

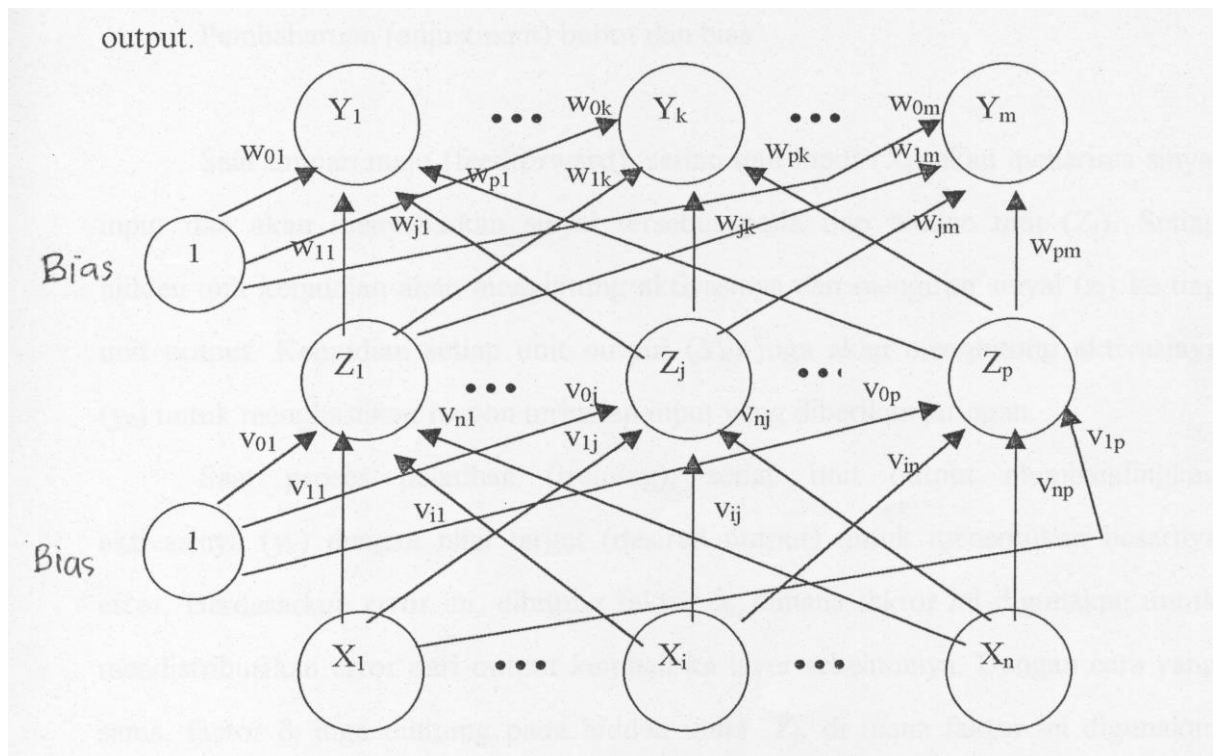
Nama : Daffa Asyqar Ahmad Khalisheka

NIM : 1103200034

Kelas : TK-44-G4

Backpropagation

- Arsitektur Jaringan
 - Salah satu metode pelatihan terawasi pada jaringan syaraf adalah metode Backpropagation, di mana ciri dari metode ini adalah meminimalkan error pada output yang dihasilkan oleh jaringan.
 - Dalam metode Backpropagation, biasanya digunakan jaringan multilayer.
 - Sebagai contoh pada gambar berikut diberikan jaringan dengan sebuah hidden layer.



- Pada gambar, unit input dilambangkan dengan X, hidden unit dilambangkan dengan Z, dan unit output dilambangkan dengan Y.
 - Bobot antara X dan Z dilambangkan dengan v sedangkan bobot antara Z dan Y dilambangkan dengan w.
- Algoritma Metode Backpropagation
 - Pada intinya, pelatihan dengan metode backpropagation terdiri dari tiga langkah, yaitu:

