

# **Praktikum 3**

## **Membuat Program Aplikasi Perceptron**

### **A. Tujuan**

1. Mahasiswa dapat memahami dan menjelaskan konsep Perceptron
2. Mahasiswa dapat menjelaskan model Perceptron
3. Mahasiswa dapat membuat aplikasi Perceptron

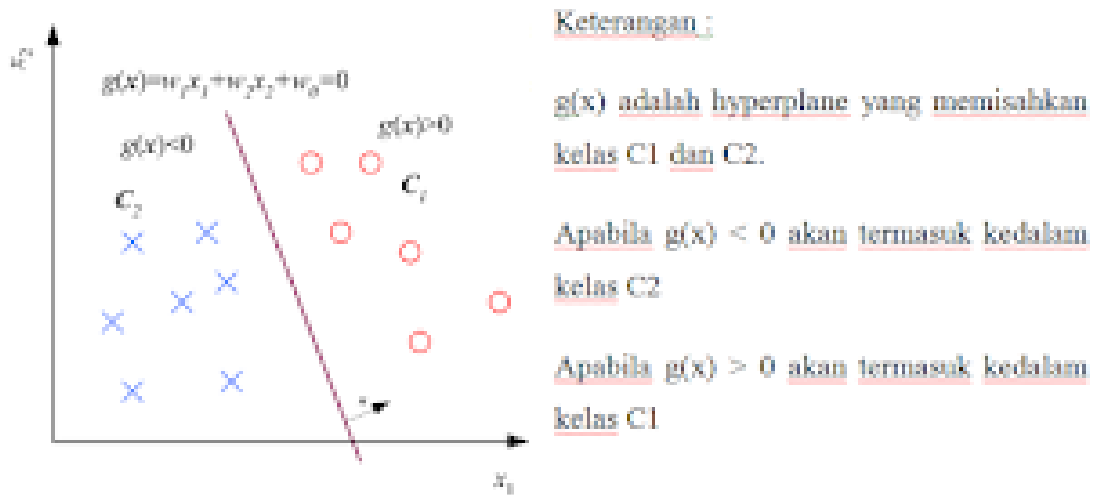
Software yang diperlukan:

- Microsoft Visual C++
- PyCharm

### **B. Pendahuluan**

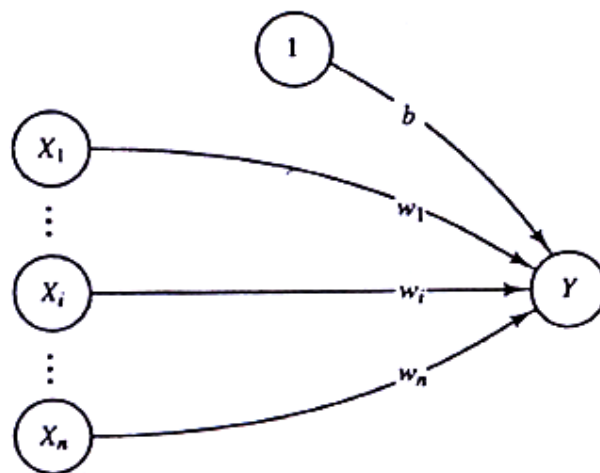
#### **1. Perceptron**

Perceptron adalah salah satu metode Jaringan Syaraf Tiruan (JST) sederhana yang menggunakan algoritma training untuk melakukan klasifikasi secara linier. Perceptron digunakan untuk melakukan klasifikasi sederhana dan membagi data untuk menentukan data mana yang masuk dalam klasifikasi dan data mana yang missclasifikasi (diluar klasifikasi). Perceptron dapat kita gunakan untuk memisahkan data yang dapat kita bagi menjadi 2 kelas, misalnya kelas C1 dan kelas C2. Perceptron dalam Jaringan Syaraf Tiruan memiliki bobot yang bisa diatur dalam suatu ambang batas (threshold). Melalui proses pembelajaran (learning), Algoritma Perceptron akan mengatur parameter-parameter bebasnya.



