TALLER 1-PERCEPTRON

Presentado por

JONATHAN SNEYDER ARDILA NEIRA

20191007058

Docente

CESAR ANDREY PERDOMO CHARRY

Inteligencia Computacional Aplicada

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

Facultad de Ingeniería

Maestría en Ciencias de la información y las Comunicaciones

Bogotá, D.C

2024

Contents

Perceptrón

Enunciado

1. Desarrollar un script en Matlab que realice una red de una sola neurona y capa usando el modelo Perceptrón Simple que se muestra en la figura:

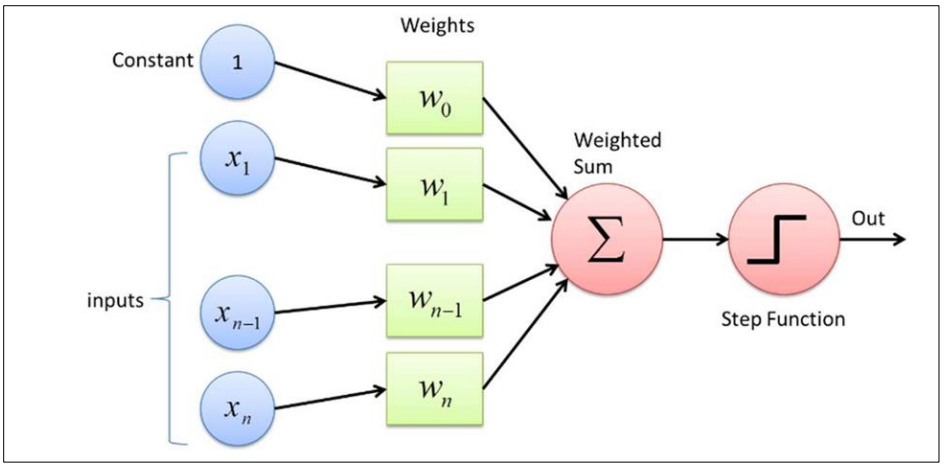


Fig 1. Modelo de perceptrón simple

En este modelo se debe dejar el script de tal forma que se puedan elegir las siguientes formas de corrección de los pesos:

- $\Delta W_i = d(x) * X_i$
- $\Delta W_i = [d(x) - Y(x)] * X_i$
- $\Delta W_i = \alpha * [d(x) - Y(x)] * X_i$

Donde:

- d(x) = Es la salida deseada del modelo
- Y(x) = Es la salida generada por el modelo
- α = Factor relativo de aprendizaje

2. Se debe dejar el script totalmente parametrizable de los parámetros del modelo: Número de entradas, Umbral de la función de activación, regla de aprendizaje, α, etc.
3. Probar el modelo desarrollado para resolver compuertas AND, OR, de 2, 3 y 4 entradas. Entregar reporte de aprendizaje ante distintas compuertas y distintos parámetros del modelo.
4. Modificar el modelo de tal forma que permita trabajar con los dataset generados en el script Dataset_train_test.mlx. Probar distintos modelos para los distintos dataset que se generan en distintas proporciones: 60-40, 70-30, 80-20 y 90-10. Ejemplo : Comparar varios modelos con distinto valor de α
5. Modificar Dataset_train_test.mlx. para generar a partir del dataset data_banknote_authentication.txt distintos dataset con proporciones 60-40, 70 30, 80-20 y 90-10.
6. Repetir el paso 4 para los dataset generados en el paso 5.

Desarrollo.

Compuerta AND

Dos entradas:

X0	X1	X2	Yd
1	-1	-1	-1
1	-1	1	-1
1	1	-1	-1
1	1	1	1

- Primera forma de corrección de pesos.

  W = d(x)*Xi

  1. Parámetros versión 1:
     a. Pesos 0.5
     b. Umbral 0

  El perceptrón alcanza una convergencia en dos iteraciones

```
Convergencia alcanzada después de 2 iteraciones
   -1.5000    2.5000    2.5000
```

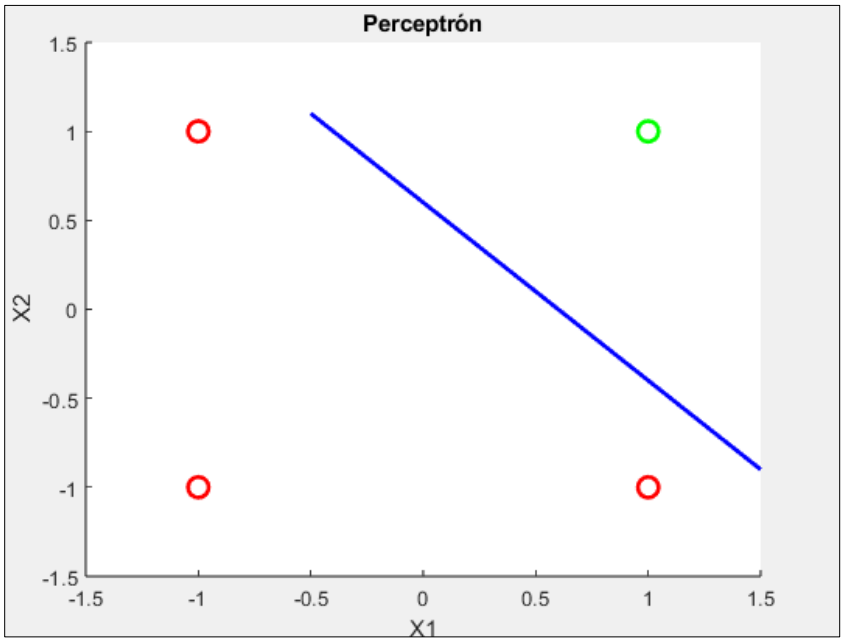


Fig. Línea que divide los diferentes puntos

2. Parámetros versión 2:
    a. Pesos aleatorios
    b. Umbral 0.5

    En esta configuración ocurre que no siempre llega a una convergencia, por lo tanto, los datos no quedan bien clasificados

```
Convergencia alcanzada después de 100 iteraciones
    0.2858    0.2858    0.2858
```

