

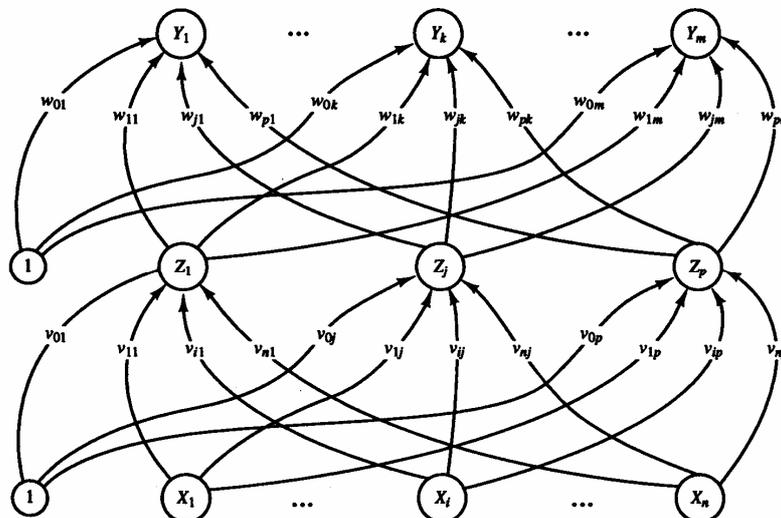
BAB VIIB

BACKPROPAGATION dan CONTOH

7B.1 Standar Backpropagation (BP)

- Backpropagation (BP) merupakan JST multi-layer. Penemuannya mengatasi kelemahan JST dengan layer tunggal yang mengakibatkan perkembangan JST sempat tersendat disekitar tahun 1970. Algoritma BP merupakan generalisasi aturan delta (Widrow-Hoff), yaitu menerapkan metode *gradient descent* untuk meminimalkan error kuadrat total dari keluaran yang dihitung oleh jaringan.
- Banyak aplikasi yang dapat diselesaikan dengan BP, akibatnya JST semakin banyak diminati orang.
- JST layer tunggal memiliki kelemahan dalam pengenalan pola. Hal ini di atasi dengan menambah satu/beberapa layer tersembunyi diantara layar masukan dan layer keluaran. Banyak layar tersembunyi memiliki kelebihan manfaat untuk beberapa kasus, namun pelatihannya memerlukan waktu yang lama.
- Sesuai dengan ide dasar JST, BP melatih jaringan untuk memperoleh keseimbangan antara “kemampuan jaringan” untuk mengenali pola yang digunakan selama pelatihan dan “kemampuan jaringan” merespon secara benar terhadap pola masukan yang serupa (tapi tidak sama) dengan pola pelatihan.

7B.1.1 Arsitektur BP



Jaringan BP dengan satu layer tersembunyi (unit Z ($Z_1 \dots Z_p$)) tampak pada gambar di atas. Unit keluaran (unit Y ($Y_1 \dots Y_m$)) dan unit tersembunyi memiliki bias.

Bobot bias pada unit keluaran Y_k dinyatakan dengan w_{0k} ,

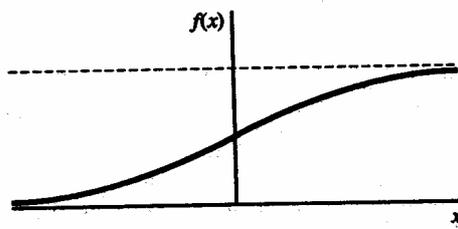
Bobot bias pada unit tersembunyi Z_j dinyatakan dengan v_{0j} .

v_{ij} merupakan bobot garis dari unit X_i ke unit layer tersembunyi Z_j .

w_{jk} merupakan bobot garis dari Z_j ke unit keluaran Y_k

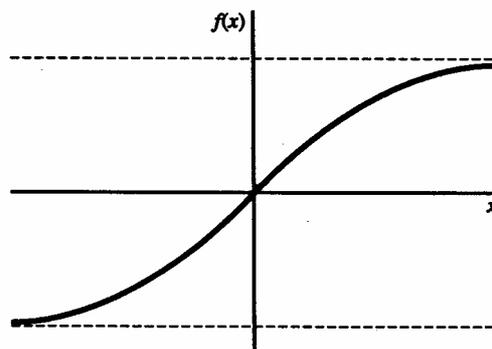
7B.1.2 Fungsi Aktivasi

Dalam BP, fungsi aktivasi yang dipakai harus memenuhi beberapa syarat yaitu: kontinu, dapat didiferensiasi, dan secara monoton tidak menurun. Salah satu fungsi yang sesuai adalah fungsi Sigmoid biner dengan range (0,1).