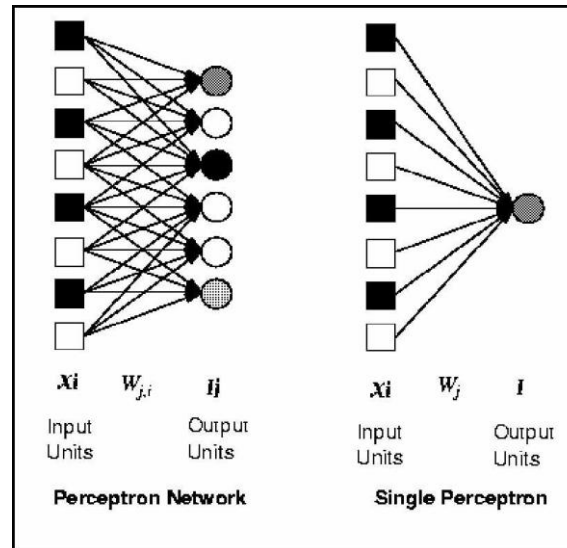
Gambar 1. Klasifikasi Perceptron

Tanda x (biru) dan tanda o (merah) pada gambar diatas dicontohkan adalah dua buah data yang akan diklasifikasi. Dimana kedua data tersebut akan dipisahkan dengan suatu garis hyperplane yang membagi data menjadi dua klasifikasi. Dimana (  $g(x)$  ) merupakan vektor hyperplane yang memisahkan kelas C1 (data x) dan C2 (data o). Apabila nilai  $g(x) < 0$  ( $g(x)$  lebih kecil dari nol) maka data tersebut termasuk dalam kelas C2, sedangkan apabila  $g(x) > 0$  ( $g(x)$  lebih besar dari nol) maka data tersebut termasuk kedalam kelas kelas C1.



Gambar 2. Model Perseptron

Jaringan lapis tunggal Perseptron (single layer perceptron) terdiri dari beberapa unit pemroses (neuron), yang terhubung dan mempunyai beberapa masukan serta sebuah atau beberapa keluaran. Single Perceptron sering disebut juga dengan Perceptron.



Gambar 3. Jenis Perceptron

Perseptron menghitung jumlah nilai perkalian penimbang dan masukan dari parameter permasalahan yang kemudian dibandingkan dengan nilai threshold. Bila nilai keluaran lebih besar dari threshold maka keluarannya adalah satu, sebaliknya adalah nol. Jenis perceptron yaitu single output dan multi output.

## 2. Algoritma Perceptron

1. Inisialisasi semua input, matriks vector, target, dan learning rate.
2. Kita lakukan pengecekan untuk semua vector yang misclassification dengan menggunakan rumus berikut :

$$n = \sum x_i w_i + b$$

Jaringan terdiri dari beberapa unit masukan ( ditambah sebuah bias ), dan memiliki sebuah unit keluaran. Hanya saja fungsi aktivasi merupakan fungsi biner ( atau bipolar ), tetapi memiliki kemungkinan nilai -1, 0 atau 1.

Dengan ketentuan untuk suatu harga threshold  $\theta$  yang ditentukan sebagai berikut :

$$a = f(n) = \begin{cases} 1 & n > \theta \\ 0 & -\theta \leq n \leq \theta \\ -1 & n < -\theta \end{cases}$$

3. Jika hasilnya bernilai negatif, maka terjadi miss classification dan kalikan hasilnya dengan negatif learning rate. Kemudian hasilnya disimpan dalam bentuk matriks untuk seluruh inputan yang miss clasification.
4. Kemudian lakukan perubahan bobot yang baru dengan menggunakan rumus :

$$W_i(\text{baru}) = w_i(\text{lama}) + \Delta w \quad (i = 1, \dots, n) \text{ dengan } \Delta w = \alpha t x_i$$

