Trabajo Práctico N°3

Perceptron simple y multicapa

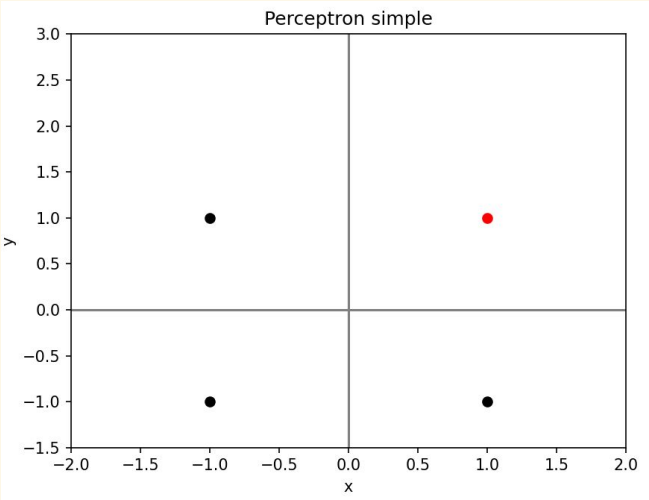
Grupo 14

Ian Mejalelaty
Justina Vacas Castro
Josefina Assaff Alvarez

Ej 1: Perceptron simple escalón

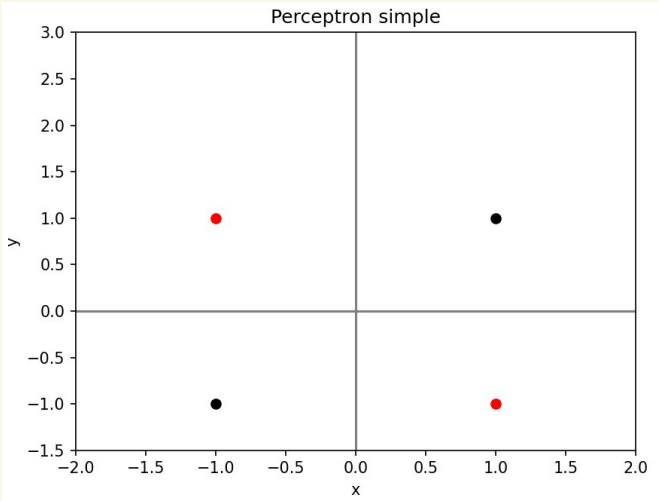
Problemas a analizar con entradas x = {{−1, 1}, {1, −1}, {−1, −1}, {1, 1}}