$$f(x) = 1/(1 + e^{-x}) \text{ dengan turunannya } f'(x) = f(x) (1 - f(x))$$

Fungsi lainnya adalah fungsi Sigmoid bipolar dengan range (-1,1)



$$f(x) = (2/(1 + e^{-x})) - 1 \text{ dengan turunannya } f'(x) = \frac{1}{2} (1+f(x))(1- f(x))$$

Fungsi Sigmoid memiliki nilai maksimum = 1. Maka untuk pola yang memiliki target > 1 , pola masukan dan keluaran harus ditransformasi terlebih dahulu sehingga semua polanya memiliki range yang sama seperti fungsi Sigmoid yang dipakai. Alternatif lain adalah menggunakan fungsi Sigmoid hanya pada layer yang bukan keluaran. Pada layer keluaran menggunakan fungsi identitas $f(x)=x$.

7B.1.3 Algoritma Pelatihan

Pelatihan BP meliputi 3 fase: i) fase propagasi maju (feedforward) pola pelatihan masukan. Pola masukan dihitung maju mulai dari layer masukan hingga layer keluaran dengan fungsi aktivasi yang ditentukan; ii) fase propasi mundur (backpropagation) dari error yang terkait. Selisih antara keluaran dan target merupakan kesalahan yang terjadi. Kesalahan tersebut dipropagasi mundur, dimulai dari garis yang berhubungan langsung dengan unit-unit dilayar keluaran; iii) fase modifikasi bobot.

i) Fase propagasi maju

Selama propagasi maju, sinyal masukan (x_1) dipropagasikan ke layer tersembunyi menggunakan fungsi aktivasi yang ditentukan. Keluaran dari unit tersembunyi (Z_j) tersebut selanjutnya dipropagasi maju lagi ke layer tersembunyi berikutnya dengan fungsi aktivasi yang telah ditentukan. Dan seterusnya hingga menghasilkan keluaran jaringan (y_k).

Berikutnya, keluaran jaringan (y_k) dibandingkan dengan target yang harus dicapai (t_k). Selisih $t_k - y_k$ adalah kesalahan yang terjadi. Jika kesalahan ini lebih kecil dari batas toleransi yang ditentukan, maka iterasi dihentikan. Jika kesalahan masih lebih besar dari batas toleransi, maka bobot setiap garis dari jaringan akan dimodifikasi untuk mengurangi kesalahan

ii) Fase Propagasi Mundur

Berdasarkan kesalahan $t_k - y_k$ dihitung faktor δ_k ($k= 1, \dots, m$) yang dipakai untuk mendistribusikan kesalahan di unit Y_k ke semua unit tersembunyi yang terhubung langsung dengan Y_k . δ_k juga dipakai untuk mengubah bobot garis yang berhubungan langsung dengan unit keluaran.

Dengan cara yang sama, dihitung faktor δ_j di setiap layer tersembunyi sebagai dasar perubahan bobot semua garis yang berasal dari unit tersembunyi di layer di bawahnya. Dan seterusnya hingga semua faktor δ di unit tersembunyi yang terhubung langsung dengan unit masukan dihitung.

iii) Fase Modifikasi Bobot

Setelah semua faktor δ dihitung, bobot semua garis dimodifikasi bersamaan. Perubahan bobot suatu garis didasarkan atas faktor δ neuron di layer atasnya.

Sebagai contoh, perubahan bobot garis yang menuju ke layer keluaran didasarkan atas δ_k yang ada di unit keluaran

Ketiga fase tersebut diulang-ulang hingga kondisi penghentian dipenuhi. Umumnya kondisi penghentian yang sering dipakai adalah jumlah iterasi atau kesalahan. Iterasi akan dihentikan jika jumlah iterasi yang dilakukan sudah melebihi jumlah maksimum iterasi yang ditetapkan, atau jika kesalahan yang terjadi sudah lebih kecil dari batas toleransi yang ditetapkan.

Algoritma BP dengan satu layer tersembunyi dengan fungsi aktivasi Sigmoid biner adalah sbb:

0. Inisialisasi semua bobot dengan bilangan acak kecil
1. Jika kondisi henti belum terpenuhi, lakukan langkah 2 – 9
2. Untuk setiap pasangan data pelatihan, lakukan langkah 3 – 8

