

Instituto Politécnico Nacional

Escuela Superior de Cómputo

Neural Network

3CM1

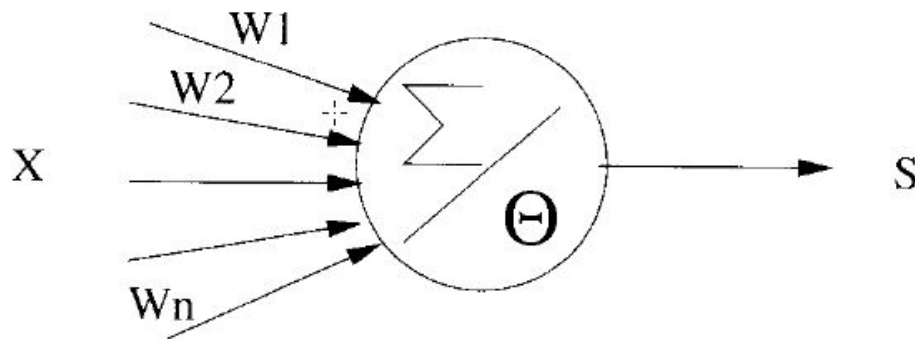
Practica 1: Célula de McCulloch–Pitts

Hernández López Ángel Zait

2014080682

Ciclo escolar: 2019/2

21 de febrero de 2019



Introducción:

Las redes neuronales son muy importantes dentro de la Inteligencia Artificial. Ya que estos están inspirados en el comportamiento del cerebro humano, principalmente a las neuronas y sus conexiones, ya que se trata de crear modelos artificiales que solucionen problemas difíciles de resolver con distintos tipos de algoritmos. La arquitectura de una RNA es la estructura o patrón de conexiones de la red. Las conexiones sinápticas son direccionales, es decir, la información sólo se transmite en un sentido. En general, las neuronas suelen agruparse en unidades estructurales llamadas capas. Dentro de una capa, la neurona suelen ser del mismo tipo.

Ya entrando más en tema y sabiendo que es una red neuronal, esta práctica tiene como objetivo desarrollar un programa que imite el funcionamiento de la célula de McCulloch-Pitts, ya que fue una de las primeras propuestas estructurales de las redes neuronales.

Las tareas que debe realizar es aprender las tablas de verdad de las compuertas lógicas más conocidas, las cuales son: NOT, AND y OR. Teniendo en cuenta la tabla de verdad de cada uno de las compuertas. Con la pequeña diferencia de que las compuertas AND y OR deber recibir n entradas que desee el usuario.

Como sabemos, podemos hacer una serie de operaciones para poder identificar si la célula aprende de manera correcta la tabla de verdad con los pesos sinápticos y el umbral, pero ese no es el objetivo de esta práctica, ya que sacaremos los valores de cada uno de los datos de n y a , para poder identificar si está funcionando correctamente o no la célula.

Marco Teórico.

En esta práctica, vimos la célula de McCulloch-Pitts, podemos decir que puede ser una red neuronal artificial, por el tipo de estructura que tiene. Este modelo de neurona fue propuesto por Warren McCulloch y Walter Pitts en 1943. En él modelizaban una estructura y un funcionamiento simplificado de las neuronas del cerebro, considerándose como dispositivos con sólo dos estados posibles, apagado y encendido (0 y 1 respectivamente).

La célula recibe como entrada un conjunto de n valores binarios, procedentes de las salidas de otras células, o de la entrada a la red; y produce una única salida también binaria, s . Cada célula se caracteriza por $n + 1$ valores reales, de los cuales n son los pesos de las conexiones (w_i) correspondientes a las entradas x_i , y otro es un valor de umbral θ , que puede ser distinto para cada célula. La célula opera en lapsos discretos. La forma de procesar la entrada es la siguiente: la célula se activará y, por lo tanto, producirá un valor 1, si la suma de las entradas multiplicadas por los pesos supera al umbral θ .

El modelo matemático es:

$$n = \sum_{i=1}^R w_i p_i; \quad a = f(n)$$

Donde:

