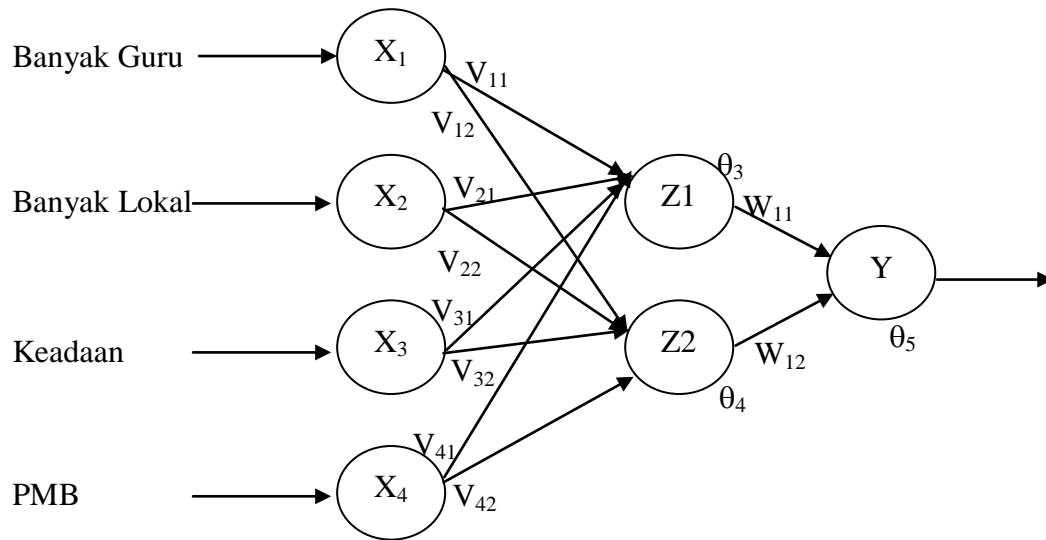
Data input dari jumlah tingkat kelulusan masa lalu adalah sebagai berikut :

x_1	x_2	x_3	x_4
50	26	3	1
53	26	3	1

Data tersebut di skalakan menjadi :

0,1	0,1	0,9	0.1
0,2	0,1	0,9	0.1

Arsitektur yang akan dipilih dimisalkan seperti pada gambar 4.2. Jumlah Simpul pada lapisan input 4 masing-masing variabelnya adalah banyak guru, banyak lokal, keadaan dan PBM. Jumlah simpul pada lapisan tersembunyi (*hidden*) ada 2 (dua). Jumlah simpul pada lapisan output 1 (satu) untuk mempresentasikan pola.



Keterangan :

X = Masukan (input).

$J = 1$ s/d n ($n = 10$).

V = Bobot pada lapisan tersembunyi.

W = Bobot pada lapisan keluaran.

n = Jumlah unit pengolah pada lapisan tersembunyi.

b = Bias pada lapisan tersembunyi dan lapisan keluaran.

k = Jumlah unit pengolah pada lapisan keluaran.

Y = Keluaran hasil.

Untuk membentuk jaringan syaraf tiruan, terlebih dahulu dilakukan inisialisasi bobot awal. Bobot awal yang menghubungkan simpul-simpul pada lapisan input dan lapisan tersembunyi untuk arsitektur di atas adalah $v = (v_{11}, v_{12}, w_{21}, v_{22},)$ dan bobot bias dipilih secara acak. Demikian pula bobot awal yang menghubungkan simpul-simpul pada lapisan tersembunyi dan lapisan output (w_{11}, w_{12}) juga dipilih secara acak.

