Bab VIII
LVQ (Learning Vector Quantization)

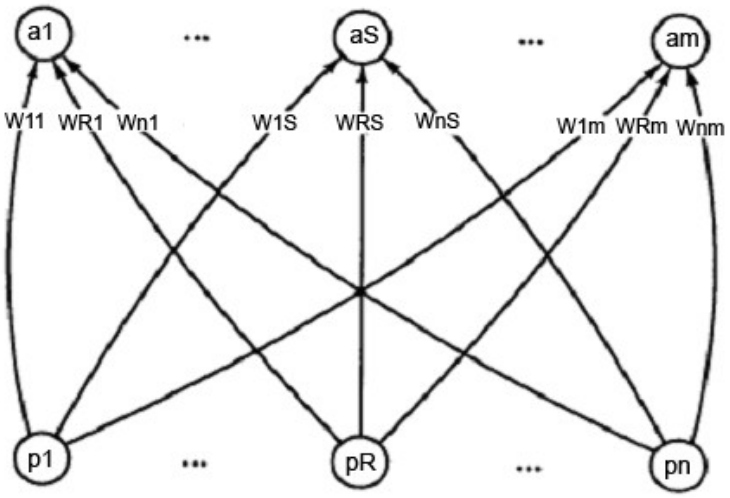
8.1 Definisi LVQ

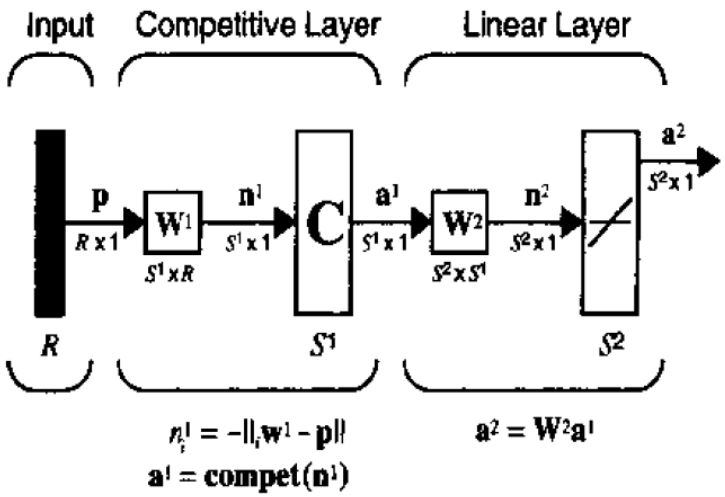
LVQ merupakan suatu metode untuk melakukan pembelajaran pada lapisan kompetitif dan merupakan gabungan dari terbimbing (supervised). Suatu lapisan kompetitif akan secara otomatis belajar untuk mengklasifikasikan vektor-vektor input. Jika dua vektor input mendekati sama, maka lapisan kompetitif akan meletakkan kedua vektor input tersebut ke dalam kelas yang sama. Jadi LVQ adalah metode untuk klasifikasi (pengelompokan) pola dan memiliki output yang mewakili dari kelas tertentu. Bobot vektor untuk sebuah layer output sering berfungsi sebagai vektor referensi untuk kelas dimana layer ditampilkan. Sepanjang pelatihan layer output ditempatkan (dengan mengatur bobot sepanjang paletihan terbimbing) untuk memperkirakan permukaan keputusan dari teori klasifikasi Bayes. Hal ini diasumsikan sebagai satu set dari pola pelatihan dengan klasifikasi yang disediakan. Sepanjang sebuah distribusi inisial atau vektor referensi (masing-masingnya melambangkan klasifikasi yang diketahui).