AND



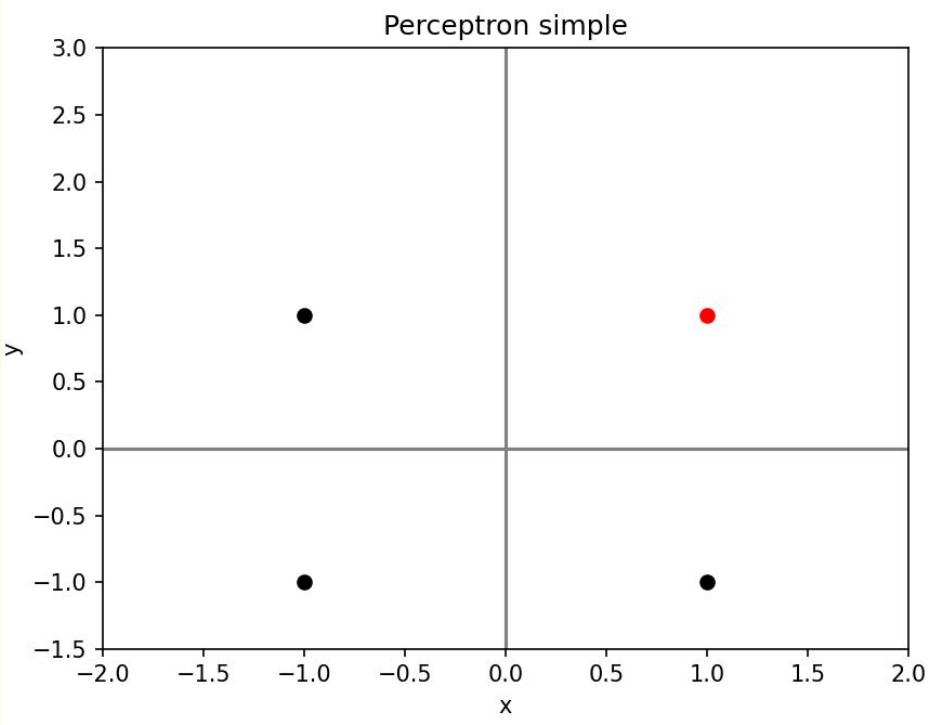
y = {−1, −1, −1, 1}

XOR



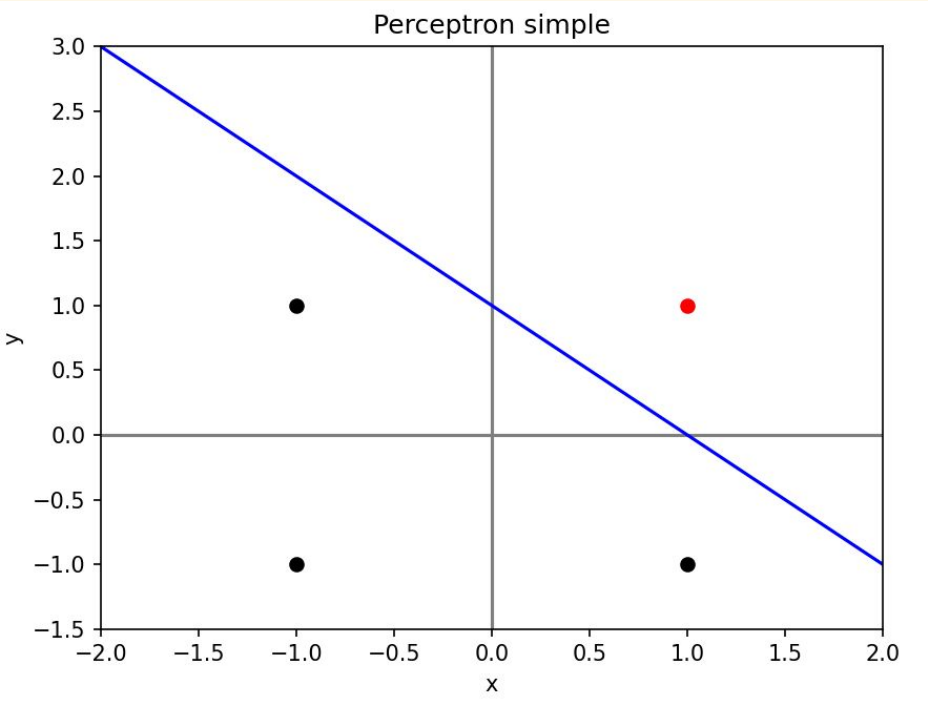
y = {1, 1, −1, −1}

Ej 1: AND



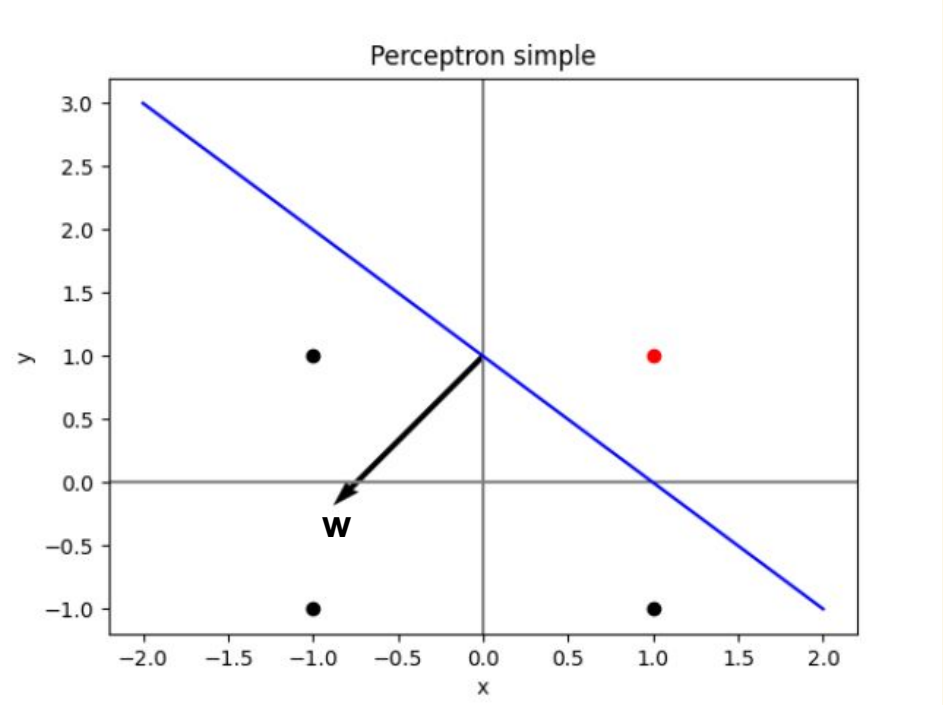
ξ_1	ξ_2	ζ	
-1	1	-1	●
1	-1	-1	●
-1	-1	-1	●
1	1	1	●