Fase I Propagasi Maju

3. Tiap unit masukan menerima sinyal dan meneruskan ke unit tersembunyi di atasnya
4. Hitung semua keluaran di unit tersembunyi Z_j ($j = 1 \dots p$)

$$Z_{netj} = v_{0j} + \sum_{i=1}^p x_i v_{ij}$$

$$z_j = f(Z_{netj}) = \frac{1}{1 + e^{-z_{netj}}}$$

5. Hitung semua keluaran jaringan di unit Y_k ($k = 1 \dots m$)

$$y_{netk} = w_{0k} + \sum_{j=1}^m z_j w_{jk}$$

$$y_k = f(y_{netk}) = \frac{1}{1 + e^{-y_{netk}}}$$

Fase II Propagasi Mundur

6. Hitung faktor δ unit keluaran berdasarkan kesalahan di setiap unit keluaran Y_k ($k = 1, 2, \dots, m$)

$$\delta_k = (t_k - y_k) f'(y_{netk}) = (t_k - y_k) y_k (1 - y_k)$$

δ_k merupakan kesalahan yang akan dipakai dalam perubahan bobot layer dibawahnya (langkah 7)

Hitung suku perubahan bobot w_{jk} (yang akan dipakai nanti untuk merubah bobot w_{jk}) dengan laju pembelajaran α

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (k= 1,2, \dots, m ; j = 0,1,2, \dots, p)$$

7. Hitung faktor δ unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi Z_j ($j=1\dots p$)

$$\delta_{_net j} = \sum_{k=1}^m \delta_k w_{jk}$$

Faktor δ unit tersembunyi

$$\delta_j = \delta_{_net j} f'(z_{_net j}) = \delta_{_net j} z_j (1 - z_j)$$

Hitung suku perubahan bobot v_{ij} (yang akan dipakai untuk merubah v_{ij})

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (j = 1,2, \dots, p ; i = 0,1,2, \dots, n)$$

Fase III Modifikasi Bobot

8. Hitung semua perubahan bobot

Perubahan bobot garis yang menuju ke unit keluaran:

$$w_{jk} \text{ (baru)} = w_{jk} \text{ (lama)} + \Delta w_{jk} \quad (k=1,2, \dots, m ; j=0,1,2, \dots, p)$$

Perubahan bobot garis yang menuju ke unit tersembunyi:

$$v_{ij} \text{ (baru)} = v_{ij} \text{ (lama)} + \Delta v_{ij} \quad (j=1,2, \dots, p ; i=0,1,2, \dots, n)$$

9. Stop

Setelah pelatihan selesai dilakukan, jaringan dapat digunakan untuk pengenalan pola. Dalam hal ini, hanya **Propagasi Maju** (langkah 4 dan 5) saja yang dipakai untuk menentukan keluaran jaringan.

Apabila fungsi aktivasi yang dipakai bukan Sigmoid biner, maka langkah 4 dan 5 harus disesuaikan. Demikian juga turunannya pada langkah 6 dan 7

Contoh 1

Gunakan BP dengan sebuah layer tersembunyi (dengan 3 unit) untuk mengenali fungsi logika XOR dengan 2 masukan x_1 dan x_2 . Buatlah iterasi untuk menghitung bobot jaringan untuk pola pertama ($x_1 = 1, x_2 = 1, \text{ dan } t = 0$). Gunakan laju pembelajaran $\alpha = 0,2$

Penyelesaian

Gambar seperti gambar di atas dengan hanya satu unit keluaran Y, 3 unit Z, dan 2 unit X serta 2 bias (untuk unit tersembunyi dan unit keluaran

Mula2 bobot diberi nilai acak yang kecil (range [-1,1]). Misalkan bobot v_{ij} seperti tampak pada tabel berikut

	Z ₁	Z ₂	Z ₃
X ₁	0,2	0,3	-0,1
X ₂	0,3	0,1	-0,1
1	-0,3	0,3	0,3

Bobot w_{jk} sbb:

	Y
Z ₁	0,5
Z ₁	-0,3
Z ₁	-0,4
1	-0,1

Langkah 4: hitung keluaran unit tersembunyi (Z_j)

$$Z_{\text{net } j} = v_{0j} + \sum_{i=1}^p x_i v_{ij}$$

$$Z_{\text{net } 1} = -0,3 + 1 (0,2) + 1 (0,3) = 0,2$$

$$Z_{\text{net } 2} = 0,3 + 1 (0,3) + 1 (0,1) = 0,7$$

$$Z_{\text{net } 3} = 0,3 + 1 (-0,1) + 1 (-0,1) = 0,1$$

$$z_j = f(Z_{\text{net } j}) = \frac{1}{1 + e^{-z_{\text{net } j}}}$$

$$z_1 = \frac{1}{1 + e^{-0,2}} = 0,55 ; z_2 = \frac{1}{1 + e^{-0,7}} = 0,67 ; z_3 = \frac{1}{1 + e^{-0,1}} = 0,52$$