LVQ menggunakan pendekatan SOM, dimana terdapat training vector yang secara rekursif mengoptimasi penempatan hidden layer yang akan saling berkompetisi. Setelah dilakukan pembelajaran, sebuah vector masukan dikelompokkan ke dalam kelas, dimana kelas tersebut adalah representasi dari hidden layer yang terdekat, terhadap vektor input. Pembelajaran pada LVQ dilakukan dengan memasukkan vektor input dan menyesuaikan lokasi hidden layer berdasarkan kedekatannya terhadap vektor input tersebut. Hidden layer terdekat dipindahkan (secara proporsional terhadap tingkat pembelajaran) mendekati training vektor jika kelas hidden layer dan training vektornya cocok, dan menjauhi training vektor jika sebaliknya.

8.2 Arsitektur LVQ

Pada dasarnya arsitektur LVQ sama dengan self-organizing map kohonen (tanpa struktur topologi yang diasumsikan dengan unit output). Sebagai tambahan setiap layer output mempunyai kelas yang diketahui.





Gambar 8.1 Arsitektur LVQ

Arsitektur LVQ terdiri dari sebuah competitive layer, dan sebuah linear layer. Competitive layer mengklasifikasikan vektor input ke dalam sejumlah cluster berdasarkan jarak yang terdapat diantara masing – masing vektor masukan. Pada tahap kedua,linear layer memetakan kelas yang didapatkan oleh kompetitive layer ke dalam kelas yang telah didefinisikan sebelumnya oleh pengguna.

3.1 Algoritma LVQ

Algoritma LVQ bertujuan akhir mencari nilai bobot yang sesuai untuk mengelompokkan vektor – vektor kedalam kelas tujuan yang telah di inisialisasi pada saat pembentukan jaringan LVQ. Sedangkan algoritma pengujiannya adalah menghitung nilai output (kelas vektor) yang terdekat dengan vektor input, atau dapat disamakan dengan proses pengklasifikasian (pengelompokan).

Keterangan yang kita gunakan adalah sebagai berikut:

p : vektor pelatihan (input) (p_1,…...,p_R,……,p_n)
T : kategori yang tepat atau kelas untuk vektor pelatihan
w_s : bobot vektor untuk unit output ke-s (w_{11},…….,w_{R1},.....,w_{n1})
C_s : kategori atau kelas yang ditampilkan oleh unit output ke-s
$||p - w_j||$: jarak Euclidean antara vektor input dan bobot vektor untuk layer output ke-s

Berikut ini adalah algoritma pembelajaran LVQ:

langkah 0 : inisialisasi vektor referensi ; inisialisasi drating pembelajaran α (0)
langkah 1 : ketika kondisi berhenti adalah false, lakukan langkah 2 sampai 6
langkah 2 : untuk setiap input pelatihan vektor x lakukan langkah 3 – 4
langkah 3 : temukan j hingga $||p – w_p||$ minimum
langkah 4 : perbaharui w_s sebagai berikut :
jika T = C_s, maka
$$W_s(baru) = W_s(lama) + \alpha[x – w_s(lama)];$$
jika T ≠ C_s, maka
$$W_s(baru) = W_s(lama) - \alpha[x – w_s(lama)];$$

langkah 5 : kurang rating pelatihan
langkah 6 : tes kondisi berhenti:
yaitu kondisi yang mungkin menetapkan sebuah juumlah tetap dari iterasi atau rating pembelajaran mencapai nilai kecil yang cukup.

8.4 Aplikasi LVQ

Metode yang paling sederhana dari inisialisasi bobot vektor adalah mengambil vektor pelatihan pertama dan menggunakannya sebagai bobot vektor. Vektor yang tersisa digunakan untuk pelatihan.

Metode lain yang mungkin untuk menginisialisasi bobot adalah menggunakan pengelompokan K-Means atau self organizing map kohonen untuk menempatkan bobot. Setiap bobot vektor kemudian dikalibrasi dengan menetukan pola input yang terdekat kepadanya, mencari kelas yang mempunyai jumlah terbeasr dari pola input yang termasuk kedalamnya dan menetapkan kelas itu pada bobot vektor.