    Al limite de iteraciones configurado es de 100, acá vemos que llega a ese limite y la clasificación quedaría

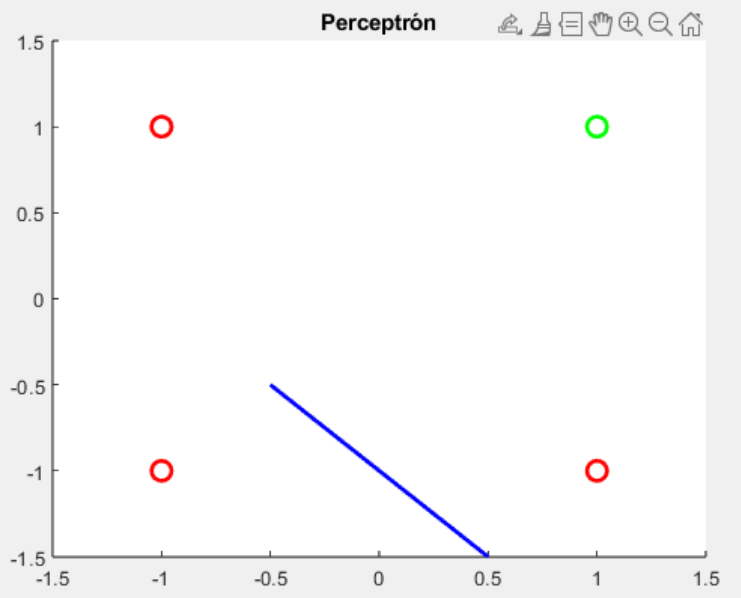


Fig. Clasificación errónea

Como también hay casos (la mayoría) donde hace una correcta clasificación

```
Convergencia alcanzada después de 2 iteraciones
   -1.2428    2.7572    2.7572
```

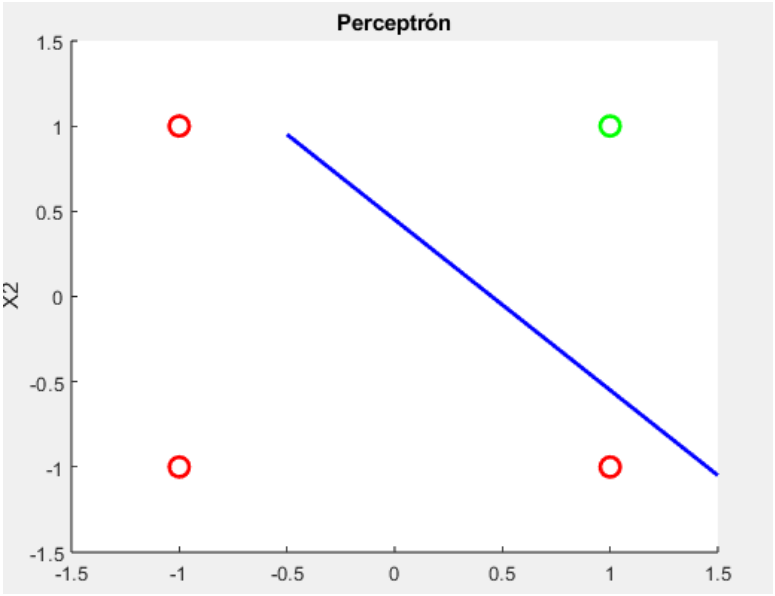


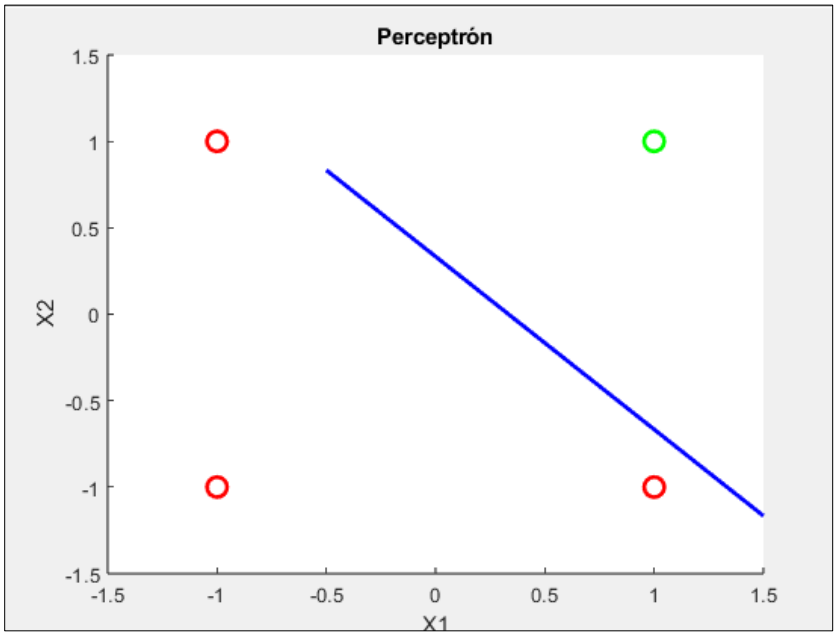
Fig. Clasificación

- Segunda forma de corrección de pesos.

