Instituto Politécnico Nacional

Escuela Superior de Cómputo

Asignatura:

Neural Networks

Grupo: 3CM2

Práctica 3. **Red perceptrón**

Alumno:

Garcia Garcia Rafael

Profesor: Moreno Armendariz Marco Antonio

Introducción:

A fines de la década de 1950, Frank Rosenblatt y varios otros investigadores desarrollaron una clase de redes neuronales llamadas perceptrones. Las neuronas en éstas redes eran similares a las de McCulloch y Pitts. La contribución clave de Rosenblatt fue la introducción de una regla de aprendizaje para capacitar a las redes de perceptrón para resolver problemas de reconocimiento de patrones. Con esto se demostró que su regla de aprendizaje siempre convergerá a las ponderaciones de red correctas, si existen ponderaciones que resuelvan el problema. El aprendizaje era simple y automático. Ejemplos de comportamiento apropiado fueron presentados a la red, que aprendió de sus errores. El perceptrón podría incluso aprender cuando se inicializa con valores aleatorios para los pesos sinápticos y bias.

Desafortunadamente, la red de perceptrón era limitada. Estas limitaciones se publicaron ampliamente en el libro *Perceptrons* de Marvin Minsky y Seymour Papert. Demostraron que la red perceptrón era incapaz de implementar ciertas funciones elementales. Dicho suceso marco el principio de la caída de las redes neuronales, lo que ocasionó que muchos científicos abandonaran la rama y a continuación se suspendieran las investigaciones de las redes neuronales por aproximadamente 10 años.

A continuación, se podrá observar una descripción de la arquitectura de dicho perceptrón y una de las muchas maneras en las que éste se puede implementar para la resolución de ciertos problemas.

Marco teórico.

Explicación de la arquitectura,

La arquitectura del perceptrón se muestra en la figura 1.

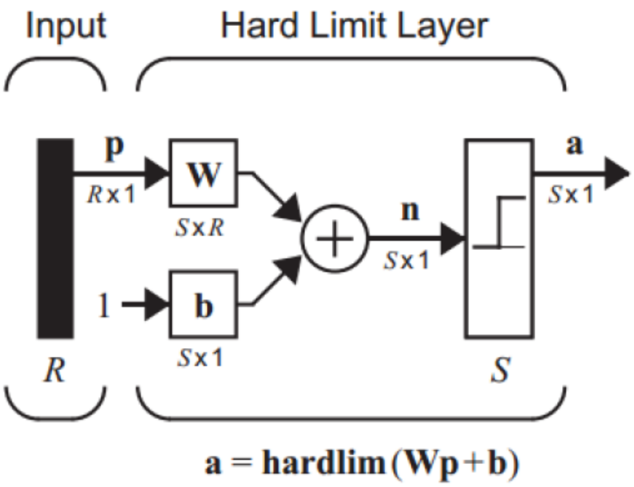


Figura 1.

Cómo se aprecia, la salida del perceptrón consta de una función hard limit, lo que indica salidas binarias. Se puede apreciar que la red perceptrón puede constar de muchas neuronas sin embargo solo tendrá una sola capa.

Las conexiones entre las neuronas de entrada y la capa de neuronas asociativa son fijas y no se modifican durante el período de entrenamiento. En cambio, los pesos sinápticos entre las neuronas asociativas y las de respuesta se alteran según una regla específica, de la siguiente manera: Cada vez que se presenta un patrón **p**, cada una de las neuronas de respuesta, $\mathbf{S_i}$ proporciona una salida, (0, 1) que puede ser correcta o no. De ello depende la forma como se alteran los pesos sinápticos, $\mathbf{W_{ij}}$, de las conexiones entre cada unidad asociativa, a¡, con las neuronas de respuesta, $\mathbf{S_i}$.

Si la salida $\mathbf{a_i}$ es correcta, los pesos no se alteran, pero si no lo es, entonces cada $\mathbf{W_{ij}}$ se incrementa en una cantidad, para cada neurona asociadora $\mathbf{S_i}$ activa, es decir, cuya salida sea 1, siendo $\mathbf{P_i}$ la salida deseada de la i-ésima neurona de respuesta.