1. Data dimasukkan ke input jaringan (feedforward)
 2. Perhitungan dan propagasi balik dari error yang bersangkutan
 3. Pembaharuan (adjustment) bobot dan bias.
- Saat umpan maju (feedforward), setiap unit input (X_i) akan menerima sinyal input dan akan menyebarkan sinyal tersebut pada tiap hidden unit (Z_j).
 - Setiap hidden unit kemudian akan menghitung aktivasinya dan mengirim sinyal (z_j) ke tiap unit output.
 - Kemudian setiap unit output (Y_k) juga akan menghitung aktivasinya (y_k) untuk menghasilkan respons terhadap input yang diberikan jaringan.
 - Saat proses pelatihan (training), setiap unit output membandingkan aktivasinya (y_k) dengan nilai target (t_k) untuk menentukan besarnya error.
 - Berdasarkan error ini, dihitung faktor δ_k , di mana faktor ini digunakan untuk mendistribusikan error dari output kembali ke layer sebelumnya.
 - Dengan cara yang sama, faktor δ_j juga dihitung pada hidden unit Z_j , di mana faktor ini digunakan untuk memperbaharui bobot antara hidden layer dan input layer.
 - Setelah semua faktor δ ditentukan, bobot untuk semua layer diperbaharui.
- Algoritma Metode Backpropagation secara detail
 - Step 0 : Inisialisasi bobot dan bias
Baik bobot maupun bias dapat diset dengan sembarang angka (acak) dan biasanya angka di sekitar 0 dan 1 atau -1 (bias positif atau negatif)
 - Step 1 : Jika stopping condition masih belum terpenuhi, jalankan step 2-9.
 - Step 2 : Untuk setiap data training, lakukan step 3-8.
 - Umpan maju (feedforward)
 1. Step 3 : Setiap unit input ($X_i, i=1, \dots, n$) menerima sinyal input x_i dan menyebarkan sinyal tersebut pada seluruh unit pada hidden layer. Perlu diketahui bahwa input x_i yang dipakai di sini adalah input training data yang sudah diskalakan.
 2. Step 4 : Setiap hidden unit ($Z_j, j=1, \dots, p$) akan menjumlahkan sinyal-sinyal input yang sudah berbobot, termasuk biasnya

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

dan memakai fungsi aktivasi yang telah ditentukan untuk menghitung sinyal output dari hidden unit yang bersangkutan,

$$z_j = f(z_in_j)$$

lalu mengirim sinyal output ini ke seluruh unit pada unit output

3. Step 5 : Setiap unit output ($Y_k, k=1, \dots, m$) akan menjumlahkan sinyal-sinyal input yang sudah berbobot, termasuk biasnya,

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

dan memakai fungsi aktivasi yang telah ditentukan untuk menghitung sinyal output dari unit output yang bersangkutan:

$$y_k = f(y_in_k)$$

Propagasi error (backpropagation of error)

4. Step 6 : Setiap unit output ($Y_k, k=1, \dots, m$) menerima suatu target pattern (desired output) yang sesuai dengan input training pattern untuk menghitung kesalahan (error) antara target dengan output yang dihasilkan jaringan:

$$\delta_k = (t_k - y_k) f'(y_in_k)$$

- Faktor δ_k ini digunakan untuk menghitung koreksi error (Δw_{jk}) yang nantinya akan dipakai untuk memperbaharui w_{jk} , di mana:

$$\Delta w_{jk} = \alpha \delta_k z_j$$
- Selain itu juga dihitung koreksi bias Δw_{0k} yang nantinya akan dipakai untuk memperbaharui w_{0k} , di mana:

$$\Delta w_{0k} = \alpha \delta_k$$

- Faktor δ_k ini kemudian dikirimkan ke layer yang berada pada step 7.

5. Step 7 : Setiap hidden unit ($Z_j, j=1, \dots, p$) menjumlah input delta (yang dikirim dari layer step 6) yang sudah berbobot.

$$\delta_in_j = \sum_{k=1}^m \delta_k w_{jk}$$

- Kemudian hasilnya dikalikan dengan turunan dari fungsi aktivasi yang digunakan jaringan untuk menghasilkan faktor koreksi error δ_j , di mana:

$$\delta_j = \delta_in_j f'(z_in_j)$$

- Faktor δ_j ini digunakan untuk menghitung koreksi error (Δv_{ij}) yang nantinya akan dipakai untuk memperbaharui v_{ij} , di mana:

$$\Delta v_{ij} = \alpha \delta_j x_i$$

- Selain itu juga dihitung koreksi bias Δv_{0j} yang nantinya akan dipakai untuk memperbaharui v_{0j} , di mana:

$$\Delta v_{0j} = \alpha \delta_j$$

Pembaharuan bobot dan bias:

6. Step 8 :

Setiap unit output ($Y_k, k=1, \dots, m$) akan memperbaharui bias dan bobotnya dari setiap hidden unit.