  W = α*[d(x)-Y(x)]*Xi

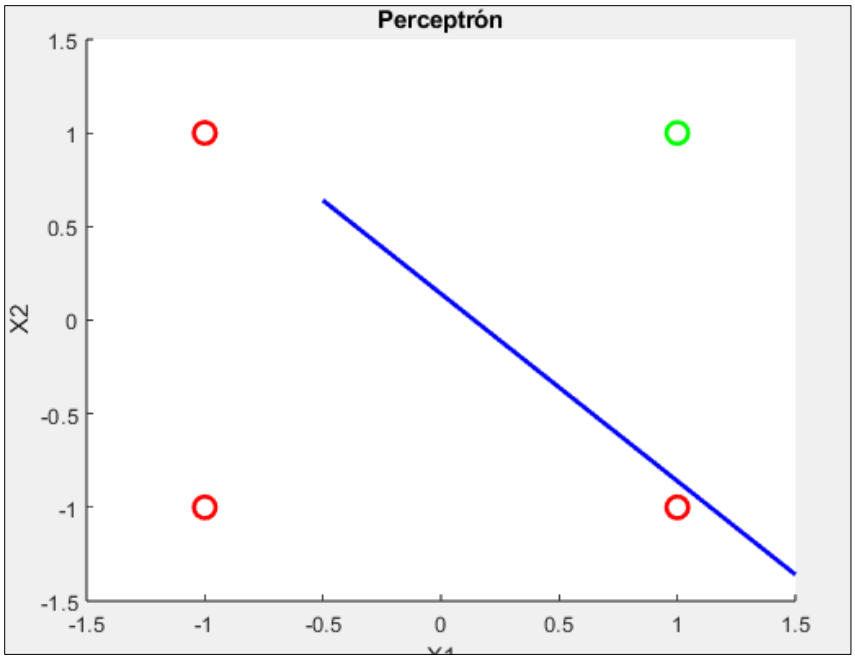
  1. Parámetros versión 1:
     a. Pesos 0.5
     b. Umbral 0

```
Convergencia alcanzada después de 2 iteraciones
   -0.5000    1.5000    1.5000
```



  2. Parámetros versión 2:
     a. Pesos aleatorios
     b. Umbral 0

```
Convergencia alcanzada después de 2 iteraciones
   -0.2463    1.7537    1.7537
```



- Tercera forma de corrección de pesos.

  W = α*[d(x)-Y(x)]*Xi

  1. Parámetros versión 1
     a. Pesos 0.5
     b. Umbral 0

```
Convergencia alcanzada después de 3 iteraciones
   -0.3000    0.5000    0.5000
```

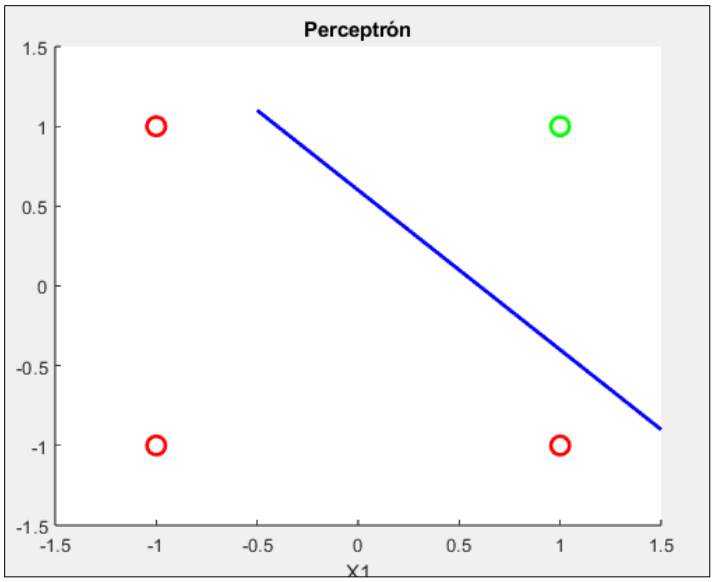


Fig. Clasificación

2. Parámetros versión 2

    a. Pesos aleatorios
    b. Umbral 0.3

```
Convergencia alcanzada después de 2 iteraciones
   -0.0196    0.3804    0.3804
```

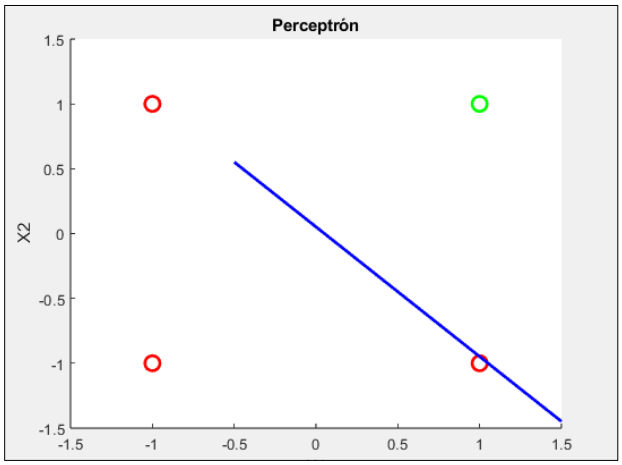


Fig. Clasificación

Tres entradas:

X0	X1	X2	X3	Yd
1	-1	-1	-1	-1
1	-1	-1	1	-1
1	-1	1	-1	-1
1	-1	1	1	-1
1	1	-1	-1	-1
1	1	-1	1	-1
1	1	1	-1	-1
1	1	1	1	1

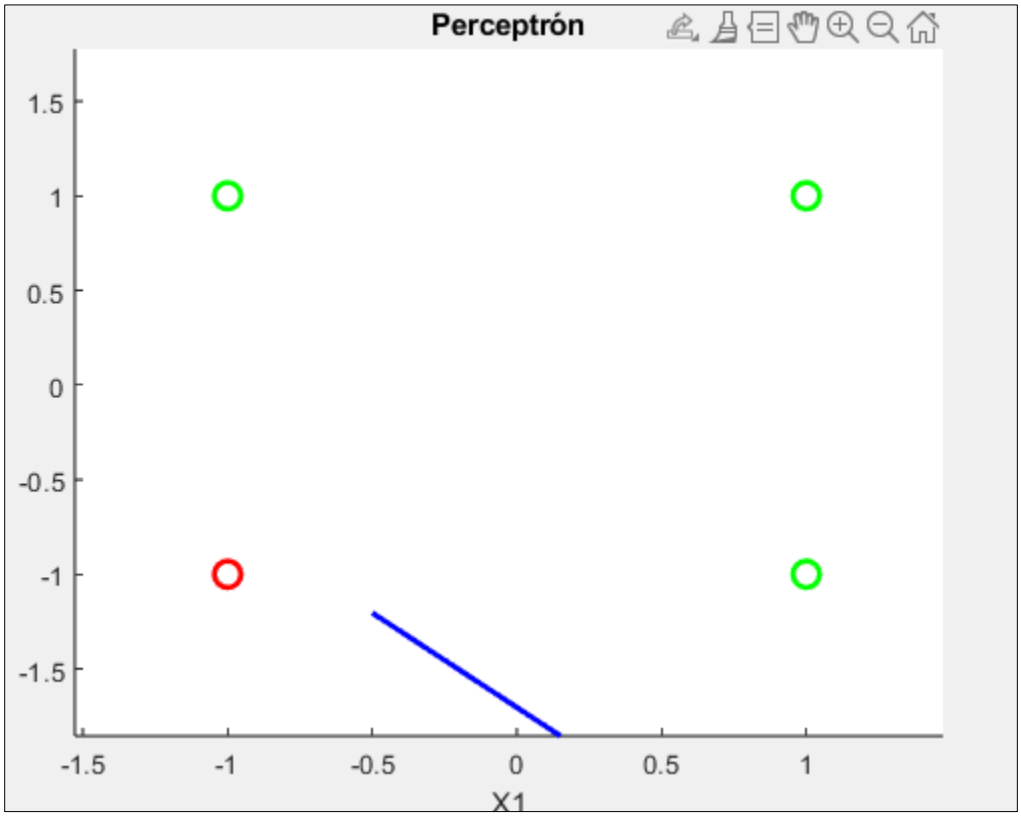
Fig. And tres entradas

- Primera forma de corrección de pesos.

  W = d(x)*Xi

1. Parámetros:
   a. Umbral 0.2
   b. Pesos aleatorios

```
Convergencia alcanzada después de 2 iteraciones
    0.9678    0.5678    0.5678    0.1678
```



- Segunda forma de corrección de pesos.

W = [d(x)-Y(x)]*Xi

1. Parámetros:
    a. Umbral 0
    b. Pesos aleatorios

```
Convergencia alcanzada después de 4 iteraciones
    2.0759    2.0759    2.0759    0.0759
```

- Tercera forma ce corrección de pesos

W = α*[d(x)-Y(x)]*Xi

1. Parámetros
    a. Umbral 0
    b. Pesos aleatorios

```
Convergencia alcanzada después de 1 iteraciones
    0.0540    0.0540    0.0540    0.0540
```

Cuatro entradas:

X0	X1	X2	X3	X3	Yd
1	-1	-1	-1	-1	-1
1	-1	-1	-1	1	-1
1	-1	-1	1	-1	-1
1	-1	-1	1	1	-1
1	-1	1	-1	-1	-1
1	-1	1	-1	1	-1
1	-1	1	1	-1	-1
1	-1	1	1	1	-1
1	1	-1	-1	-1	-1
1	1	-1	-1	1	-1
1	1	-1	1	-1	-1
1	1	-1	1	1	-1
1	1	1	-1	-1	-1
1	1	1	-1	1	-1
1	1	1	1	-1	-1
1	1	1	1	1	1

- Primera forma de corrección de pesos.

W = d(x)*Xi

2. Parámetros:
    a. Umbral 0.2
    b. Pesos aleatorios

```
Convergencia alcanzada después de 3 iteraciones
   -3.0660    0.9340    0.9340    0.9340   -1.0660
```

- Segunda forma de corrección de pesos.

W = [d(x)-Y(x)]*Xi

2. Parámetros:
    a. Umbral 0
    b. Pesos aleatorios

```
Convergencia alcanzada después de 4 iteraciones
   -4.4312    1.5688    1.5688    1.5688   -0.4312
```

- Tercera forma de corrección de pesos

W = α*[d(x)-Y(x)]*Xi

2. Parámetros
    a. Umbral 0
    b. Pesos 0.4

```
Convergencia alcanzada después de 4 iteraciones
   -4.6000    1.4000    1.4000    1.4000   -0.6000
```

Compuerta OR

Dos entradas:

X0	X1	X2	Yd
1	-1	-1	-1
1	-1	1	1
1	1	-1	1
1	1	1	1

- Primera forma de corrección de pesos.

  W = d(x)*Xi

  3. Parámetros versión 1:
     a. Pesos 0.5
     b. Umbral 0

  El perceptrón alcanza una convergencia en dos iteraciones

```
Convergencia alcanzada después de 1 iteraciones
    0.5000    0.5000    0.5000
```