Ej 1: AND



$$y = -\frac{w_1}{w_2}x - \frac{w_0}{w_2}$$

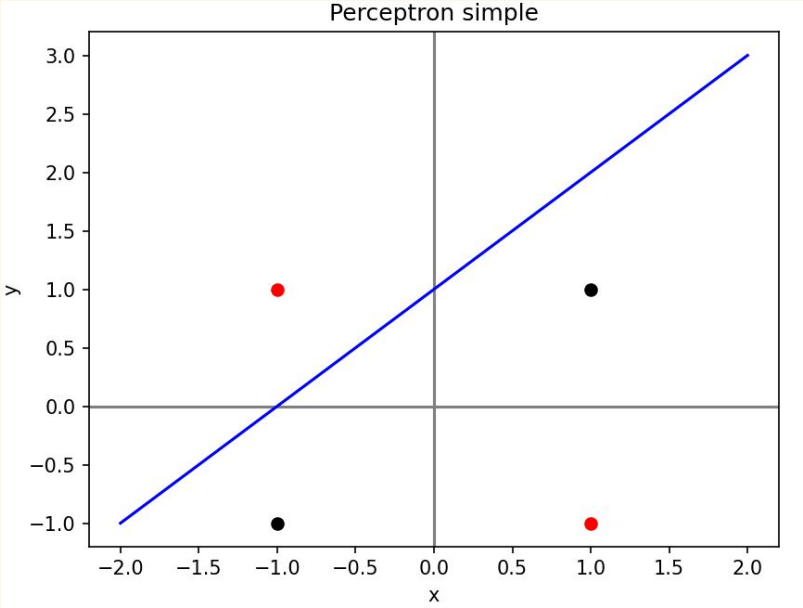
Ej 1: AND

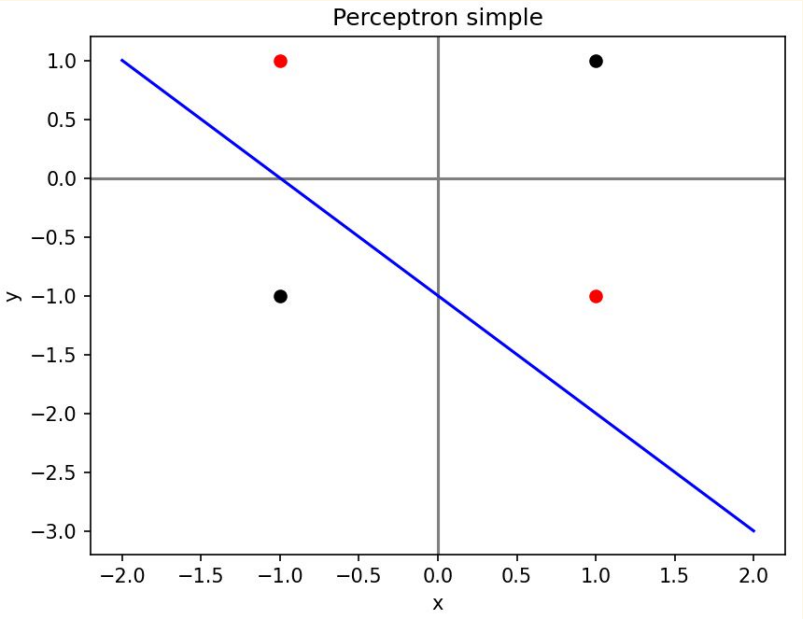


$$y = -\frac{w_1}{w_2}x - \frac{w_0}{w_2}$$

$$w = (-\frac{w_1}{w_2}, -1)$$

Ej 1: OR exclusivo





$$y = -\frac{w_1}{w_2}x - \frac{w_0}{w_2}$$

¿Qué puede decir acerca de los problemas que puede resolver el perceptrón simple escalón en relación a la resolución de los problemas que se le pidió que haga que el perceptrón aprenda?

Ej 2: Perceptron Simple Lineal y No Lineal

Conjunto de entrenamiento: 200 valores de entrada

Ej 2: Perceptron Simple No Lineal

- Función de activación: tangente hiperbólica
- Los valores de salida del conjunto de entrenamiento fueron escalados al intervalo [-1, 1]
- Se normalizaron los valores con la fórmula:

$$2\,\frac{(X - t_{min})}{t_{max} - t_{min}}$$

t_min = valor mínimo del conjunto
t_max = valor máximo del conjunto

Ej 2: Perceptron Simple Lineal y no Lineal

n = 0.01 limit = 1000 beta = 0.7

```
------ Linear Training... --------
Iterations =  1000
Time: 0.390625 s
Error_min = 1883.3664762524215
```

```
------ Not Linear Training... --------
Iterations =  1000
Time: 0.578125 s
Error_min = 54.045676934769745
Error_min (without scale) = 0.02195613930678377
```

n = 0.1 limit = 500 beta = 0.7

```
------ Linear Training... --------
Iterations =  500
Time: 0.21875 s
Error_min = 3240.961136725567
```

```
------ Not Linear Training... --------
Iterations =  500
Time: 0.28125 s
Error_min = 54.833857336710196
Error_min (without scale) = 0.02227633880626953
```

darle mas iteraciones!

Ej 2: Perceptron Simple Lineal y no Lineal

n = 0.01 limit = 1000 beta = 0.7

```
------ Linear Training... --------