Entonces, para cada patrón p presentado a las neuronas de entrada, se tiene que:

$$\mathbf{W_{ij} = W_{ij}+(t_i-a_i)*(P_i)}$$

Donde:

$\mathbf{W_{ij}}$: Valores de pesos sinápticos.
$\mathbf{T_i}$: Target.
$\mathbf{A_i}$: Salida de la red.
$\mathbf{P_i}$: Entrada de la red
Y de manera más sencilla:

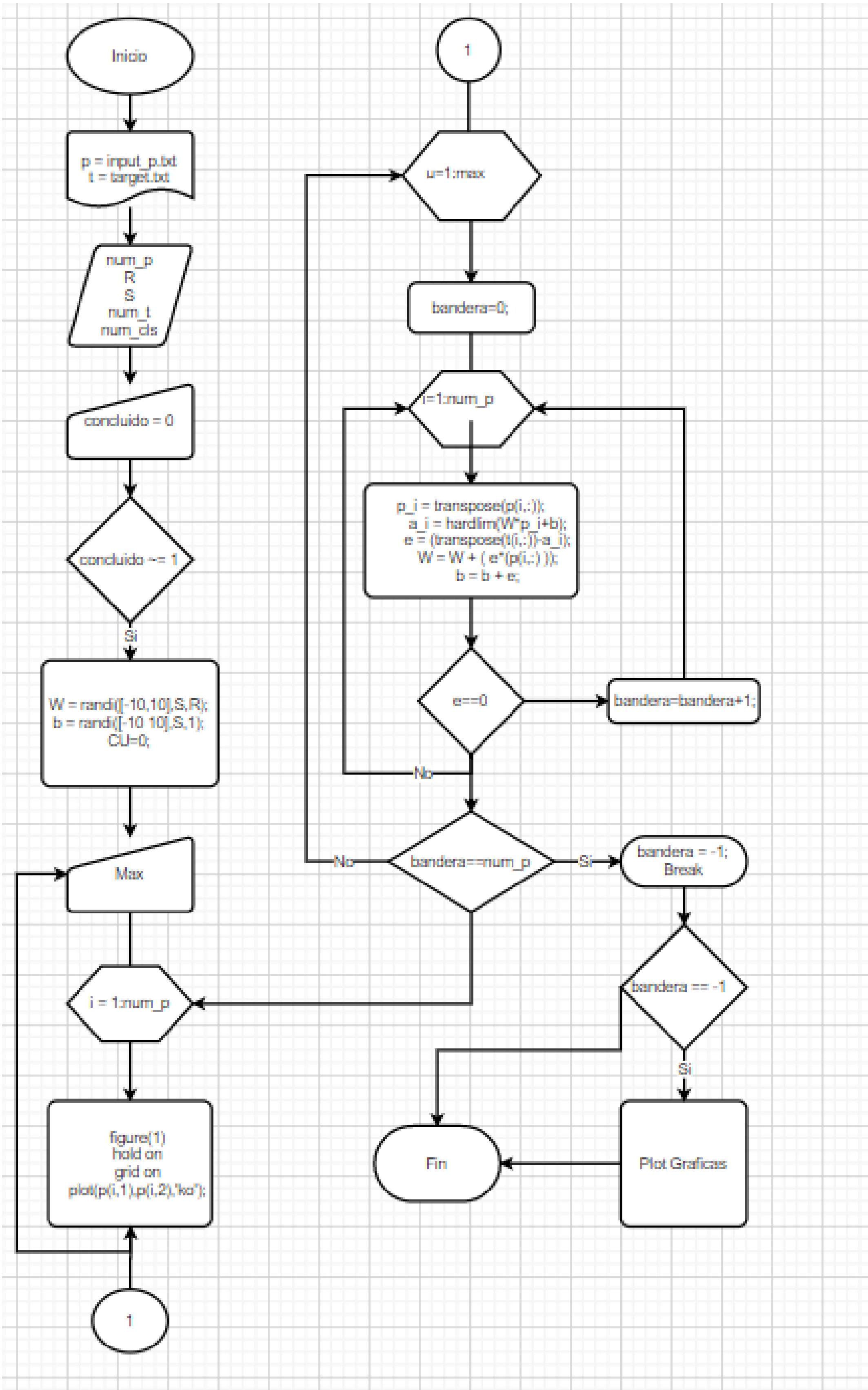
$${}_1\mathbf{w}^{new} = {}_1\mathbf{w}^{old} + e\mathbf{p}$$

Donde podemos decir que e=t-a, o dicho de manera más figurativa: El error.

Esta regla se puede ampliar para entrenar al bias al observar que el bias es simplemente un peso cuya entrada es siempre 1. Por lo tanto, podemos reemplazar la entrada en la ecuación:

$$b^{new} = b^{old} + e\,.$$

Diagrama de Flujo:



Experimentos.

Experimento 1:

Este experimento fue el realizado en clase, donde se buscaba generar las salidas de una compuerta AND mediante un perceptrón.