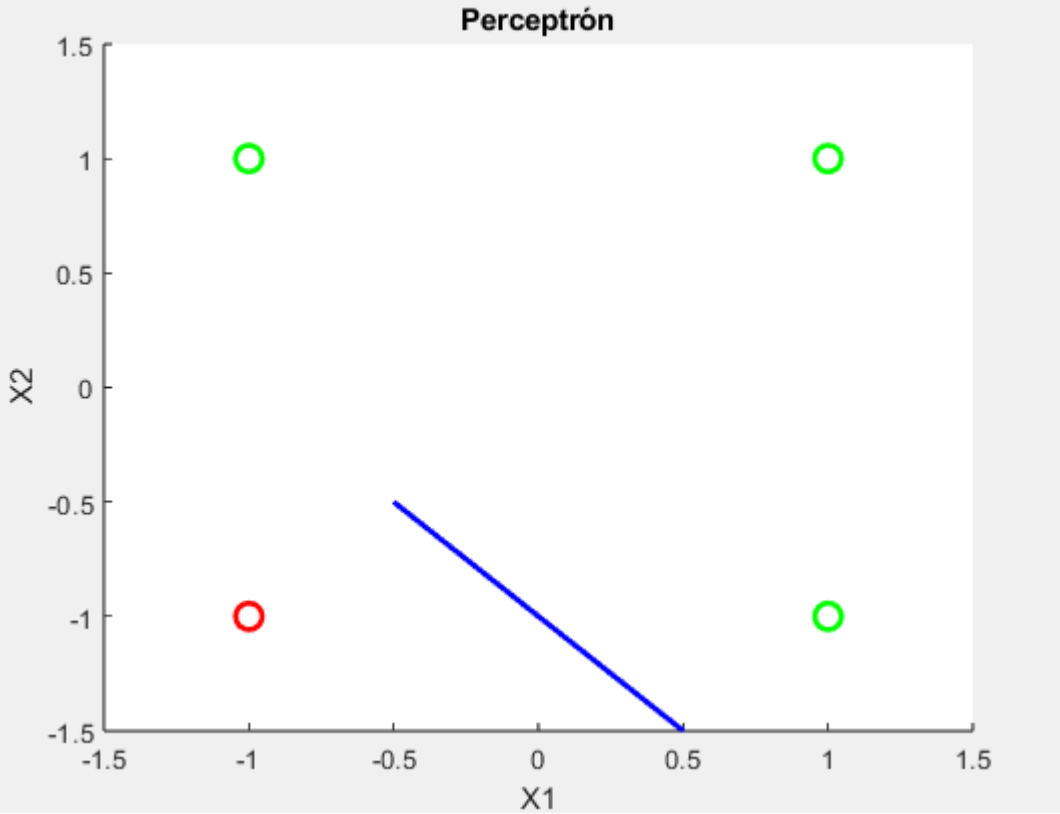


Fig. Línea que divide los diferentes puntos

4. Parámetros versión 2:
    a. Pesos aleatorios
    b. Umbral 0.5

    En esta configuración ocurre que no siempre llega a una convergencia, por lo tanto, los datos no quedan bien clasificados

```
Convergencia alcanzada después de 3 iteraciones
    1.1622    1.1622    1.1622
```

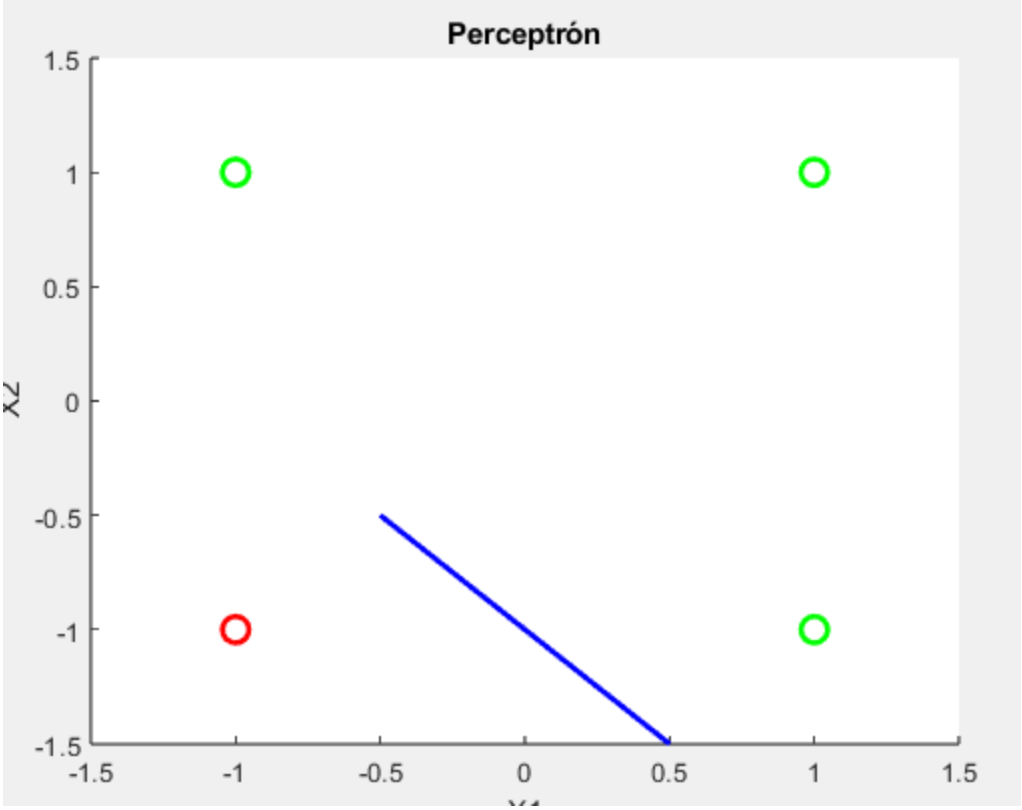


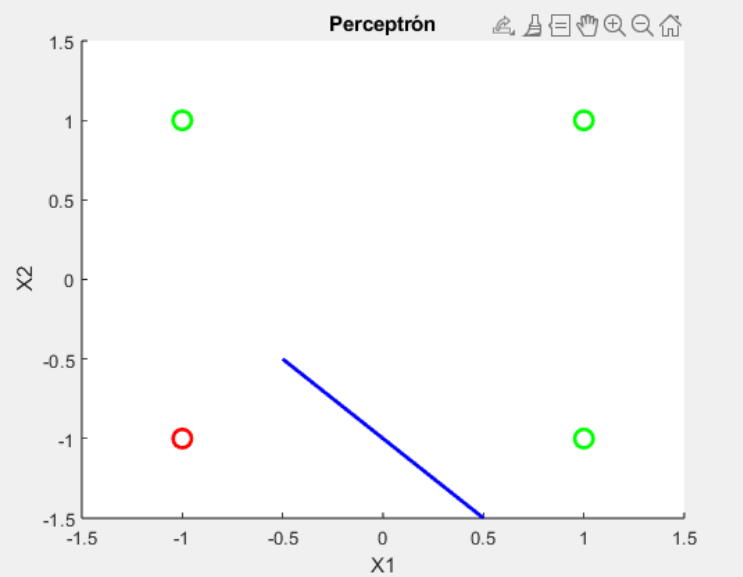
Fig. Clasificación

- Segunda forma de corrección de pesos.