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

Demikian pula untuk setiap hidden unit akan memperbaharui bias dan bobotnya dari setiap unit input.

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

7. Step 9 : Memeriksa stopping condition

Jika stop condition telah terpenuhi, maka pelatihan jaringan dapat dihentikan.

- Untuk menentukan stopping condition terdapat dua cara yang biasa dipakai, yaitu:

Membatasi iterasi yang ingin dilakukan.

- Misalnya jaringan akan dilatih sampai iterasi yang ke-500.
- Yang dimaksud dengan satu iterasi adalah perulangan step 3 sampai step 8 untuk semua training data yang ada.

Membatasi error.

- Misalnya menentukan besar Mean Square Error antara output yang dikehendaki dan output yang dihasilkan oleh jaringan.
- Jika terdapat sebanyak m training data, maka untuk menghitung Mean Square Error digunakan persamaan berikut:

$$MSE = 0,5 \times \{(t_{k1} - y_{k1})^2 + (t_{k2} - y_{k2})^2 + \dots + (t_{km} - y_{km})^2\}$$

- Setelah pelatihan selesai, jika kemudian jaringan diberi input, jaringan akan menghasilkan output seperti yang diharapkan.
- Cara mendapatkan output adalah dengan mengimplementasikan metode backpropagation, tetapi hanya pada bagian umpan majunya saja, yaitu dengan langkah-langkah sebagai berikut:
- Step 0: Inisialisasi bobot sesuai dengan bobot yang telah dihasilkan pada proses pelatihan di atas.
- Step 1: Untuk setiap input, lakukan step 2-4.
- Step 2: Untuk setiap input $i=1, \dots, n$ skalakan bilangan dalam range fungsi aktivasi seperti yang dilakukan pada proses pelatihan di atas.
- Step 3: untuk $j=1, \dots, p$:

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

$$z_j = f(z_in_j)$$

- Step 4 : Untuk $k=1, \dots, m$:

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

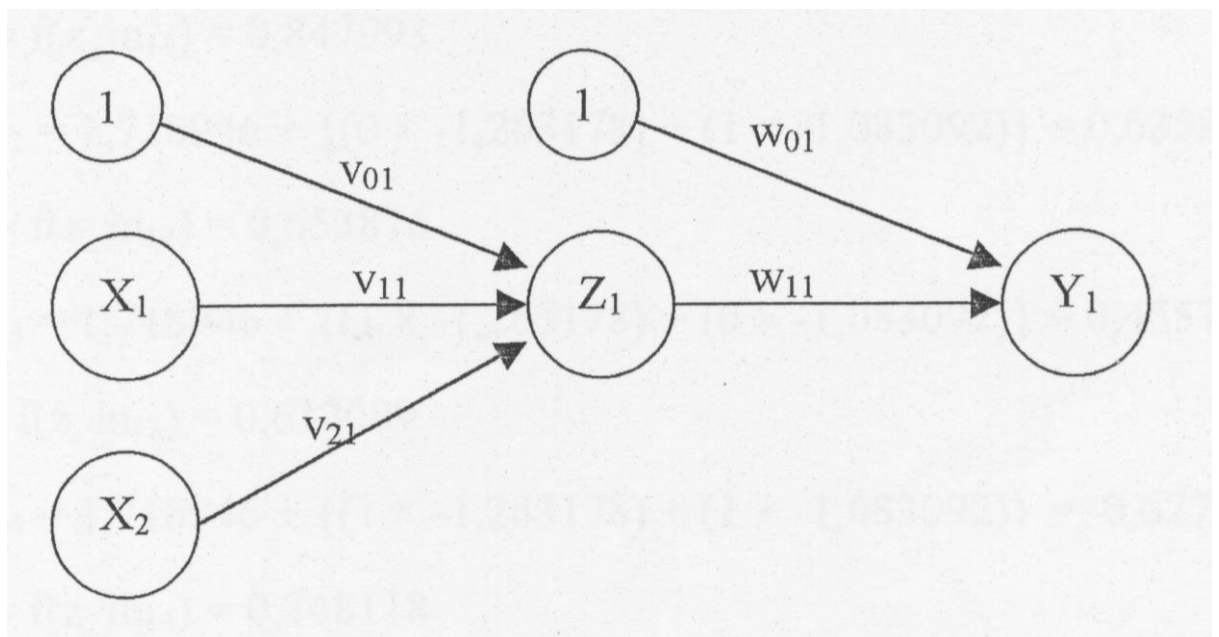
$$y_k = f(y_in_k)$$

- Variabel y_k adalah output yang masih dalam skala menurut range fungsi aktivasi.
- Untuk mendapatkan nilai output yang sesungguhnya, y_k harus dikembalikan seperti semula.
- Contoh aplikasi
 - Berikut ini adalah contoh cara pelatihan jaringan syaraf tiruan sederhana menggunakan metode Backpropagation.
 - Misalkan, jaringan terdiri dari 2 unit input, 1 hidden unit (dengan 1 hidden layer), dan 1 unit output
 - Jaringan akan dilatih untuk memecahkan fungsi XOR.
 - Fungsi aktivasi yang digunakan adalah sigmoid biner dan learning rate (α) = 0,01.

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

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

- Arsitektur jaringan yang akan dilatih adalah sebagai berikut:



- Training data yang digunakan terdiri dari 4 pasang input-output, yaitu:

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	0

- Sebelum pelatihan, harus ditentukan terlebih dahulu stopping conditionnya.
- Misalnya dihentikan jika error telah mencapai 0,41.

- Langkah-langkah pelatihan

- Step 0: Misalnya inisialisasi bobot dan bias adalah:

$$v_{01}=1,718946$$

$$v_{11}=-1,263178$$

$$v_{21}=-1,083092$$

$$w_{01}=-0,541180$$

$$w_{11}=0,543960$$

- Step 1: Dengan bobot di atas, tentukan error untuk training data secara keseluruhan dengan Mean Square Error:

- $z_{in11}=1,718946+{(0 \times -1,263178)+(0 \times -1,083092)}=1,718946$

- $z_{11}=f(z_{in11})=0,847993$

- $z_{in12}=1,718946+{(0 \times -1,263178)+(1 \times -1,083092)}=0,635854$

- $z_{12}=f(z_{in12})=0,653816$

- $z_{in13}=1,718946+{(1 \times -1,263178)+(0 \times -1,083092)}=0,455768$

- $z_{13}=f(z_{in13})=0,612009$

- $z_{in14}=1,718946+{(1 \times -1,263178)+(1 \times -1,083092)}=-0,627324$

- $z_{14}=f(z_{in14})=0,348118$

- di mana indeks z_{jn} berarti hidden unit ke-j dan training data ke-n.