El conjunto de entrada fue el siguiente:

$$\left\{\mathbf{p}_1 = \begin{bmatrix}0\\0\end{bmatrix}, t_1 = 0\right\} \left\{\mathbf{p}_2 = \begin{bmatrix}0\\1\end{bmatrix}, t_2 = 0\right\} \left\{\mathbf{p}_3 = \begin{bmatrix}1\\0\end{bmatrix}, t_3 = 0\right\} \left\{\mathbf{p}_4 = \begin{bmatrix}1\\1\end{bmatrix}, t_4 = 1\right\}.$$

Donde la salida de nuestro programa fue el siguiente código y su representación gráfica de los pesos (rojo) y frontera de decisión (azul):

```
>> practica_3
El numero de rasgos es: 2
El numero de clases ingresadas es: 2
El numero de neuronas necesarias es: 1
    7     9
Ingrese el numero de MAX_EPOCH:5
Epoca num: 1
Epoca num: 2
Aprendizaje Correcto!!!
7        8        (W)
-9                (B)
Teclee 1 para salir/0 para volver a intentar: 1
```

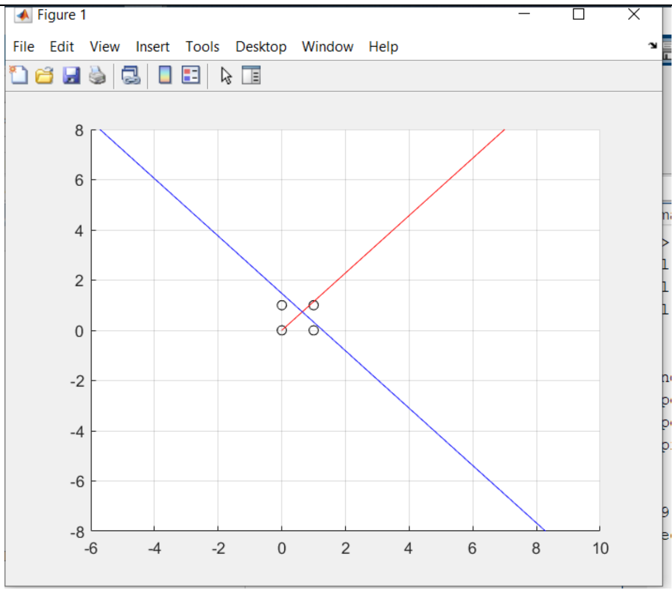


Figura 2

Experimento 2:

Este experimento fue con una entrada con 4 clases donde el número de neuronas fue 2. Se buscaba tener diferentes vectores prototipo con la misma salida t en donde varios pertenecían a un conjunto especifico,

$$\text{class 1:}\left\{\mathbf{p}_1 = \begin{bmatrix}1\\1\end{bmatrix}, \mathbf{p}_2 = \begin{bmatrix}1\\2\end{bmatrix}\right\}, \text{class 2:}\left\{\mathbf{p}_3 = \begin{bmatrix}2\\-1\end{bmatrix}, \mathbf{p}_4 = \begin{bmatrix}2\\0\end{bmatrix}\right\},$$

$$\text{class 3:}\left\{\mathbf{p}_5 = \begin{bmatrix}-1\\2\end{bmatrix}, \mathbf{p}_6 = \begin{bmatrix}-2\\1\end{bmatrix}\right\}, \text{class 4:}\left\{\mathbf{p}_7 = \begin{bmatrix}-1\\-1\end{bmatrix}, \mathbf{p}_8 = \begin{bmatrix}-2\\-2\end{bmatrix}\right\}.$$

$$\text{class 1:}\left\{\mathbf{t}_1 = \begin{bmatrix}0\\0\end{bmatrix}, \mathbf{t}_2 = \begin{bmatrix}0\\0\end{bmatrix}\right\}, \text{class 2:}\left\{\mathbf{t}_3 = \begin{bmatrix}0\\1\end{bmatrix}, \mathbf{t}_4 = \begin{bmatrix}0\\1\end{bmatrix}\right\},$$

$$\text{class 3:}\left\{\mathbf{t}_5 = \begin{bmatrix}1\\0\end{bmatrix}, \mathbf{t}_6 = \begin{bmatrix}1\\0\end{bmatrix}\right\}, \text{class 4:}\left\{\mathbf{t}_7 = \begin{bmatrix}1\\1\end{bmatrix}, \mathbf{t}_8 = \begin{bmatrix}1\\1\end{bmatrix}\right\}.$$