  W = α*[d(x)-Y(x)]*Xi

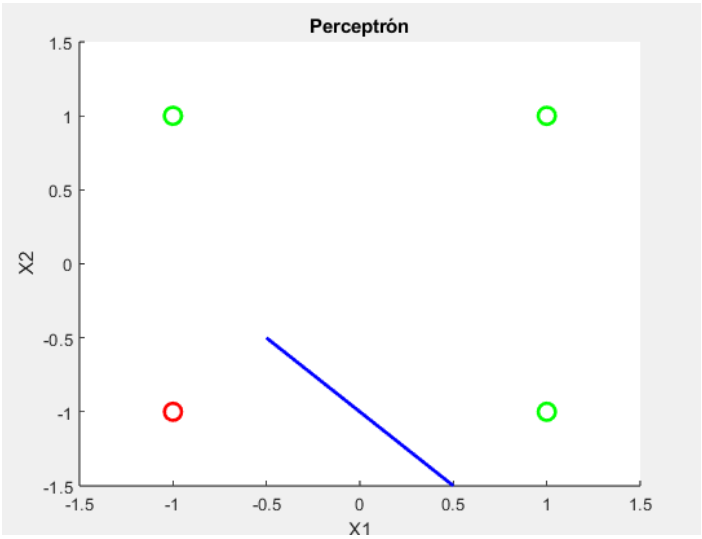
  3. Parámetros versión 1:
     a. Pesos 0.5
     b. Umbral 0

```
Convergencia alcanzada después de 1 iteraciones
    0.5000    0.5000    0.5000
```



4. Parámetros versión 2:
    a. Pesos aleatorios
    b. Umbral 0

```
Convergencia alcanzada después de 1 iteraciones
    0.3112    0.3112    0.3112
```



- Tercera forma de corrección de pesos.

  W = α*[d(x)-Y(x)]*Xi

3. Parámetros versión 1
    a. Pesos 0.5
    b. Umbral 0

```
Convergencia alcanzada después de 1 iteraciones
    0.1656    0.1656    0.1656
```

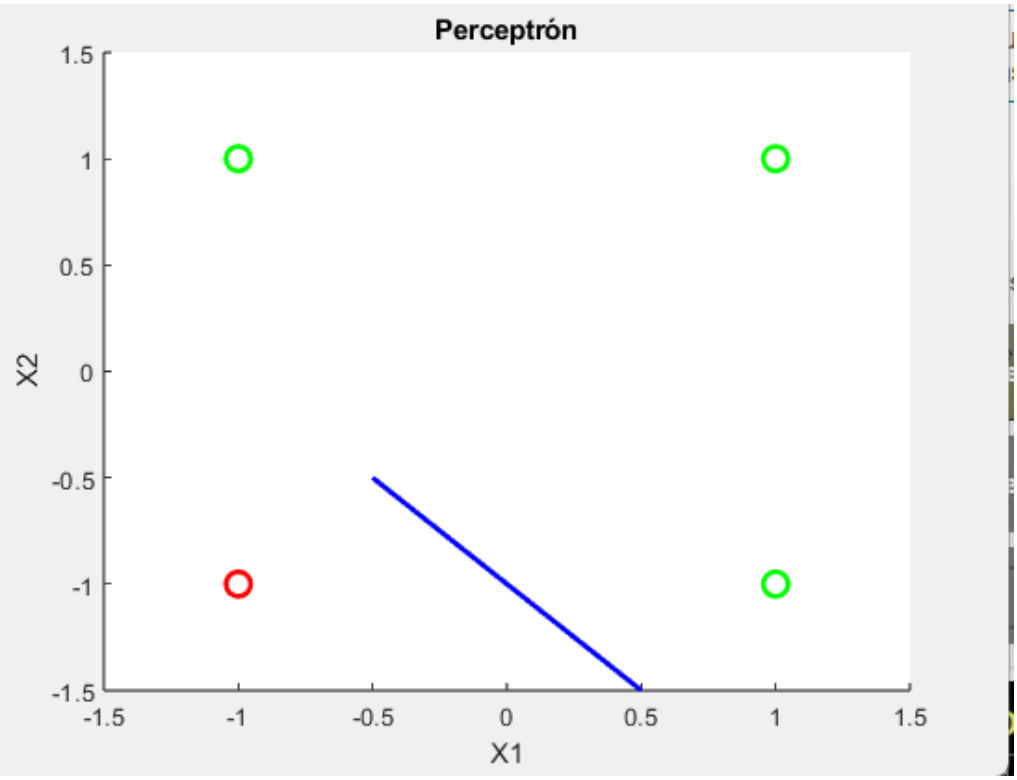


Fig. Clasificación

4. Parámetros versión 2

    a. Pesos aleatorios
    b. Umbral 0.3

```
Convergencia alcanzada después de 1 iteraciones
    0.6020    0.6020    0.6020
```