Iterations =  1000
Time: 0.390625 s
Error_min = 1883.3664762524215
```

```
------ Not Linear Training... --------
Iterations =  1000
Time: 0.578125 s
Error_min = 54.045676934769745
Error_min (without scale) = 0.02195613930678377
```

n = 0.1 limit = 1000 beta = 0.7

```
------ Linear Training... --------
Iterations =  1000
Time: 0.375 s
Error_min = 2951.764077042306
```

```
------ Not Linear Training... --------
Iterations =  1000
Time: 0.609375 s
Error_min = 60.003097615904935
Error_min (without scale) = 0.024376350613267924
```

Ej 2: Perceptron Simple Lineal y no Lineal

n = 0.01 limit = 1000 **beta = 0.7**

```
------ Linear Training... --------
Iterations =  1000
Time: 0.390625 s
Error_min = 1883.3664762524215
```

```
------ Not Linear Training... --------
Iterations =  1000
Time: 0.578125 s
Error_min = 54.045676934769745
Error_min (without scale) = 0.02195613930678377
```

n = 0.01 limit = 1000 **beta = 1**

```
------ Linear Training... --------
Iterations =  1000
Time: 0.421875 s
Error_min = 1885.081774765427
```

```
------ Not Linear Training... --------
Iterations =  1000
Time: 0.6875 s
Error_min = 53.95046423420098
Error_min (without scale) = 0.02191745900086427
```

Ej 2: Test no lineal

n = 0.01 limit = 1000 **beta = 0.7**

```
------ Not Linear Training... --------
Iterations =  1000
Time: 0.578125 s
Error_min = 54.045676934769745
Error_min (without scale) = 0.02195613930678377
```

```
------ Testing... --------
Test error =  2792.8319672523075
Hits =  22
Success:  44.0 %
```

¿Cómo podría escoger el mejor conjunto de entrenamiento?

¿Cómo podría evaluar la máxima capacidad de generalización del perceptrón para este conjunto de datos?

Ej 3: Perceptron Multicapa