Langkah 5: hitung keluaran unit Y_k

$$y_{\text{net } k} = w_{0k} + \sum_{j=1}^m z_j w_{jk}$$

Karena jaringan hanya memiliki sebuah unit keluaran Y, maka $y_{net\ k} = y_{net} = w_{01}$

$$+ \sum_{k=1}^m z_j w_{jk} = -0,1 + 0,55 (0,5) + 0,67 (-0,3) + 0,52 (-0,4) = -0,24$$

$$y = f(y_{net\ k}) = \frac{1}{1 + e^{-y_{net\ k}}} = \frac{1}{1 + e^{0,24}} = 0,44$$

Langkah 6 : Hitung faktor δ di unit keluaran Y_k

$$\delta_k = (t_k - y_k) f'(y_{net\ k}) = (t_k - y_k) y_k (1 - y_k)$$

Karena jaringan hanya memiliki satu keluaran maka

$$\delta_k = (t_k - y_k) f'(y_{net\ k}) = (t - y) y (1 - y) = (0 - 0,44) (0,44) (1 - 0,44) = -0,11$$

Suku perubahan bobot w_{jk} (dengan $\alpha = 0,2$) :

$$\Delta w_{jk} = \alpha \delta_k z_j = \alpha \delta z_j \quad (k=1 ; j = 0,1,2,3)$$

$$\Delta w_{01} = 0,2 (-0,11) (1) = -0,02$$

$$\Delta w_{11} = 0,2 (-0,11) (0,55) = -0,01$$

$$\Delta w_{21} = 0,2 (-0,11) (0,67) = -0,01$$

$$\Delta w_{31} = 0,2 (-0,11) (0,52) = -0,01$$

Langkah 7 : Hitung penjumlahan kesalahan dari unit tersembunyi (δ)

$$\delta_{net\ j} = \sum_{k=1}^m \delta_k w_{jk} . \text{ Karena jaringan hanya memiliki sebuah unit keluaran maka}$$

$$\delta_{net\ j} = \delta w_{j1} \quad (j = 1,2,3)$$

$$\delta_{net\ 1} = (-0,11) (0,5) = -0,05$$

$$\delta_{net\ 2} = (-0,11) (-0,3) = 0,03$$

$$\delta_{net\ 3} = (-0,11) (-0,4) = 0,04$$

Faktor kesalahan δ di unit tersembunyi :

$$\delta_j = \delta_{net\ j} f'(z_{net\ j}) = \delta_{net\ j} z_j (1 - z_j)$$

$$\delta_1 = -0,05 (0,55) (1 - 0,55) = -0,01$$

$$\delta_2 = 0,03 (0,67) (1 - 0,67) = 0,01$$

$$\delta_3 = 0,04 (0,52) (1 - 0,52) = 0,01$$

Suku perubahan bobot ke unit tersembunyi $\Delta v_{ij} = \alpha \delta_j x_i \quad (j = 1,2,3 ; i = 0,1,2)$

	Z ₁	Z ₂	Z ₃
X ₁	$\Delta v_{11} = (0,2) (-0,01)$ (1) = -0,002 ≈ 0	$\Delta v_{12} = (0,2) (0,01)$ (1) = 0,002 ≈ 0	$\Delta v_{13} = (0,2) (0,01)$ (1) = 0,002 ≈ 0
X ₂	$\Delta v_{21} = (0,2) (-0,01)$ (1) = -0,002 ≈ 0	$\Delta v_{22} = (0,2) (0,01)$ (1) = 0,002 ≈ 0	$\Delta v_{23} = (0,2) (0,01)$ (1) = 0,002 ≈ 0
1	$\Delta v_{01} = (0,2) (-0,01)$ (1) = -0,002 ≈ 0	$\Delta v_{02} = (0,2) (0,01)$ (1) = 0,002 ≈ 0	$\Delta v_{03} = (0,2) (0,01)$ (1) = 0,002 ≈ 0

Langkah 8 : Hitung semua perubahan robot

Perubahan bobot unit keluaran:

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (k = 1 ; j = 0,1,2,3)$$

$$w_{11}(\text{baru}) = 0,5 - 0,01 = 0,49$$

$$w_{21}(\text{baru}) = -0,3 - 0,01 = -0,31$$

$$w_{31}(\text{baru}) = -0,4 - 0,01 = -0,41$$

$$w_{01}(\text{baru}) = -0,1 - 0,02 = -0,12$$

Perubahan bobot unit tersembunyi:

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (j=1,2,3 ; i=0,1,2)$$

Tabel 7.4

	Z ₁	Z ₂	Z ₃
X ₁	$v_{11} = 0,2 + 0 = 0,2$	$v_{12} = 0,3 + 0 = 0,3$	$v_{13} = -0,1 + 0 = -0,1$
X ₂	$v_{21} = 0,3 + 0 = 0,3$	$v_{22} = 0,1 + 0 = 0,1$	$v_{23} = -0,1 + 0 = -0,1$
1	$v_{01} = -0,3 + 0 = -0,3$	$v_{02} = 0,3 + 0 = 0,3$	$v_{03} = 0,3 + 0 = 0,3$

Iterasi diteruskan untuk pola kedua (X₁ = 1, X₂ = 0, t = 1) dan seterusnya.