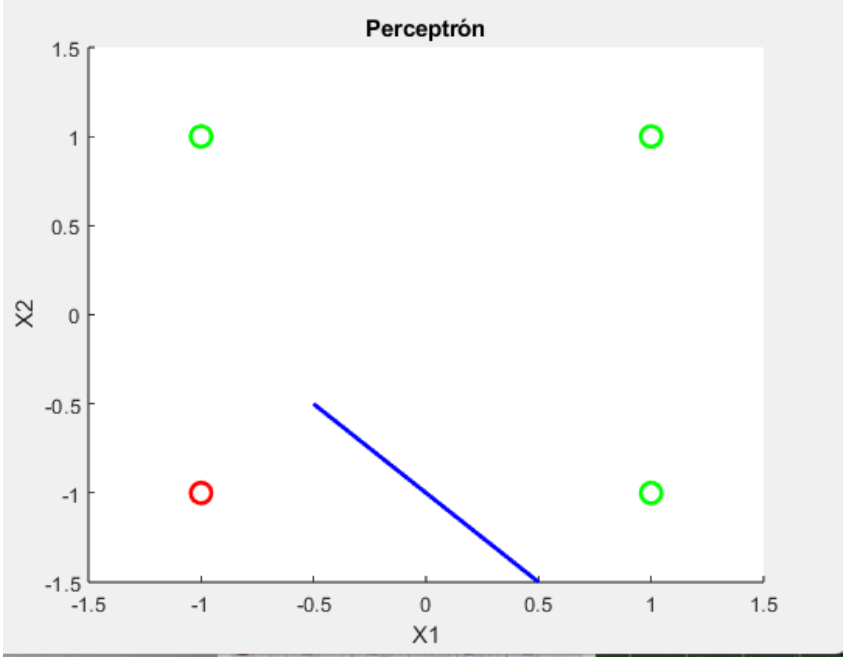


Fig. Clasificación

Tres entradas:

X0	X1	X2	X3	Yd
1	-1	-1	-1	-1
1	-1	-1	1	1
1	-1	1	-1	1
1	-1	1	1	1
1	1	-1	-1	1
1	1	-1	1	1
1	1	1	-1	1
1	1	1	1	1

Fig. Or tres entradas

- Primera forma de corrección de pesos.

  W = d(x)*Xi

3. Parámetros:
   a. Umbral 0.3
   b. Pesos aleatorios

      No converge

```
Convergencia alcanzada después de 100 iteraciones
    1.2630   -0.7370   -0.7370   -0.7370
```

- Segunda forma de corrección de pesos.

  W = [d(x)-Y(x)]*Xi

3. Parámetros:
   a. Umbral 0
   b. Pesos aleatorios

   No converge

```
Convergencia alcanzada después de 100 iteraciones
    2.6541   -1.3459   -1.3459   -1.3459
```

- Tercera forma ce corrección de pesos

  W = α*[d(x)-Y(x)]*Xi

3. Parámetros
   a. Umbral 0
   b. Pesos 0.5

   No converge

```
Convergencia alcanzada después de 100 iteraciones
    0.9000    0.1000    0.1000    0.1000
```

Cuatro entradas:

X0	X1	X2	X3	X3	Yd
1	-1	-1	-1	-1	-1
1	-1	-1	-1	1	1
1	-1	-1	1	-1	1
1	-1	-1	1	1	1
1	-1	1	-1	-1	1
1	-1	1	-1	1	1
1	-1	1	1	-1	1
1	-1	1	1	1	1
1	1	-1	-1	-1	1
1	1	-1	-1	1	1
1	1	-1	1	-1	1
1	1	-1	1	1	1
1	1	1	-1	-1	1
1	1	1	-1	1	1
1	1	1	1	-1	1
1	1	1	1	1	1

- Primera forma de corrección de pesos.

  W = d(x)*Xi

4. Parámetros:
   a. Umbral 0.1
   b. Pesos aleatorios

```
Convergencia alcanzada después de 3 iteraciones
    3.0782    1.0782    1.0782    1.0782   -0.9218
```

- Segunda forma de corrección de pesos.

  W = [d(x)-Y(x)]*Xi

4. Parámetros:
   a. Umbral 0
   b. Pesos aleatorios

```
Convergencia alcanzada después de 3 iteraciones
    4.4427    0.4427    0.4427    0.4427   -3.5573
```

- Tercera forma de corrección de pesos

  W = α*[d(x)-Y(x)]*Xi

4. Parámetros
   a. Umbral 0
   b. Pesos 0.4

```
Convergencia alcanzada después de 3 iteraciones
    6.0046    2.0046    2.0046    2.0046   -1.9954
```

Punto 4

Para este punto ejecuté el archivo que trae Matlab de Iris por defecto ya que tuve problemas con el proporcionado por el profe

Proporción 60-40

```
Probando con α = 0.01
Perceptrón - Precisión: 33.33%
Probando con α = 0.05
Perceptrón - Precisión: 33.33%
Probando con α = 0.10
Perceptrón - Precisión: 33.33%
Probando con α = 0.50
Perceptrón - Precisión: 33.33%
Probando con α = 1.00
Perceptrón - Precisión: 33.33%
```

Proporción 70-30

```
Probando con α = 0.01
Perceptrón - Precisión: 33.33%
Probando con α = 0.05
Perceptrón - Precisión: 33.33%
Probando con α = 0.10
Perceptrón - Precisión: 33.33%
Probando con α = 0.50
Perceptrón - Precisión: 33.33%
Probando con α = 1.00
Perceptrón - Precisión: 33.33%
```

Proporción 80-20

```
Probando con α = 0.01
Perceptrón - Precisión: 34.48%
Probando con α = 0.05
Perceptrón - Precisión: 34.48%
Probando con α = 0.10
Perceptrón - Precisión: 34.48%
Probando con α = 0.50
Perceptrón - Precisión: 34.48%
Probando con α = 1.00
Perceptrón - Precisión: 34.48%
```

Proporción 90-10

```
Probando con α = 0.01
Perceptrón - Precisión: 28.57%
Probando con α = 0.05
Perceptrón - Precisión: 28.57%
Probando con α = 0.10
Perceptrón - Precisión: 28.57%
Probando con α = 0.50
Perceptrón - Precisión: 28.57%
Probando con α = 1.00
Perceptrón - Precisión: 28.57%
```

