Praktikum Sistem Cerdas

NRP	:	3223600019
Nama	:	Muhammad Bimo Fachrizky
Materi	:	Membuat Program Aplikasi Jaringan Kohonen
Tanggal	:	Senin, 24 Maret 2025

Praktikum 6

Membuat Program Aplikasi Jaringan Kohonen

I. Tujuan Pembelajaran

- Mahasiswa dapat memahami dan menjelaskan konsep Jaringan Kohonen
- Mahasiswa dapat menjelaskan model Jaringan Kohonen
- Mahasiswa dapat membuat aplikasi Jaringan Kohonen

Software yang di perlukan

- Microsoft Visual C++
- PyCharm

II. Langkah percobaan

1. Program Jaringan Kohonen

```
#include "stdio.h"
#include <conio.h>
#include <math.h>
void main() {
   float w[2][4] = { {0.2,0.6,0.5,0.9},
                     {0.8,0.4,0.7,0.3} };
   int x[4][4] = { {1,1,0,0},
                   {0,0,0,1},
                   {1,0,0,0},
                   {0,0,1,1} };
   float D[2];
   int i, j, k, l;
   float LR = 0.6;
   //training
   for (l = 0; l < 100; l++) {
      for (k = 0; k < 4; k++) {
         for (j = 0; j < 2; j++) {
            D[j] = 0.0;
            for (i = 0; i < 4; i++) {
               D[j] = D[j] + (w[j][i] - x[k][i]) * (w[j][i] - x[k][i]);
            }
         }
         if (D[0] < D[1]) {
            for (i = 0; i < 4; i++) {
               w[0][i] = w[0][i] + LR * (x[k][i] - w[0][i]);
            }
         }
         else {
```

```
            for (i = 0; i < 4; i++) {
               w[1][i] = w[1][i] + LR * (x[k][i] - w[1][i]);
            }
         }
      }
      LR = 0.5 * LR;
   }
   //mapping cluster
   x[0][0] = 1; x[0][1] = 1; x[0][2] = 0; x[0][3] = 0;
   for (j = 0; j < 2; j++) {
      D[j] = 0.0;
      for (i = 0; i < 4; i++) {
         D[j] = D[j] + sqrt((w[j][i] - x[0][i]) * (w[j][i] - x[0][i]));
      }
      printf("D(%d) = %.2f\n", (j + 1), D[j]);
   }
   if (D[0] < D[1]) {
      printf("Cluster 1");
   }
   else {
      printf("Cluster 2");
   }
}
```

2. Tugas

   Misalkan kita ingin mengcluster data-data berikut:

X1	X2
0,10	0,10
0,20	0,20
0,30	0,10
0,50	0,30
0,40	0,40
0,20	0,40

   menjadi 2 cluster. Bobot awal yang akan kita gunakan adalah matriks berukuran 2x2 dengan tiap-tiap elemen bernilai 0,5. Learning rate (a=0, 6) dengan tiap kenaikan epoh akan diset 0,5x(a) . Maksimum epoh ditetapkan sebesar 10.

   - Program

```
#include "stdio.h"
#include <math.h>
```

```
int main() {
    float x[6][2] = {
        {0.10, 0.10},
        {0.20, 0.20},
        {0.30, 0.10},
        {0.50, 0.30},
        {0.40, 0.40},
        {0.20, 0.40}
    };

    float w[2][2] = { {0.5, 0.5}, {0.5, 0.5} };
    float D[2];
    int i, j, k, l;
    float LR = 0.6;

    // Training
    for (l = 0; l < 10; l++) {
        for (k = 0; k < 6; k++) {
            for (j = 0; j < 2; j++) {
                D[j] = 0.0;
                for (i = 0; i < 2; i++) {
                    D[j] += (w[j][i] - x[k][i]) * (w[j][i] - x[k][i]);
                }
            }

            if (D[0] < D[1]) {
                for (i = 0; i < 2; i++) {
                    w[0][i] += LR * (x[k][i] - w[0][i]);
                }
            } else {
                for (i = 0; i < 2; i++) {
                    w[1][i] += LR * (x[k][i] - w[1][i]);
```

```
            }
        }
    }
    LR *= 0.5;
}

printf("w[0] = [%.4f, %.4f]\n", w[0][0], w[0][1]);
printf("w[1] = [%.4f, %.4f]\n", w[1][0], w[1][1]);

// Mapping Cluster
float test[2] = {0.10, 0.10};
for (j = 0; j < 2; j++) {
    D[j] = 0.0;
    for (i = 0; i < 2; i++) {
        D[j] += (w[j][i] - test[i]) * (w[j][i] - test[i]);
    }
    printf("D(%d) = %.4f\n", j + 1, D[j]);
}

if (D[0] <= D[1]) {
    printf("Cluster 1\n");
} else {
    printf("Cluster 2\n");
}

return 0;
}
```