La red los logró seprar de la siguiente manera y tuvimos la siguiente representación de la matriz de pesos y bias.

```
>> practica_3
El numero de rasgos es: 2
El numero de clases ingresadas es: 4
El numero de neuronas necesarias es: 2
   -8    -5
   -7    -2

Ingrese el numero de MAX_EPOCH:100
Epoca num: 1
Epoca num: 2
Epoca num: 3
Aprendizaje Correcto!!!

-10     -1
1       -8
-7
6
Teclee 1 para salir/0 para volver a intentar: 1
>>
```

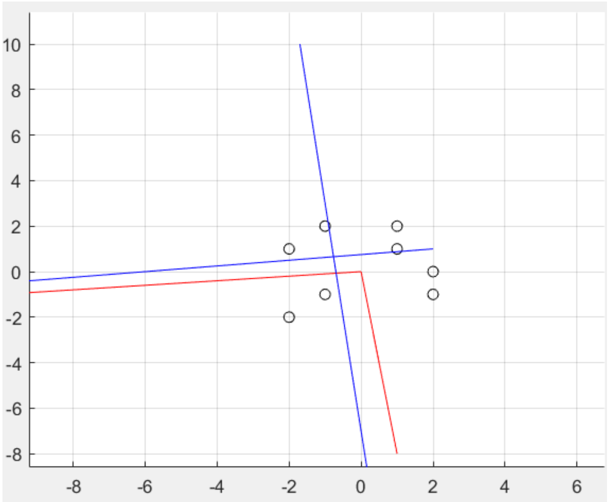


Figura 3

Experimento 3:

Este es el último que se realizo teniendo un total de 6 clases las cuales fueron etiquetadas de la siguiente manera:

$$P_1 = \begin{bmatrix}0\\8\end{bmatrix};\ P_2 = \begin{bmatrix}2\\8\end{bmatrix};\ P_3 = \begin{bmatrix}4\\4\end{bmatrix};\ P_4 = \begin{bmatrix}6\\2\end{bmatrix};R = 2$$
$$P_5 = \begin{bmatrix}-6\\2\end{bmatrix};\ P_6 = \begin{bmatrix}-6\\-2\end{bmatrix};\ P_7 = \begin{bmatrix}-6\\6\end{bmatrix};\ P_8 = \begin{bmatrix}-6\\4\end{bmatrix}$$
$$P_9 = \begin{bmatrix}-4\\-4\end{bmatrix};\ P_{10} = \begin{bmatrix}-2\\-4\end{bmatrix};\ P_{11} = \begin{bmatrix}2\\2\end{bmatrix};\ P_{12} = \begin{bmatrix}2\\-1\end{bmatrix}$$

Donde:

$$P_1, P_2 \to t_1 \begin{bmatrix}0\\1\\1\end{bmatrix}\ P_3, P_4 \to t_2 \begin{bmatrix}1\\1\\1\end{bmatrix}$$
$$P_5, P_6 \to t_3 \begin{bmatrix}0\\0\\0\end{bmatrix}\ P_7, P_8 \to t_4 \begin{bmatrix}0\\1\\0\end{bmatrix}$$
$$P_9, P_{10} \to t_5 \begin{bmatrix}1\\0\\0\end{bmatrix}\ P_{11}, P_{12} \to t_6 \begin{bmatrix}1\\1\\0\end{bmatrix}$$

En este caso al ser introducidos los valores en el perceptrón al ser un número mucho más grande el primer experimento falló

```
practica_3
El numero de rasgos es: 2
El numero de clases ingresadas es: 6
El numero de neuronas necesarias es: 3
    10    -8
     0    -2
     6     9

Ingrese el numero de MAX_EPOCH:6
```

```
Epoca num: 1
Epoca num: 2
Epoca num: 3
Epoca num: 4
Epoca num: 5
Epoca num: 6
Aprendisaje fallido. Teclee 1 para salir/0 para volver a intentar
```

Se presiona el botón 1 para volver a intentarlo:

Y tuvimos una mejor suerte con los valores de salida teniendo una representación gráfica como la siguiente:

```
El numero de rasgos es: 2
El numero de clases ingresadas es: 6
El numero de neuronas necesarias es: 3
    -9    -2
     5     1
    -5     9
Epoca num: 1
Epoca num: 2
Epoca num: 3
Epoca num: 4
Epoca num: 5
Epoca num: 6
Epoca num: 7
Epoca num: 8
Epoca num: 9
Epoca num: 10
Epoca num: 11
Epoca num: 12
Epoca num: 13
Epoca num: 14
Epoca num: 15
Epoca num: 16
Epoca num: 17
Epoca num: 18
Epoca num: 19
Epoca num: 20
Epoca num: 21
Epoca num: 22
Epoca num: 23
Epoca num: 24
Epoca num: 25
Epoca num: 26
Epoca num: 27
Epoca num: 28
Epoca num: 29
Epoca num: 30
Epoca num: 31
Epoca num: 32
Epoca num: 33
Epoca num: 34
Epoca num: 35
Epoca num: 36
Epoca num: 37
Epoca num: 38
```

```
Epoca num: 39
Epoca num: 40
Epoca num: 41
Epoca num: 42
Epoca num: 43
Epoca num: 44
Epoca num: 45
Epoca num: 46
Epoca num: 47
Epoca num: 48
Epoca num: 49
Epoca num: 50
Epoca num: 51
Epoca num: 52
Epoca num: 53
Epoca num: 54
Epoca num: 55
Epoca num: 56
Epoca num: 57
Epoca num: 58
Epoca num: 59
Epoca num: 60
Epoca num: 61
Epoca num: 62
Epoca num: 63
Epoca num: 64
Epoca num: 65
Epoca num: 66
Epoca num: 67
Epoca num: 68
Epoca num: 69
Epoca num: 70
Epoca num: 71
Epoca num: 72
Epoca num: 73
Epoca num: 74
Epoca num: 75
Epoca num: 76
Epoca num: 77
Epoca num: 78
Epoca num: 79
Epoca num: 80
Epoca num: 81
Epoca num: 82
Epoca num: 83
Aprendizaje Correcto!!!

5       -5
5       7
1       1
0
4
-5
Teclee 1 para salir/0 para volver a intentar: 1
>>
```

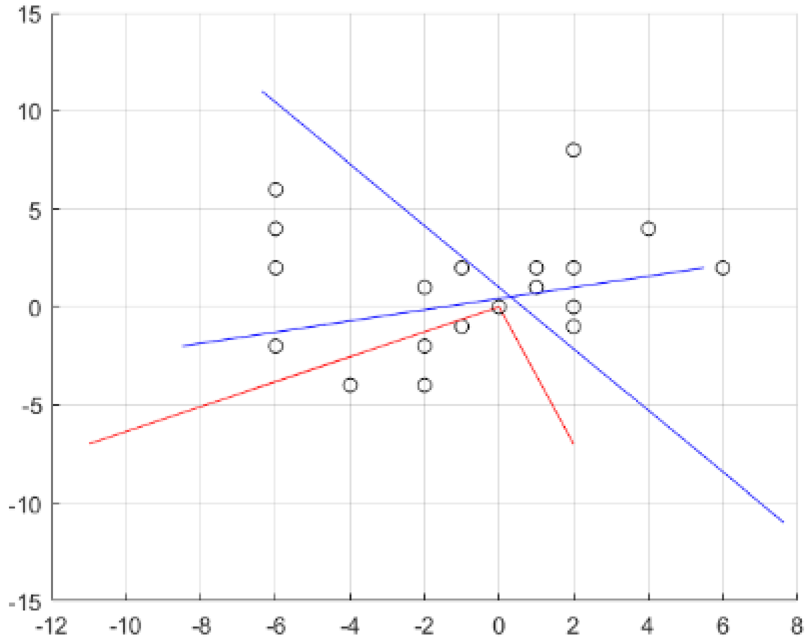


Figura 3.

Como se puede ver se logró separar de manera correcta los valores de pesos y bias para que se pudiera llegar a converger, pero algo curioso de esta red fue de que hubo muchos experimentos fallidos

Discusión:

Es evidente que el programa tiende a graficar las fronteras de desición de manera errática ya que depende mucho del vector de pesos sinápticos para imprimir su longitud, sin embargo eso no afecta al momento del aprendizaje, solamente en el ámbito visual lo que impide ver con claridad resultados en caso de tener que dividir en 6 o más clases, por ejemplo.

Conclusiones

Esta red es perfecta para poder comenzar a entender las reglas de aprendizaje que se aproximan, ya que ésta red es una de las más importantes tanto por su peso historico como su aportación científica así como marcar el antes y el después del declive de las redes neuronales. A pesar de tener una regla de aprendizaje sencilla y no funcional con la compuerts XOR, ésta regla de aprendizaje demostró que muchas otras clases de problemas podían ser solucionados facilmente.

Es de agradacer el hecho de poder comprender al perceptrón para así poder dar paso a las demás redes como: ADALINE o el ya mencionado de manera periódica en clase; el perceptró multicapa.

Bibliografía

Hagan, M. T. (s.f.). Neural Network Design. En M. T. Hagan, *Neural Network Design* (pág. 1010).

Pereira, U. T. (2000). *http://www.medicinaycomplejidad.org/pdf/redes/Competitivas.pdf*. Obtenido de medicinaycomplejidad.