Tahap 1. Initialitation

Misalkan :

$X_1 = 0,1$ $X_2 = 0,1$ $X_3 = 0,9$ $X_4 = 0,1$

$J = 3$ $K = 5$

$V_{11} = 0,5$ $V_{21} = 0,6$ $V_{31} = 0,4$ $V_{41} = 0,1$

$V_{12} = 0,8$ $V_{22} = 0,7$ $V_{32} = 0,2$ $V_{42} = 0,9$

$W_{11} = 0,9$ $W_{12} = 0,6$

$\theta_3 = 0,8$ $\theta_4 = 0,2$ $\theta_5 = 0,4$ $\alpha = 0,1$ $Y_d = 0$

Tahap 2. Activation (iterasi P= 1)

- Menghitung actual output pada hidden layer

$$Y_j(P) = Fungs_i \left[\sum_{i=1}^n x_i(P) \cdot w_{ij}(P) - \theta_j \right]$$

$$\begin{aligned} Y3(1) &= Sigmoid [x_1(1) \cdot v_{11}(1) + x_2(1) \cdot v_{21}(1) + x_3(1) \cdot v_{31}(1) + x_4(1) \cdot v_{41}(1) - \theta_3] \\ &= Sigmoid [0,1 \cdot 0,5 + 0,1 \cdot 0,6 + 0,9 \cdot 0,4 + 0,1 \cdot 0,1 - 0,8] \\ &= Sigmoid [0,05 + 0,06 + 0,36 + 0,01 - 0,8] \\ &= Sigmoid [-0,32] \rightarrow \frac{1}{1 + e^{(0,32)}} = \frac{1}{2,3771} = 0,4207 \end{aligned}$$

$$\begin{aligned} Y4(1) &= Sigmoid [x_1(1) \cdot v_{12}(1) + x_2(1) \cdot v_{22}(1) + x_3(1) \cdot v_{32}(1) + x_4(1) \cdot v_{42}(1) - \theta_4] \\ &= Sigmoid [0,1 \cdot 0,6 + 0,1 \cdot 0,7 + 0,9 \cdot 0,2 + 0,1 \cdot 0,9(-0,1)] \\ &= Sigmoid [0,06 + 0,07] + 0,18 + 0,09 \\ &= Sigmoid [0,20] \rightarrow \frac{1}{1 + e^{-(0,20)}} = \frac{1}{1,8187} = 0,5498 \end{aligned}$$

Menghitung actual output pada output layer

$$Y_k(P) = Fungs_i \left[\sum_{j=1}^m x_{jb}(p) \cdot w_{jk}(p) - \theta_k \right]$$

$$\begin{aligned} Y5(1) &= Sigmoid [Y_3 \cdot w_{11} + Y_4 \cdot w_{12} - \theta_5] \\ &= Sigmoid [0,4 \cdot 0,9 + 0,5 \cdot 0,6 - 0,4] \end{aligned}$$

$$= Sigmoid [0,3 + 0,4 - 0,4]$$

$$= Sigmoid [0,7 - 0,4]$$

$$= 0,3085 \text{ Sigmoid } [0,3085] \rightarrow \frac{1}{1 + e^{-(0,3085)}} = \frac{1}{1,7345} = 0,5765$$

Menghitung error:

$$\begin{aligned} e(5) &= Y_d * Y5 - Y5 \\ &= 0 * 0,3085 - 0,3085 \\ e(5) &= -0,3085 \end{aligned}$$

Tahap 3: Weight training

- Menghitung error gradient pada output layer

$$\delta k(p) = y_k(p) \cdot x [1 - y_k(p)] \cdot e_k(p)$$

$$\begin{aligned} \delta 5(1) &= y5(1) \cdot x [1 - y5(1)] \cdot e5(1) \\ &= 0,30 \cdot [1 - (0,3)] \cdot -0,3085 \\ &= -0,0658 \end{aligned}$$