Contoh sederhana:
Lima vektor ditetapkan pada dua kelas.
Ini adalah contoh yang sangat sederhana, dua vektor referensi akan digunakan. vektor input dibawah ini melambangkan dua kelas, 1 dan 2

vektor	kelas
(1, 1, 0,0)	1
(0, 0, 0, 1)	2
(0, 0, 1, 1)	2
(1, 0, 0, 0)	1
(0, 1, 1, 0)	2

Dua vektor pertama akan digunakan untuk menginisialisasi dua vektor referensi. Layer output pertama melambangkan kelas 1, kelas yang kedua adalah 2(disimbolkan. $C_1 = 1$ dan $C_2 = 2$). Ini meninggalkan vektor (0, 0, 1, 1), (1, 0, 0, 0), dan (0, 1, 1, 0) sebagai vektor pelatihan. Hanya satu iterasi (satu epoh) ditunjukkan.

Langkah 1. inisialisasi bobot:
w_1= (1, 1, 0,0);
w_2= (0, 0, 0, 1);
inisialisasi rating pembelajaran α=1
Langkah 2. mulai perhitungan
Langkah 3. untuk vektor input x=(0, 0, 1, 1) dengan T=2, lakukan langkah 3-4
langkah 3. J=2, sejak x adalah lebih dekat dengan w_2 daripada w_1
langkah 4. sejak T=2 dan C_2=2, perbaharui w_2 sebagai berikut

$$w_2=(0, 0, 0, 1) +.1[(0, 0, 1, 1)- (0, 0, 0, 1)]$$
$$=(0, 0, .1, 1) .$$

Langkah 4. untuk vektor input x=(1, 0, 0, 0) dengan T=1, lakukan langkah 3-4
langkah 3. J=1, sejak x adalah lebih dekat dengan w_2 daripada w_1
langkah 4. sejak T=1 dan C_2=1, perbaharui w_1 sebagai berikut

$$w_1=(1, 1, 0, 0) +.1[(1, 0, 0, 0)- (1, 1, 0, 0)]$$
$$=(1, .9, 0, 0) .$$

Langkah 5. untuk vektor input x=(0, 1, 1, 0) dengan T=2, lakukan langkah 3-4
langkah 3. J=1, sejak x adalah lebih dekat dengan w_2 daripada w_1
langkah 4. sejak T=2 dan C_1=1, perbaharui w_1 sebagai berikut

$$w_1=(1, .9, 0, 0) +.1[(0, 1, 1, 0)- (1, .9, 0, 0)]$$
$$=(1.1, .89, -.1, 0) .$$

Langkah 6. Ini menyelesaikan 1 epoh pelatihan
kurangi rating pelatihan
Langkah 7. tes kondisi berhenti.

Soal – Soal Latihan

1. Diketahui lima buah vektor yang diklasifikasikan menjadi dua kelas:

Data	kelas
1100	1
0001	2
0011	2
1000	1
0111	2

Hitunglah bobot pembelajaran dengan menggunakan metode LVQ.

2. Disajikan empat buah data dengan simbol kelas menggunakan + # * $ dan empat kelas berbeda sebagai berikut:

	inisial	W
Kelas 1 (+)	1	1
Kelas 2 (#)	0	0
Kelas 3 (*)	1	0
Kelas 4 ($)	0	1

DAFTAR PUSTAKA

[1]. Fausett, Laurene.1994. *Fundamentals Of Neural Networks*. Prentice-Hall.

[2]. Hagan, Martin T. DKK. 1996. *Neural Network Design.* USA: PWS Publishing Company.

[3]. Zaknich, Anthony. 2002. *Neural Network For Intelligent Signal Processing*. Australia: World Scientific.

DAFTAR PUSTAKA dari Internet

[4]. http://www.en.wikipedia.org/wiki/LVQ

[5]. http://www.mathworks.fr/access/helpdesk.html

[6]. http://www.cis.hut.fi/jhollmen/Publications/lvqprob.pdf

[7]. http://www.cs.ui.ac.id/id/wp-content/uploads/2008/08/42%20Perbandingan%20kinerja%20jaringan%20syaraf-Rahmat%20W.07.pdf