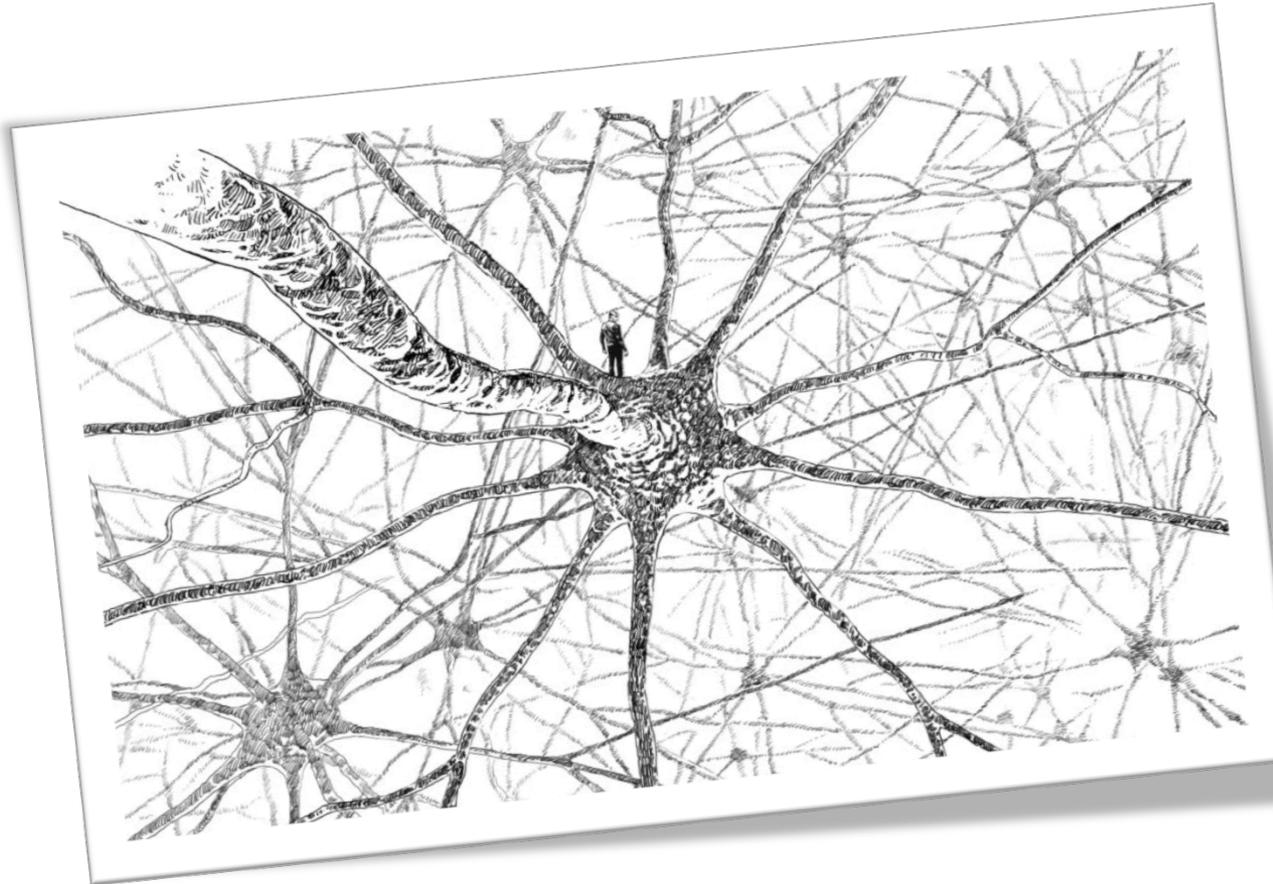


Inteligencia Artificial (EC5)

Ciclo Lectivo 2017

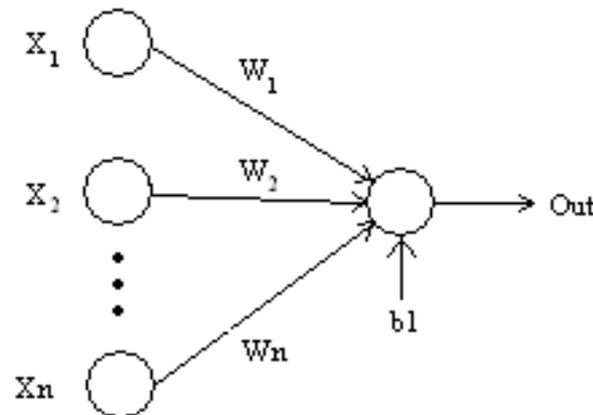




Práctica - Implementación de Perceptron en Matlab

Las Redes Neuronales pueden ser usadas para determinar relaciones y patrones entre las entradas y las salidas.