Menghitung koreksi nilai weight:

$$W_{jk}(P+1) = W_{jk}(P) + \Delta w_{jk}(P)$$

$$\Delta w_{jk}(P) = \alpha * Y_j(P) * \delta_k(P)$$

$$J = 3$$

$$\begin{aligned}\Delta w_{11}(1) &= \alpha \cdot y_3(1) - \delta_5(1) \\ &= 0,1 \cdot 0,4(-0,0658) \\ &= 0,1079\end{aligned}$$

$$J = 4$$

$$\begin{aligned}\Delta w_{12}(1) &= \alpha \cdot y_4(1) - \delta_5(1) \\ &= 0,1 \cdot 0,5 - (-0,0658) \\ &= 0,1208\end{aligned}$$

$$\begin{aligned}\Delta \theta_5 &= \alpha \cdot (-1) \cdot \delta_5(1) \\ &= 0,1(-1) \cdot -0,0658 \\ &= -0,0065\end{aligned}$$

$$\begin{aligned}w_{11}(2) &= w_{11}(1) + \Delta w_{11}(1) \\ &= 0,9 + 0,1079 \\ &= -0,0078\end{aligned}$$

$$\begin{aligned}w_{12}(2) &= w_{12}(1) + \Delta w_{12}(1) \\ &= 0,6 + 0,1208 \\ &= 0,7208\end{aligned}$$

$$\delta_5 = \delta_5 + \Delta \delta_5 = 0,4 + -0,0658 = 0,3342$$

- Menghitung error gradient pada hidden layer

$$\delta_j(P) = Y_j(P) \cdot [1 - Y_j(P)] \sum \delta_k(P) \cdot w_{jk}(P)$$

$$J = 3$$

$$\begin{aligned}\delta_3(1) &= Y_3(1) \cdot [1 - Y_3(1)] \sum \delta_5(P) \cdot w_{11}(P) \\ \delta_3(1) &= 0,4206 \cdot [1 - 0,4206] - 0,3085 \cdot (0,9) \\ &= 0,2437 - (-0,2777) \\ &= 0,5214\end{aligned}$$

$$J = 4$$

$$\begin{aligned}\delta_4(1) &= Y_4(1) \cdot [1 - Y_4(1)] \sum \delta_5(P) \cdot w_{12}(P) \\ &= 0,5498 \cdot [1 - 0,5498] - 0,3085 \cdot 0,6 \\ &= 0,02475 - 0,1851 \\ &= -0,4326\end{aligned}$$

Koreksi nilai bobot:

$$\Delta v_{11} = \alpha \cdot x_1(1) \cdot \delta_3(1) = 0,1 \cdot 0,1 \cdot 0,5214 = 0,0052$$

$$\Delta v_{12} = \alpha \cdot x_1(1) \cdot \delta_4(1) = 0,1 \cdot 0,1 \cdot -0,4326 = -0,0043$$

$$\Delta v_{21} = \alpha \cdot x_2(1) \cdot \delta_3(1) = 0,1 \cdot 0,1 \cdot 0,5214 = 0,0052$$

$$\Delta v_{22} = \alpha \cdot x_2(1) \cdot \delta_4(1) = 0,1 \cdot 0,1 \cdot -0,4326 = -0,0043$$

$$\begin{aligned}
\Delta v_{31} &= \alpha \cdot x_3(1) \cdot \delta_3(1) = 0,1 \cdot 0,095214 = 0,0095214 \\
\Delta v_{32} &= \alpha \cdot x_3(1) \cdot \delta_4(1) = 0,1 \cdot 0,094326 = 0,0094326 \\
\Delta v_{41} &= \alpha \cdot x_4(1) \cdot \delta_3(1) = 0,1 \cdot 0,105214 = 0,0105214 \\
\Delta v_{42} &= \alpha \cdot x_4(1) \cdot \delta_4(1) = 0,1 \cdot 0,104326 = 0,0104326 \\
\Delta \theta_3 &= \alpha \cdot (-1) \cdot \delta_3(1) = 0,1 \cdot (-1) \cdot 0,05214 = -0,005214 \\
\Delta \theta_4 &= \alpha \cdot (-1) \cdot \delta_4(1) = 0,1 \cdot (-1) \cdot 0,05214 = -0,005214 \\
\Delta v_{11}(2) &= v_{11}(1) + \Delta v_{11} = 0,5 + 0,0052 = 0,5052 \\
\Delta v_{12}(2) &= v_{12}(1) + \Delta v_{12} = 0,6 + 0,0052 = 0,6052 \\
\Delta v_{21}(2) &= v_{21}(1) + \Delta v_{21} = 0,6 + 0,0052 = 0,6052 \\
\Delta v_{22}(2) &= v_{22}(1) + \Delta v_{22} = 0,7 + 0,0043 = 0,7043 \\
\Delta v_{31}(2) &= v_{31}(1) + \Delta v_{31} = 0,4 + 0,0095 = 0,4095 \\
\Delta v_{32}(2) &= v_{32}(1) + \Delta v_{32} = 0,2 + 0,0094 = 0,2094 \\
\Delta v_{41}(2) &= v_{41}(1) + \Delta v_{41} = 0,1 + 0,0105 = 0,1105 \\
\Delta v_{42}(2) &= v_{42}(1) + \Delta v_{42} = 0,9 + 0,0104 = 0,9104 \\
\theta_3(2) &= \theta_3(1) + \Delta \theta_3 = 0,8 + 0,05214 = 0,85214 \\
\theta_4(2) &= \theta_4(1) + \Delta \theta_4 = -0,2 + 0,05214 = -0,14786
\end{aligned}$$