III. Hasil Percobaan

- Jaringan Kohonen Percobaan Pertama

```
E:\Program Files\Documents\Kuliah Semester 4\
D(1) = 3.54
D(2) = 0.64
Cluster 2
```

- Jaringan Kohonen Percobaan Kedua

```
E:\Program Files\Documents\Kuliah Semester 4\
w[0] = [0.3424, 0.3754]
w[1] = [0.2190, 0.1351]
D(1) = 0.1346
D(2) = 0.0154
Cluster 2
```

IV. Analisa

Praktikum pertama adalah implementasi sederhana dari jaringan kohonen untuk clustering atau pengelompokan data. Matriks bobot (weight) di inisialisasikan sebagai w dengan ukuran 2x4 yang menandakan ada 2 neuron/cluster. Sedangkan x adalah data input yang berjumlah 4 buah, masing-masing dengan 4 fitur biner. Pada program pertama ini loop dilakukan training sebanyak 100 iterasi. Program akan menghitung jarak Euclidean kuadrat antara setiap input x[k] dan kedua bobot w[0] dan w[1]. Nilai terkecil atau jarak terdekat akan dianggab sebagai cluster pemenang. Hanya bobot dari neuron menang yang akan diupdate untuk mendekati data input. Setelah training selesai, program menguji satu data baru [1,1,0,0] untuk menentukan ke cluster mana data tersebut termasuk. Jarak yang dihitung menggunakan akar dari selisih kuadrat. Kemudian program akan mencetak jarak dari data ke masing-masing cluster dan menentukan ke cluster mana data tersebut lebih dekat.

Praktikum kedua menggunakan penddekatan competitive learning dengan data yang berdimensi 2 dan hanya 2 cluster. X[6][2] berisi 6 data titik dalam bentuk (x,y). w[2][2] adalah 2 bobot (centroid cluster), awalnya keduanya berada di titik (0.5,0.5). Menghitung jarak kuadrat Euclidean antara data ke- k dengan masing-masing pusat cluster w[0] dan w[1]. kemudian proses akan menggeser bobot mendekati data yang paling mirip. Titik test diujikan ke dua bobot hasil training, jarak kuadrat Euclidean dihitung dan titik diklasifikasikan ke cluster dengan jarak terkecil. Output akan menampilkan bobot akhir (posisi dua cluster) dan menampilkan jarak ke titik uji dan hasil klasifikasinya.

V. Kesimpulan

Dua praktikum ini menunjukkan implementasi dasar dari jaringan saraf kompetitif (competitive learning), khususnya jaringan Kohonen, yang berfungsi untuk melakukan clustering atau pengelompokan data tanpa label (unsupervised learning). Pada praktikum pertama, sistem menggunakan input biner berdimensi 4 dengan dua neuron sebagai representasi cluster. Bobot awal diupdate secara bertahap berdasarkan jarak Euclidean kuadrat antara data input dan bobot neuron. Hanya bobot dari neuron pemenang (yang paling dekat dengan input) yang diperbarui. Proses ini dilakukan berulang untuk mencapai konvergensi. Setelah pelatihan, data uji diberikan untuk menentukan cluster terdekat menggunakan jarak Euclidean biasa. Sementara pada praktikum kedua, pendekatan yang sama diterapkan pada data dua dimensi (x, y), yang lebih mudah divisualisasikan. Proses pelatihan juga menggunakan prinsip kompetitif, dengan dua bobot awal yang identik, dan learning rate yang menurun seiring waktu agar sistem menjadi stabil. Hasil akhirnya digunakan untuk mengelompokkan satu titik uji ke dalam cluster terdekat. Secara keseluruhan, kedua praktikum ini menunjukkan bagaimana jaringan Kohonen dapat mengenali pola dan membentuk representasi cluster dari data, meskipun tanpa label atau kategori yang diberikan sebelumnya.