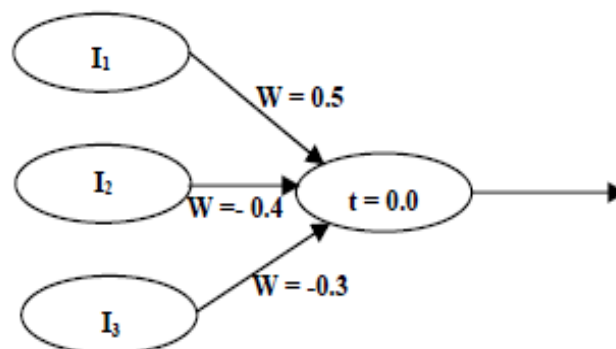
$$b(\text{baru}) = b(\text{lama}) + \Delta b = \alpha t$$

5. Kemudian ulangi langkah 2 hingga tidak terdapat lagi vector misclassification.

Pelatihan pada perseptron dilakukan dengan merubah nilai penimbangnya sehingga sesuai dengan kebutuhan yang dilakukan dengan membandingkan keluaran dari jaringan dengan targetnya dan proses tersebut dituliskan seperti berikut ini:

$$W_{baru_{ji}} = W_{lama_{ji}} + \alpha(t_j - O)x_i$$

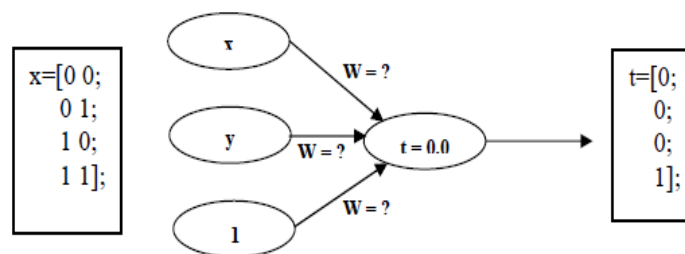
$t_j$  adalah target,  $O$  adalah output dari jaringan dan  $\alpha$  adalah bilangan konstan bernilai kecil antara 0,1 sampai 0,9 yang disebut sebagai laju pelatihan (learning rate). Mula-mula dilakukan random pada bobot ( $W$ ) kemudian dilakukan perhitungan perkalian antara input ( $I$ ) dan bobot ( $W$ ) seperti:



$I_1$	$I_2$	$I_3$	Penjumlahan	Output
0	0	1	$(0*0.5) + (0*-0.4) + (1*-0.3) = -0.3$	0
0	1	1	$(0*0.5) + (1*-0.4) + (1*-0.3) = -0.7$	0
1	0	1	$(1*0.5) + (0*-0.4) + (1*-0.3) = 0.2$	1
1	1	1	$(1*0.5) + (1*-0.4) + (1*-0.3) = -0.2$	0

Gambar 4. Perhitungan Pelatihan Perseptron

Output diperoleh dengan melakukan fungsi aktivasi dalam kasus AND dilakukan dengan unit step yaitu bila kurang dari nol output = 0, bila lebih output = 1. Error diperoleh apabila terjadi perbedaan antara target dengan output jaringan. Jika error tidak sama dengan nol maka bobot (W) diperbaiki.  $W_{baru_{ji}} = W_{lama_{ji}} + \alpha(t_j - O)x_i$