7B.2 Optimalitas Arsitektur Backpropagation (BP)

Masalah utama yang dihadapi dalam BP adalah lamanya iterasi yang harus dilakukan. BP tidak dapat memberikan kepastian tentang berapa epoch yang harus dilalui untuk mencapai kondisi yang diinginkan. Oleh karena itu orang berusaha meneliti bagaimana parameter-parameter jaringan dibuat sehingga menghasilkan jumlah iterasi yang relatif lebih sedikit.

7B.2.1 Pemilihan Bobot dan Bias Awal

Bobot awal akan mempengaruhi apakah jaringan mencapai titik minimum lokal (local minimum) atau global, dan seberapa cepat konvergensinya

Bobot yang menghasilkan nilai turunan aktivasi yang kecil sedapat mungkin dihindari karena akan menyebabkan perubahan bobotnya menjadi sangat kecil. Demikian pula nilai bobot awal tidak boleh terlalu besar karena nilai turunan fungsi aktivasinya menjadi sangat kecil juga. Oleh karena itu dalam “Standar BP”, bobot dan bias diisi dengan bilangan acak kecil.

Nguyen dan Widrow (1990) mengusulkan cara membuat inisialisasi bobot dan bias ke unit tersembunyi sehingga menghasilkan iterasi lebih cepat.

Misalkan, n = jumlah unit masukan, p = jumlah unit tersembunyi, dan β = faktor skala = $0,7 \sqrt{p}$

Algoritma inisialisasi Nguyen-Widrow adalah sbb:

1. Inisialisasi semua bobot (v_{ij} (lama)) dengan bilangan acak dalam interval $[-0,5: 0,5]$
2. Hitung $\|v_j\| = \sqrt{v_{1j}^2 + v_{2j}^2 + \dots + v_{nj}^2}$
3. Bobot yang dipakai sebagai inisialisasi $v_{ij} = \frac{\beta v_{ij}(\text{lama})}{\|v_j\|}$
4. Bias yang dipakai sebagai inisialisasi $v_{0j} =$ bilangan acak antara $-\beta$ dan β

Contoh 2

Buatlah bobot awal ke unit tersembunyi contoh 1 menggunakan algoritma inisialisasi Nguyen-Widrow

Penyelesaian

Pada contoh 1, n = jumlah unit masukan = 2 dan p = jumlah unit tersembunyi =

3. Bobot lama (v_{ij} (lama)) yang didapat secara acak tampak pada tabel

	Z ₁	Z ₂	Z ₃
X ₁	0,2	0,3	-0,1
X ₂	0,3	0,1	-0,1
1	-0,3	0,3	0,3

$\beta = \text{faktor skala} = 0,7 \sqrt[3]{3} = 1,21$

$$\|v_1\| = \sqrt{v_{11}^2 + v_{21}^2} = \sqrt{0,2^2 + 0,3^2} = 0,36$$

$$\|v_2\| = \sqrt{v_{12}^2 + v_{22}^2} = \sqrt{0,3^2 + 0,1^2} = 0,32$$

$$\|v_3\| = \sqrt{v_{13}^2 + v_{23}^2} = \sqrt{(-0,1)^2 + (-0,1)^2} = 0,14$$

Tabel berikut merupakan bobot yang dipakai sebagai insialisasi

	Z ₁	Z ₂	Z ₃
X ₁	(1,21*0,2)/0,36 = 0,67	(1,21*0,3)/0,32 = 1,13	(1,21*-0,1)/0,14 = 0,86
X ₂	(1,21*0,3)/0,36 = 1	(1,21*0,1)/0,32 = 0,38	(1,21*(-0,1))/0,14 = 0,86

Bias yang dipakai adalah bilangan acaka antara -1,21 hingga 1,21

7B.2.2 Jumlah Unit Tersembunyi

Berdasarkan hasil teoritis, BP dengan sebuah layer tersembunyi sudah cukup untuk mampu mengenali sembarang pasangan antara masukan dan target dengan tingkat ketelitian yang ditentukan. Akan tetapi penambahan jumlah layer tersembunyi kadangkal membuat pelatihan lebih mudah.

Jika jaringan memiliki lebih dari layer tersembunyi, maka algoritma pelatihan yang dijabarkan sebelumnya perlu direvisi.