- $y_{in11}=-0,541180+(0,847993 \times 0,543960)=0,079906$

- $y_{11}=f(y_{in11})=0,480034$

- $y_{in12}=-0,541180+(0,653816 \times 0,543960)=-0,185530$

- $y_{12}=f(y_{in12})=0,453750$

- $y_{in13} = -0,541180 + (0,612009 \times 0,543960) = 0,208271$
- $y_{13} = f(y_{in13}) = 0,448119$
- $y_{in14} = -0,541180 + (0,348118 \times 0,543960) = -0,351818$
- $y_{14} = f(y_{in14}) = 0,412941$
- Maka $E = 0,5 \times \{(0 - 0,480034)^2 + (1 - 0,453750)^2 + (1 - 0,448119)^2 + (0 - 0,412941)^2\} = 0,501957$
- Step 2. Karena error masih lebih besar dari 0,41 maka step 3-8 dijalankan.
- Step 3. $x_1 = 0$; $x_2 = 0$ (iterasi pertama, training data pertama)
- Step 4.
 - $z_{in1} = 1,718946 + \{(0 \times -1,263126) + (0 \times -1,083049)\} = 1,718946$.
 - $z_1 = f(z_{in1}) = 0,847993$
- Step 5.
 - $y_{in11} = -0,541180 + (0,847993 \times 0,543960) = 0,079906$
 - $y_{11} = f(y_{in11}) = 0,480034$
- Step 6.
 - $\delta_1 = (0 - 0,480034) f'(0,079906) = -0,119817$
 - $\Delta w_{11} = 0,01 \times -0,119817 \times 0,847993 = -0,001016$
 - $\Delta w_{01} = 0,01 \times -0,119817 = -0,00119817$
- Step 7.
 - $\delta_{in1} = -0,00119817 \times 0,543960 = -0,00065176$
 - $\delta_1 = -0,00065176 \times f'(1,718946) = -0,00008401$
 - $\Delta v_{11} = 0,01 \times -0,00008401 \times 0 = 0$
 - $\Delta v_{21} = 0,01 \times -0,00008401 \times 0 = 0$
 - $\Delta v_{01} = 0,01 \times -0,00008401 = -0,0000008401$
- Step 8.
 - $w_{01}(\text{baru}) = -0,541180 + (-0,00119817) = -0,542378$
 - $w_{11}(\text{baru}) = 0,543960 + (-0,001016) = 0,542944$
 - $v_{01}(\text{baru}) = 1,718946 + (-0,0000008401) = 1,718862$
 - $v_{11}(\text{baru}) = -1,263178 + 0 = -1,263178$
 - $v_{21}(\text{baru}) = -1,083092 + 0 = -1,083092$
 - Saat ini v_{11} dan v_{12} masih belum berubah karena kedua inputnya = 0. Nilai v_{01} dan v_{02} baru berubah pada iterasi pertama untuk training data yang kedua
- Setelah step 3-8 untuk training data pertama dijalankan, selanjutnya kembali lagi ke step 3 untuk training data yang kedua ($x_1 = 0$ dan $x_2 = 1$).
- Langkah yang sama dilakukan sampai pada training data yang keempat.

- Bobot yang dihasilkan pada iterasi pertama, training data ke-2,3, dan 4 adalah:
- Training data ke-2:
 - $w_{01} = -0,541023$
 - $w_{11} = 0,543830$
 - $v_{01} = 1,718862$
 - $v_{11} = -1,263178$
 - $v_{21} = -1,083092$
- Training data ke-3:
 - $w_{01} = -0,539659$
 - $w_{11} = 0,544665$
 - $v_{01} = 1,719205$
 - $v_{11} = -1,263002$
 - $v_{21} = -1,082925$
- Training data ke-4:
 - $w_{01} = -0,540661$
 - $w_{11} = 0,544316$
 - $v_{01} = 1,719081$
 - $v_{11} = -1,263126$
 - $v_{21} = -1,083049$
- Setelah sampai pada training data ke-4, maka iterasi pertama selesai.
- Berikutnya, pelatihan sampai pada step9, yaitu memeriksa stopping condition dan kembali pada step 2.
- Demikian seterusnya sampai stopping condition yang ditentukan terpenuhi.
- Setelah pelatihan selesai, bobot yang didapatkan adalah:
 - $v_{01} = 12,719601$
 - $v_{11} = -6,779127$
 - $v_{21} = -6,779127$
 - $w_{01} = -5,018457$
 - $w_{11} = 5,719889$
- Jika ada input baru, misalnya $x_1 = 0,2$ dan $x_2 = 0,9$ maka outputnya dapat dicari dengan langkah umpan maju sebagai berikut:
- Step 0. Bobot yang dipakai adalah bobot hasil pelatihan di atas.
- Step 1. Perhitungan dilakukan pada step 2-4

- Step 2. Dalam contoh ini, bilangan telah berada dalam interval 0 sampai dengan 1, jadi tidak perlu diskalakan lagi.
- Step 3.
 - $z_{in1} = 12,719601 + \{(0,2x - 6,779127) + (0,9x - 6,779127)\} = 5,262561$
 - $z_1 = f(5,262561) = 0,994845$
- Step 4.
 - $y_{in1} = -5,018457 + (0,994845 \times 5,719889) = 0,671944$
 - $y_1 = f(0,671944) = 0,661938$
- Jadi jika input $x_1 = 0,2$ dan $x_2 = 0,9$; output yang dihasilkan jaringan adalah 0,661938

–