UNIVERSIDAD AUTÓNOMA DEL CARMEN

FACULTAD CIENCIAS DE LA INFORMACIÓN

ING. TECNOLOGÍAS DE CÓMPUTO Y COMUNICACIONES

Materia: Reconocimiento de Patrones

Mtro. Jesús Alejandro Flores Hernández

Alumna: Analí del Carmen Pérez Martínez

1.Generación del conjunto de trabajo

```
//////////////-Genera-el-conjunto-de-trabajo-//////////////
////////////////////////////////////////////////////////////
nct=20;-//tamaño-del-conjunto-de-trabajo
x=2*rand(2,nct)-1;
x1=x(1,:);//Arreglo-x1-contiene-coordenadas-x
y1=x(2,:);//Arreglo-y1-contiene-coordenadas-y
plot(x1,y1,'*');
```

- rand(2,nct) genera una matriz de 2×20 con valores entre 0 y 1.
- Multiplicarlo por 2 y restar 1 lo transforma a valores entre -1 y 1.
- Así se crean 20 puntos aleatorios en el plano XY.
- Se grafican con asteriscos ('*').

2.Definición de la línea base (y=2x)

```
//Graficamos-una-linea-arbitraria-y-2x=0-f(x):-y=2x
//Salvamos-los-coheficientes-de-x-y-y-en-el-arreglo-F
F=[1;-2];
```

Esto representa la línea y=2x, escrita como f(x,y)= x-2y =0

```
//mostramos área de trabajo para fines visuaes solamente
x2=linspace(-1,1,100);
for i=1:100
    y2(i)=2*x2(i);
end
plot(x2,y2,'r') //trazamos una línea roja
```

Se crea un conjunto de puntos para dibujar la línea y = 2x en color rojo.

2.Clasificación de puntos

```
for i=1:nct
    l(i)=-F(2)*x1(i)-y1(i);//l(
tamos la y d ela recta
    class_F(i)=sign(l(i));
end
```

Esto clasifica los puntos según su posición respecto a la línea roja:

- Si están encima de la línea: clase +1
- Si están debajo de la línea: clase -1

```
for i=1:nct
    if class_F(i)==1 then
        plot(xl(i),yl(i),'gre*');
    else
        plot(xl(i),yl(i),'blu*');
    end
end
```

Se muestran los puntos con diferentes colores según su clase. Verde para un lado de la línea, azul para el otro.

4.Entrenamiento del perceptrón

Inicializar pesos

```
//pesos iniciales
w=[0;0];
```

Calcular salida del perceptrón con esos pesos

```
for i=1:nct
    g(i)=sign(w(2)*yl(i)+w(1)*xl(i));
end
```

El perceptrón calcula una salida binaria (+1 o -1) para cada punto usando:
g(x)=sign(w1·x+w2·y)

```
//clasificación
ind = find(g ~= class_F);
iter=1;
while ~isempty(ind) //si es vacio => fin
    //actualizamos los pesos w
    //for i=1:nct
     //operador .* es multiplicación de matrices elemento a elemen
     w(1) = w(1) + (class_F(ind(1))-g(ind(1))).*xl(ind(1)); //...
     w(2) = w(2) + (class_F(ind(1))-g(ind(1))).*yl(ind(1)); //
    //end
    //temp = sign(w'*x);
    for i=1:nct
        temp(i)=sign(w(2)*yl(i)+w(1)*xl(i));
    end
    ind = find(temp ~= class_F);
    iter = iter + 1;
    g=temp;
end
```

Este bucle sigue actualizando los pesos hasta que todos los puntos estén correctamente clasificados, usando la regla clásica del perceptrón.

5.Visualización de la frontera aprendida

```
for i=1:100
    y3(i)=(-w(1)*x2(i))/w(2);
    //y3(i)=-(w(2)/w(1))*x2(i);
end
plot(x2,y3,'g');
```

Esta línea verde representa la hipótesis final del perceptrón, que debería separar correctamente los puntos verdes y azules.

6.Impresión del resultado

```
mfprintf(6,'---Al final los pesos son:\nvector w-----');
disp(w);
```

Muestra los pesos finales que definen la línea aprendida.

Salida