Dalam propagasi maju, keluaran harus dihitung untuk setiap layer, dimulai dari layer tersembunyi paling bawah (terdekat dengan unit masukan). Sebaliknya dalam propagasi mundur, faktor δ perlu dihitung untuk tiap layer tersembunyi, dimulai dari layer keluaran.

7B.2.3 Jumlah Pola Pelatihan

Tidak ada kepastian tentang berapa banyak pola yang diperlukan agar jaringan dapat dilatih dengan sempurna. Jumlah pola yang dibutuhkan dipengaruhi oleh banyaknya bobot dalam jaringan serta tingkat akurasi yang diharapkan. Aturan kasarnya dapat ditentukan berdasarkan persamaan :

$$\text{Jumlah pola} = \text{jumlah bobot} / \text{tingkat akurasi}$$

Untuk jaringan dengan 80 bobot dan tingkat akurasi 0,1, maka 800 pola masukan diharapkan akan mampu dikenali dengan benar 90% pola diantaranya.

7B.2.4 Lama Iterasi

Tujuan utama penggunaan BP adalah mendapatkan keseimbangan antara pengenalan pola pelatihan secara benar dan respon yang baik untuk pola lain yang sejenis (data pengujian). Jaringan dapat dilatih terus menerus hingga semua pola pelatihan dikenali dengan benar. Akan tetapi hal itu tidak menjamin jaringan akan mampu mengenali pola pengujian dengan tepat. Jadi tidaklah bermanfaat untuk meneruskan iterasi hingga semua kesalahan pola pelatihan = 0.

Umumnya data dibagi menjadi dua bagian (saling asing), yaitu pola data pelatihan untuk dan data pengujian. Perubahan bobot dilakukan berdasarkan pola pelatihan. Akan tetapi selama pelatihan (misalkan setiap 10 epoch), kesalahan yang terjadi dihitung berdasarkan semua data (pelatihan dan pengujian). Selama kesalahan ini menurun, pelatihan terus dijalankan. Akan tetapi jika kesalahannya sudah meningkat, pelatihan tidak ada gunanya diteruskan. Jaringan sudah mulai mengambil sifat yang hanya dimiliki secara spesifik oleh data pelatihan (tapi tidak dimiliki oleh data pengujian) dan sudah mulai kehilangan kemampuan melakukan generalisasi.

7B.3 Variasi BP

Di samping model standar BP, kini sudah berkembang berbagai variasinya. Variasi tsb. Dapat berupa model BP yang digunakan untuk keperluan khusus, atau teknik modifikasi bobot untuk mempercepat pelatihan dalam kasus tertentu.

Beberapa variasi diantaranya dengan 1) teknik berdasarkan perubahan momentum (arah gradien pola); 2) aturan delta-bar-delta; 3) pengelompokan perubahan bobot

7B.3.1 Momentum

Pada standar BP, perubahan bobot didasarkan atas gradien yang terjadi untuk pola yang dimasukkan saat itu. Modifikasi yang dapat dilakukan adalah melakukan perubahan bobot yang didasarkan atas "ARAH GRADIEN" pola terakhir dan pola sebelumnya (disebut "momentum") yang dimasukkan. Jadi tidak hanya pola masukan terakhir saja yang diperhitungkan.

Penambahan momentum dimaksudkan untuk menghindari perubahan bobot yang mencolok akibat adanya data yang sangat berbeda dengan yang lain (outlier). Apabila beberapa data terakhir yang diberikan ke jaringan memiliki pola yang serupa (berarti arah gradien sudah benar), maka perubahan bobot dilakukan secara cepat. Namun apabila data terakhir yang dimasukkan memiliki pola yang berbeda dengan pola sebelumnya, maka perubahan bobot dilakukan secara lambat.

Dengan penambahan momentum, bobot baru pada waktu ke $(T + 1)$ didasarkan atas bobot pada waktu T dan $(T-1)$. Di sini harus ditambahkan 2 variabel baru yang mencatat besarnya momentum untuk 2 iterasi terakhir. Jika μ adalah konstanta ($0 \leq \mu \leq 1$) yang menyatakan parameter momentum maka bobot baru dihitung berdasarkan persamaan:

$$w_{jk}(T+1) = w_{jk}(T) + \alpha \delta_k z_j + \mu (w_{jk}(T) - w_{jk}(T-1))$$

dan

$$v_{ij}(T+1) = v_{ij}(T) + \alpha \delta_j x_i + \mu (v_{ij}(T) - v_{ij}(T-1))$$

Contoh 3

Perhatikan kembali iterasi pola pertama fungsi logika XOR dengan BP pada contoh 1. Lakukan iterasi untuk pola kedua ($X_1 = 1, X_2 = 0, t = 1$) dengan menggunakan suku momentum ($\mu = 0,5$)