Punto6

Proporción de entrenamiento: 60-40

Probando con α = 0.01

Modelo: Perceptrón

Perceptrón - Precisión con α = 0.01: 90.15%

Modelo: Regresión Logística

Regresión Logística - Precisión con α = 0.01: 0.36%

Modelo: SVM

SVM - Precisión con α = 0.01: 98.36%

Probando con α = 0.05

Modelo: Perceptrón

Perceptrón - Precisión con α = 0.05: 90.88%

Modelo: Regresión Logística

Regresión Logística - Precisión con α = 0.05: 0.36%

Modelo: SVM

SVM - Precisión con α = 0.05: 98.72%

Probando con α = 0.10

Modelo: Perceptrón

Perceptrón - Precisión con α = 0.10: 91.24%

Modelo: Regresión Logística

Regresión Logística - Precisión con α = 0.10: 0.36%

Modelo: SVM

SVM - Precisión con α = 0.10: 99.64%

Probando con α = 0.50

Modelo: Perceptrón

Perceptrón - Precisión con α = 0.50: 90.88%

Modelo: Regresión Logística

Regresión Logística - Precisión con α = 0.50: 0.36%

Modelo: SVM

SVM - Precisión con α = 0.50: 99.64%

Probando con α = 1.00

Modelo: Perceptrón

Perceptrón - Precisión con α = 1.00: 90.15%

Modelo: Regresión Logística

Regresión Logística - Precisión con α = 1.00: 0.36%

Modelo: SVM

SVM - Precisión con α = 1.00: 99.27%

Proporción de entrenamiento: 70-30

Probando con α = 0.01

Modelo: Perceptrón

Perceptrón - Precisión con α = 0.01: 88.08%

Modelo: Regresión Logística

Regresión Logística - Precisión con α = 0.01: 0.49%

Modelo: SVM

SVM - Precisión con α = 0.01: 98.05%

Probando con α = 0.05

Modelo: Perceptrón

Perceptrón - Precisión con α = 0.05: 91.00%

Modelo: Regresión Logística

Regresión Logística - Precisión con α = 0.05: 0.49%

Modelo: SVM

SVM - Precisión con α = 0.05: 98.05%

Probando con α = 0.10

Modelo: Perceptrón

Perceptrón - Precisión con α = 0.10: 91.48%

Modelo: Regresión Logística

Regresión Logística - Precisión con α = 0.10: 0.49%

Modelo: SVM

SVM - Precisión con α = 0.10: 98.05%

Probando con α = 0.50

Modelo: Perceptrón

Perceptrón - Precisión con α = 0.50: 92.70%

Modelo: Regresión Logística

Regresión Logística - Precisión con α = 0.50: 0.49%

Modelo: SVM

SVM - Precisión con α = 0.50: 99.27%

Probando con α = 1.00

Modelo: Perceptrón

Perceptrón - Precisión con α = 1.00: 90.75%

Modelo: Regresión Logística

Regresión Logística - Precisión con α = 1.00: 0.49%

Modelo: SVM

SVM - Precisión con α = 1.00: 99.27%

Proporción de entrenamiento: 80-20

Probando con α = 0.01

Modelo: Perceptrón

Perceptrón - Precisión con α = 0.01: 94.16%

Modelo: Regresión Logística

Regresión Logística - Precisión con α = 0.01: 0.73%

Modelo: SVM

SVM - Precisión con α = 0.01: 97.45%

Probando con α = 0.05

Modelo: Perceptrón

Perceptrón - Precisión con α = 0.05: 93.80%

Modelo: Regresión Logística

Regresión Logística - Precisión con α = 0.05: 0.73%

Modelo: SVM

SVM - Precisión con α = 0.05: 98.18%

Probando con α = 0.10

Modelo: Perceptrón

Perceptrón - Precisión con α = 0.10: 94.16%

Modelo: Regresión Logística

Regresión Logística - Precisión con α = 0.10: 0.73%

Modelo: SVM

SVM - Precisión con α = 0.10: 98.18%

Probando con α = 0.50

Modelo: Perceptrón

Perceptrón - Precisión con α = 0.50: 94.89%

Modelo: Regresión Logística

Regresión Logística - Precisión con α = 0.50: 0.73%

Modelo: SVM

SVM - Precisión con α = 0.50: 98.91%

Probando con α = 1.00

Modelo: Perceptrón

Perceptrón - Precisión con α = 1.00: 94.89%

Modelo: Regresión Logística

Regresión Logística - Precisión con α = 1.00: 0.73%

Modelo: SVM

SVM - Precisión con α = 1.00: 98.54%

Proporción de entrenamiento: 90-10

Probando con α = 0.01

Modelo: Perceptrón

Perceptrón - Precisión con α = 0.01: 92.70%

Modelo: Regresión Logística

Regresión Logística - Precisión con α = 0.01: 0.73%

Modelo: SVM

SVM - Precisión con α = 0.01: 97.81%

Probando con α = 0.05

Modelo: Perceptrón

Perceptrón - Precisión con α = 0.05: 92.70%

Modelo: Regresión Logística

Regresión Logística - Precisión con α = 0.05: 0.73%

Modelo: SVM

SVM - Precisión con α = 0.05: 97.81%

Probando con α = 0.10

Modelo: Perceptrón

Perceptrón - Precisión con α = 0.10: 91.24%

Modelo: Regresión Logística

Regresión Logística - Precisión con α = 0.10: 0.73%

Modelo: SVM

SVM - Precisión con α = 0.10: 98.54%

Probando con α = 0.50

Modelo: Perceptrón

Perceptrón - Precisión con α = 0.50: 91.97%

Modelo: Regresión Logística

Regresión Logística - Precisión con α = 0.50: 0.73%

Modelo: SVM

SVM - Precisión con α = 0.50: 99.27%

Probando con α = 1.00

Modelo: Perceptrón

Perceptrón - Precisión con α = 1.00: 90.51%

Modelo: Regresión Logística

Regresión Logística - Precisión con α = 1.00: 0.73%

Modelo: SVM

SVM - Precisión con α = 1.00: 98.54%