Una Red Neuronal conocida como Perceptron es una Monocapa – Feed Forward tiene la habilidad de aprender y diferenciar conjuntos de datos





Práctica - Implementación de Perceptron en Matlab

Para comprender el funcionamiento de una Red Perceptron simple, implementaremos una solución para 2 variables de entrada del tipo OR.

Definimos X_1 y X_2 de la siguiente manera:

X_1	X_2	OUT
0	0	0
1	0	1
0	1	1
1	1	1



Práctica - Implementación de Perceptron en Matlab

Comenzaremos por inicializar nuestras variables de interés, incluyendo las entradas, las salidas deseadas, coeficientes de aprendizaje y pesos.

```
input = [0 0; 0 1; 1 0; 1 1];  
numIn = 4;  
desired_out = [0;1;1;1];  
bias = -1;  
coeff = 0.7;  
rand('state',sum(100*clock));  
weights = -1*2.*rand(3,1);
```

Las **entradas y salidas** deseadas deben ser capaces de explicar los resultados que alcanzan, y podrán tomar valores entre 1 y -1, diferentes de cero. EL **bias** se lo define con un valor constante. **Coeff** Representa la tasa de aprendizaje, e indica cuan grande serán los ajustes necesarios de realizar después de cada iteración. Si el coeficiente se aproxima a 1, los pesos se ajustaran de manera mas conservadora. Los **pesos** se asignan aleatoriamente.



Práctica - Implementación de Perceptron en Matlab

La ecuación que define una red perceptron es

$$Net = \sum_{i=0}^n x_i w_i + b$$

Para el ejemplo que se desarrolla, tendríamos:

$$w_1 * x_1 + w_2 * x_2 + b = out$$

Asumiremos que los pesos (1, 1), para los bias y pesos (2; 3; 1) son para X_1 y X_2 respectivamente.



Práctica - Implementación de Perceptron en Matlab

Una de las variable mas es el numero de iteraciones, la cual especifica cuantas veces se entrenara o se modificaran los pesos.

```
iterations = 10;
```

El codigo para la red Perceptron Feed Forward es:

```
for i = 1:iterations
    out = zeros(4,1);
    for j = 1:numIn
        y = bias*weights(1,1)+...
            input(j,1)*weights(2,1)+input(j,2)*weights(3,1);
        out(j) = 1/(1+exp(-y));
        delta = desired_out(j)-out(j);
        weights(1,1) = weights(1,1)+coeff*bias*delta;
        weights(2,1) = weights(2,1)+coeff*input(j,1)*delta;
        weights(3,1) = weights(3,1)+coeff*input(j,2)*delta;
    end
end
```



Práctica - Implementación de Perceptron en Matlab

En primer lugar, la resolución de la ecuación de 'salidas' se determina como se mencionó anteriormente, y luego se ejecuta a través de una función sigmoïdal para asegurar que los valores son comprendidos entre [0 1]. Los Pesos se modifican iterativamente partiendo de la regla delta.

Cuando se ejecuta el perceptrón más de 10 iteraciones, las salidas empiezan a converger, pero todavía no son precisamente como se esperaba:

```
out =  
    0.3756  
    0.8596  
    0.9244  
    0.9952  
weights =  
    0.6166  
    3.2359  
    2.7409
```



Práctica - Implementación de Perceptron en Matlab

Cuando nos aproximamos a las 1000 iteraciones, las salidas convergen hacia la salida esperada

out =

0.0043

0.9984

0.9987

1.0000

weights =

5.4423

12.1084

11.8823



Práctica - Implementación de Perceptron en Matlab

Como la condición lógica de OR es linealmente separable, se alcanzará una solución después de un número finito de ciclos o bucles.

El tiempo de la convergencia también puede cambiar en base a los pesos iniciales, a la tasa de aprendizaje, a la función de transferencia (como sigmoide, lineal, etcetera) y a la regla de aprendizaje (en este caso se utiliza la regla delta, pero también existen otros algoritmos como el de Levenberg-Marquardt).

Mientras que los perceptrones de una sola capa como este pueden resolverse con datos simples linealmente separables, no son adecuados para datos no separables, como el XOR. Con el fin de obtener un conjunto de datos, usted necesitará utilizar un perceptrón multicapa.



Práctica - Implementación de Perceptron en Matlab

En

REFERENCIAS



Libros

*Redes Neuronales y Sistemas Difusos / Bonifacio Martin del Brío y Alfredo Sanz Molina.
Alfa Omega – Rama. Colombia/c.2005*

Sitios Web

<http://matlabgeeks.com/tips-tutorials/neural-networks-a-perceptron-in-matlab/>

<http://proton.ucting.udg.mx/posgrado/cursos/idc/neuronales2/Transferencia.htm>



Merci
Gracias
Thank You
Danke
köszönöm
σας ευχαριστώ
pakkai për
Gratias agimus tibi
Grazie
Tak
благодаря
اركش
ありがとう