Penyelesaian

Iterasi untuk pola kedua sebenarnya sama dengan iterasi contoh 1. Hanya saja perhitungan bobot baru pada langkah 8 dilakukan dengan menambah momentum (bobot pada waktu (T-1) = bobot awal)

Hasil iterasi yang diperoleh dari pola pertama

Bobot v_{ij} seperti tampak pada tabel berikut

	Z_1	Z_2	Z_3
X_1	0,2	0,3	-0,1
X_2	0,3	0,1	-0,1
1	-0,3	0,3	0,3

Bobot w_{jk} sbb:

	Y
Z_1	0,49
Z_1	-0,31
Z_1	-0,41
1	-0,12

Langkah 4: hitung keluaran unit tersembunyi (Z_j)

$$Z_{\text{net } j} = v_{0j} + \sum_{i=1}^p x_i v_{ij}$$

$$Z_{\text{net } 1} = -0,3 + 1 (0,2) + 0 (0,3) = -0,1$$

$$Z_{\text{net } 2} = 0,3 + 1 (0,3) + 0 (0,1) = 0,6$$

$$Z_{\text{net } 3} = 0,3 + 1 (-0,1) + 0 (-0,1) = 0,2$$

$$Z_j = f(Z_{\text{net } j}) = \frac{1}{1 + e^{-z_{\text{net } j}}}$$

$$z_1 = \frac{1}{1 + e^{0,1}} = 0,48 ; z_2 = \frac{1}{1 + e^{-0,6}} = 0,65 ; z_3 = \frac{1}{1 + e^{-0,2}} = 0,55$$

Langkah 5: hitung keluaran unit Y_k

$$Y_{\text{net } k} = w_{0k} + \sum_{j=1}^m z_j w_{jk}$$

Karena jaringan hanya memiliki sebuah unit keluaran Y, maka $y_{\text{net } k} = y_{\text{net}} = w_{01}$

$$+ \sum_{k=1}^m z_j w_{jk} = -0,12 + 0,48 (0,49) + 0,65 (-0,31) + 0,55 (-0,41) = -0,31$$

$$y = f(y_{\text{net } k}) = \frac{1}{1 + e^{-y_{\text{net } k}}} = \frac{1}{1 + e^{0,31}} = 0,42$$

Langkah 6 : Hitung faktor δ di unit keluaran Y_k

$$\delta_k = (t_k - y_k) f'(y_{\text{net } k}) = (t_k - y_k) y_k (1 - y_k)$$

Karena jaringan hanya memiliki satu keluaran maka

$$\delta_k = (t_k - y_k) f'(y_{\text{net } k}) = (t - y) y (1 - y) = (1 - 0,42) (0,42) (1 - 0,42) = 0,14$$

Suku perubahan bobot w_{jk} (dengan $\alpha = 0,2$) :

$$\Delta w_{jk} = \alpha \delta_k z_j = \alpha \delta z_j \quad (k=1 ; j = 0,1,2,3)$$

$$\Delta w_{01} = 0,2 (0,14) (1) = 0,03$$

$$\Delta w_{11} = 0,2 (0,14) (0,48) = 0,01$$

$$\Delta w_{21} = 0,2 (0,14) (0,65) = 0,02$$

$$\Delta w_{31} = 0,2 (0,14) (0,55) = 0,03$$

Langkah 7 : Hitung penjumlahan kesalahan dari unit tersembunyi (δ)

$$\delta_{\text{net } j} = \sum_{k=1}^m \delta_k w_{jk} . \text{ Karena jaringan hanya memiliki sebuah unit keluaran maka}$$

$$\delta_{\text{net } j} = \delta w_{j1} \quad (j = 1,2,3)$$

$$\delta_{\text{net } 1} = (0,14) (0,49) = 0,07$$

$$\delta_{\text{net } 2} = (0,14) (-0,31) = 0,04$$

$$\delta_{\text{net } 3} = (0,14) (-0,41) = -0,06$$

Faktor kesalahan δ di unit tersembunyi :

$$\delta_j = \delta_{\text{net } j} f'(z_{\text{net } j}) = \delta_{\text{net } j} z_j (1 - z_j)$$

$$\delta_1 = -0,05 (0,55) (1 - 0,55) = 0,02$$

$$\delta_2 = -0,04 (0,65) (1 - 0,65) = -0,01$$

$$\delta_3 = -0,06 (0,55) (1 - 0,55) = -0,01$$

Suku perubahan bobot ke unit tersembunyi $\Delta v_{ij} = \alpha \delta_j x_i \quad (j = 1,2,3 ; i = 0,1,2)$