SIMPULAN

Simpulan yang dapat diambil dari pengujian jaringan syaraf tiruan untuk prediksi tingkat kelulusan siswa adalah jaringan syaraf tiruan dengan *Algorithm Backgropation* adalah sebagai berikut :

1. Pengujian dilakukan dengan mengunaka sistem jaringan syaraf tiruan dengan metode *Algorithm Backgropation* untuk memprediksi tingkat kelulusan siswa
2. Keakurasian dan ketepatan dalam menggunakan jaringan syaraf tiruan tergantung pada data yang akan diuji, semakin banyak data yang diuji maka tingkat keakurasian dan ketepatan data dapat tepat.
3. Pada saat melakukan pengujian hasil yang didapat selalu *epoch* 100, yang menunjukkan bahwa setiap tahun tingkat kelulusan akan naik untuk 2 (dua) tahun akan datang.

Berdasarkan pengujian jaringan syaraf tiruan dengan *Algorithm Backgropagation*, disarankan kepada semua pihak yang berwenang dalam menangani masalah memberikan pengetahuan-pengetahuan yang berhubungan dengan tingkat kelulusan siswa dan pendidikan agar nantinya diharapkan akan selalu meningkat. Dengan memberikan pengetahuan tentang faktor-faktor apa saja

yang mempengaruhi tingkat kelulusan tersebut. Sehingga setelah melakukan prediksi nantinya akan melakukan perancangan atau planning untuk selanjutnya agar menjadi yang lebih baik.

DAFTAR PUSTAKA

- Jong Jek Siang. (2005). Jaringan Syaraf Tiruan dan pemrogramannya menggunakan MATLAB. ANDI, Yogyakarta.
- Purwanto. (2008). Sistem Pengenalan Huruf Tulisan Tangan dengan Pendekatan Heuristik. <http://rmsui.vlsm.org/fusilkom-ui/fusilkom-96-196abs.html>
- Yani, Eli. (2005). Pengantar Jaringan syaraf Tiruan. Artikel Makalah. http://trirezqiariantoro.files.wordpress.com/2007/05/jaringan_syara
- Nam, Ng Ching (2003). Ramalan Permintaan Perumahan Kos Rendah Di Kawasan Bandar Mggk-nn - Satu Pendekatan Siri Masa. www.efka.utm.my/thesis/images/3PSM/2003/5IT/NGCHINNAMCA010029D03TT1.doc.
- Halim, Siana, dan Adrian Michael Wibisono. 2000. "Penerapan Jaringan Syaraf Tiruan untuk Peramalan". Dalam *Jurnal Teknik Industri Vol.2*. No. 2. Hal. 106-113.
- Arief Heru Kuncoro (2005). Aplikasi Jaringan Syaraf Tiruan Untuk Peramalan Beban Tenaga Listrik Jangka Panjang Pada Sistem Kelistrikan Di Indonesia. Jurnal Teknologi
- Wonosobo (2009). Pengantar jaringan syaraf tiruan teori dan aplikasi. Star Publishing