Gambar 5. Error diperoleh apabila terjadi perbedaan target dan output jaringan

### C. Percobaan

#### 1. Logika AND

AND		
$X_1$	$X_2$	Y
0	0	0
0	1	0
1	0	0
1	1	1

```

double d_rand(){
    return(((double)(rand()%32767)/32767.0-0.5)*2.0);
}
void main()
{
    int i,j,out,ERR,y;
    int x[3][4]= {
        {0,0,1,1},
        {0,1,0,1},
        {1,1,1,1}
    };
    int T[4]={0,0,0,1};
    float w[3],O,LR=0.1,init=0.15;
    //inisialisasi bobot
    for(i=0;i<3;i++){
        w[i]=init*d_rand();
    }
    //traning
    for(i=0;i<10;i++){
        for(j=0;j<4;j++){
            O=x[0][j]*w[0]+x[1][j]*w[1]+x[2][j]*w[2];
            if(O>0.0)
                out=1;
            else
                out=0;
            ERR=T[j]-out;
            if(ERR!=0){
                w[0]=w[0]+LR*x[0][j]*ERR;
                w[1]=w[1]+LR*x[1][j]*ERR;
                w[2]=w[2]+LR*x[2][j]*ERR;
            }
        }
        printf("Err:%d\n",ERR);
    }
    //running
    for(i=0;i<4;i++){
        printf("Input X1:");
        scanf("%d",&x[0][0]);
        printf("Input X2:");
        scanf("%d",&x[1][0]);
        O=x[0][0]*w[0]+x[1][0]*w[1]+x[2][0]*w[2];
        if(O>0.0)
            y=1;
        else
            y=0;
        printf("Output:%d\n",y);
    }
}

```

## Hasil Program

```
"D:\PENS\Tekkom\Sistem Cerdas\2020 Sistem Cerdas\Praktikum Sistem Cerdas"
Err:1
Err:1
Err:1
Err:1
Err:1
Err:1
Err:1
Err:0
Err:0
Err:0
Err:0
Input X1:0
Input X2:0
Output:0
Input X1:0
Input X2:1
Output:0
Input X1:1
Input X2:0
Output:0
Input X1:1
Input X2:1
Output:1

Process returned 0 (0x0)   execution time : 10.503 s
Press any key to continue.
```

## 2. Logika OR

OR

$X_1$	$X_2$	Y
0	0	0
0	1	1
1	0	1
1	1	1

```

double d_rand(){
    return(((double)(rand()%32767)/32767.0-0.5)*2.0);
}
void main()
{
    int i,j,out,ERR,y;
    int x[3][4]= {
        {0,0,1,1},
        {0,1,0,1},
        {1,1,1,1}
    };
    int T[4]={0,1,1,1};
    float w[3],O,LR=0.1,init=0.15;
    //inisialisasi bobot
    for(i=0;i<3;i++){
        w[i]=init*d_rand();
    }
    //traning
    for(i=0;i<10;i++){
        for(j=0;j<4;j++){
            O=x[0][j]*w[0]+x[1][j]*w[1]+x[2][j]*w[2];
            if(O>0.0)
                out=1;
            else
                out=0;
            ERR=T[j]-out;
            if(ERR!=0){
                w[0]=w[0]+LR*x[0][j]*ERR;
                w[1]=w[1]+LR*x[1][j]*ERR;
                w[2]=w[2]+LR*x[2][j]*ERR;
            }
        }
        printf("Err:%d\n",ERR);
    }
    //running
    for(i=0;i<4;i++){
        printf("Input X1:");
        scanf("%d",&x[0][0]);
        printf("Input X2:");
        scanf("%d",&x[1][0]);
        O=x[0][0]*w[0]+x[1][0]*w[1]+x[2][0]*w[2];
        if(O>0.0)
            y=1;
        else
            y=0;
        printf("Output:%d\n",y);
    }
}

```



## Hasil Program

```
"D:\PENS\Tekkom\Sistem Cerdas\2020 Sistem Cerdas\Praktikum Sistem Cer
Err:0
Err:0
Err:0
Err:0
Err:0
Err:0
Err:0
Err:0
Err:0
Err:0
Err:0
Input X1:0
Input X2:0
Output:0
Input X1:0
Input X2:1
Output:1
Input X1:1
Input X2:0
Output:1
Input X1:1
Input X2:1
Output:1

Process returned 0 (0x0)   execution time : 10.299 s
Press any key to continue.
```

## D. Tugas

1. Buat program sederhana untuk kasus sederhana single perceptron AND dan OR.
2. Buat program untuk 7 segment input 7 output 10