	Z_1	Z_2	Z_3
X_1	$\Delta v_{11} = (0,2) (0,02)$ $(1) = 0,004 \approx 0$	$\Delta v_{12} = (0,2) (-0,01)$ $(1) = -0,002 \approx 0$	$\Delta v_{13} = (0,2) (-0,01)$ $(1) = -0,002 \approx 0$
X_2	$\Delta v_{21} = (0,2) (-0,01)$ $(0) = 0$	$\Delta v_{22} = (0,2) (-0,01)$ $(0) = 0$	$\Delta v_{23} = (0,2) (-0,01)$ $(0) = 0$
1	$\Delta v_{01} = (0,2) (0,02)$ $(1) = 0,004 \approx 0$	$\Delta v_{02} = (0,2) (-0,01)$ $(1) = -0,002 \approx 0$	$\Delta v_{03} = (0,2) (-0,01)$ $(1) = -0,002 \approx 0$

Langkah 8 : Hitung semua perubahan robot

Bobot baru unit keluaran:

$$w_{jk}(T+1) = w_{jk}(T) + \alpha \delta_k z_j + \mu (w_{jk}(T) - w_{jk}(T-1)) \quad (k=1; j=0,1,2,3)$$

Suku $\alpha \delta_k z_j$ adalah hasil yang didapat dari langkah 6.

$w_{jk}(T)$ adalah bobot mula-mula pola kedua (hasil iterasi pola pertama contoh 1) sedangkan $w_{jk}(T-1)$ adalah bobot mula-mula pada iterasi pola pertama (bobot awal contoh 1)

$$w_{11}(\text{baru}) = 0,49 + 0,01 + 0,5 (0,49 - 0,5) = 0,495$$

$$w_{21}(\text{baru}) = -0,31 + 0,02 + 0,5 (-0,31 - (-0,3)) = -0,295$$

$$w_{31}(\text{baru}) = -0,41 + 0,02 + 0,5 (-0,41 - (-0,4)) = -0,395$$

$$w_{01}(\text{baru}) = -0,12 + 0,03 + 0,5 (-0,12 - (-0,1)) = -0,1$$

Perubahan bobot unit tersembunyi

$$v_{ij}(T+1) = v_{ij}(T) + \alpha \delta_j x_i + \mu (v_{ij}(T) - v_{ij}(T-1)) \quad (j=1,2,3 ; i=0,1,2)$$

$v_{ij}(T)$ adalah bobot awal yang dipakai pada contoh 3 ini (hasil iterasi v_{ij} pada contoh 1), sedangkan $v_{ij}(T-1)$ adalah bobot awal yang dipakai pada contoh 1.

Keduanya bernilai sama sehingga suku momentum ($v_{ij}(T) - v_{ij}(T-1)$) = 0

$\alpha \delta_j x_i$ adalah hasil langkah 7, yang bernilai = 0 untuk setiap i dan j

Berarti v_{ij} tidak mengalami perubahan. v_{ij} baru hasil iterasi pola kedua sama dengan tabel 7.4 contoh 1.

7B.3.2 Delta-Bar-Delta

Dalam standar BP, laju pembelajaran α merupakan suatu konstanta yang dipakai dalam seluruh iterasinya. Perubahan dapat dilakukan dengan memberikan laju pembelajaran yang berbeda-beda untuk setiap bobotnya (atau bahkan laju pembelajaran yang berbeda-beda untuk tiap bobot dalam tiap iterasinya). Apabila perubahan bobot berada dalam arah yang sama dalam beberapa pola terakhir (dapat dilihat dari tanda suku $\delta_k z_j$ yang selalu sama), maka laju pemahaman yang bersesuaian dengan bobot w_{jk} ditambah. Sebaliknya apabila arah perubahan bobot dua pola terakhir berbeda (tanda suku $\delta_k z_j$ yang berselang seling positif – negatif), maka laju pemahaman untuk bobot tersebut harus dikurangi.

Perubahan bobot dalam aturan Delta-Bar-Delta adalah sbb:

$$w_{jk}(T+1) = w_{jk}(T) + \alpha_{jk}(T+1) \delta_k z_j$$

7B.3.3 Perubahan Bobot Berkelompok

Variasi lain yang dapat dilakukan pada standar BP adalah mengubah bobotnya sekaligus setelah semua pola dimasukkan. Untuk setiap pola yang dimasukkan, dilakukan langkah 4 – 7 standar BP. Nilai Δw_{jk} dan Δv_{ij} untuk tiap pola dijumlahkan. Langkah 8 (perhitungan bobot baru) dilakukan berdasarkan hasil jumlahan Δw_{jk} dan Δv_{ij} tersebut.

Prosedur ini memberikan efek yang lebih halus dalam perubahan bobot. Dalam beberapa kasus, variasi perubahan ini akan meningkatkan kemungkinan konvergensi ke titik minimum lokal.