$$a = 1 \text{ si } n > \theta;$$

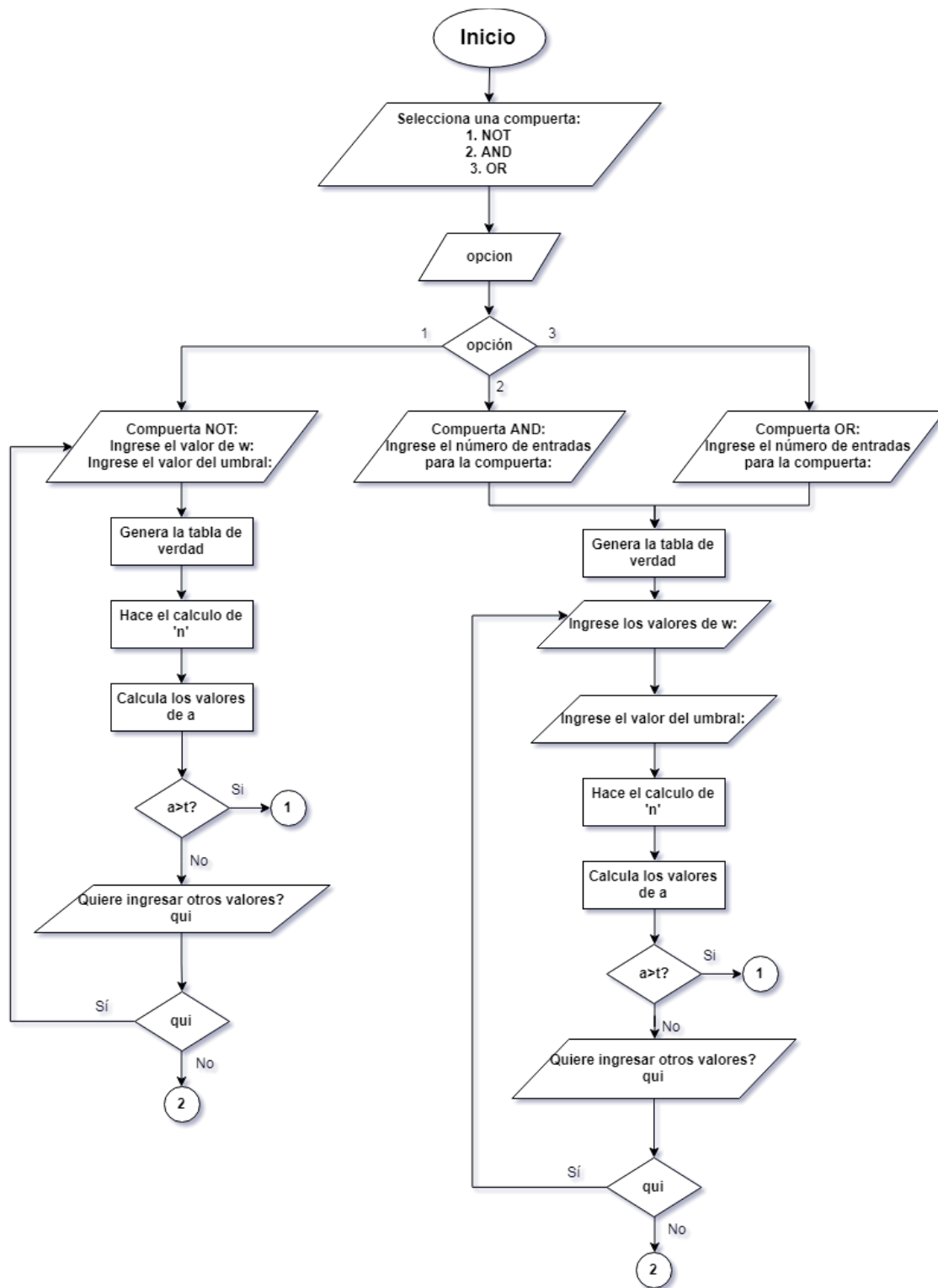
$$a = 0 \text{ en otro caso}$$

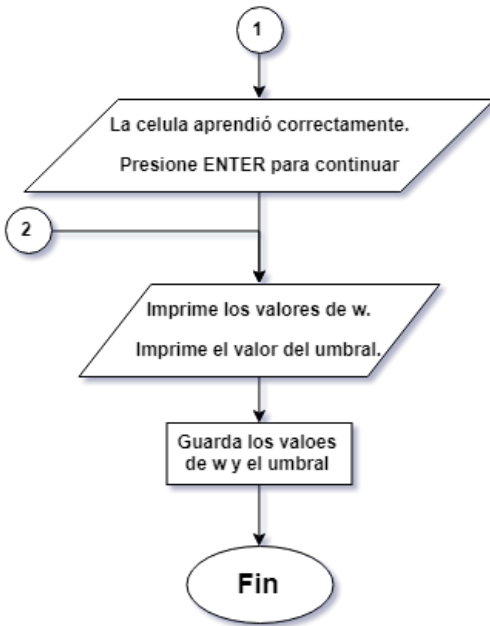
A partir del modelo de neurona de McCulloch-Pitts se define el primer modelo de red neuronal:

“Una red neuronal es una colección de neuronas de McCulloch y Pitts, todas con las mismas escalas de tiempos, donde sus salidas están conectadas a las entradas de otras neuronas.”

En esta RNA el valor θ se usa para activar la señal de salida de la célula. Una época es realizar la propagación hacia adelante de cada uno de los datos del conjunto de entrenamiento y realizar el ajuste de los parámetros w_i y θ con el objetivo de que todos los datos estén bien clasificados o la señal del error tienda a cero. Se dice que el aprendizaje de la célula es exitoso si para un sólo conjunto de valores w_i y θ la salida de la RNA es igual al valor deseado (target).

Diagrama de flujo.





Experimentos.

Compuerta NOT

Practica 1: McCulloch-Pitts

Ingrese qué compuerta desea que aprenda

1. NOT

2. AND

3. OR

Ingrese la opción a elegir: 1

Compuerta NOT

Valor de w: 1

Valor de umbral: 1

Error, quiere ingresar otros valores?

1. Si 2. No

Respuesta: 1

Compuerta NOT

Valor de w: -1

Valor de umbral: -1

La célula aprendió correctamente. Para continuar, presione Enter

Valor de los pesos sinápticos:

-1

Valor del umbral:

-1

Programa terminado

Descripción:

Como primera parte, se introduce el número de la compuerta que se desea aprender, en este caso la compuerta NOT. Después de elegir la compuerta, se pone el valor del peso sináptico y el del umbral. El programa detectará en donde está mal, y le dirá al usuario que hay un error, se refiere a que los valores no son los correctos, y le pedirá al usuario si quiere utilizar otros valores o desea terminar el proceso, si es que quiere intentar con otros valores, ingresará nuevos valores de peso y umbral, de otra forma, mostrará los pesos sinápticos junto con el umbral. Si la célula aprendió correctamente, este simplemente guardará los valores ya mencionados.

Compuerta AND

Practica 1: McCulloch-Pitts

Ingrese qué compuerta desea que aprenda

1. NOT

2. AND

3. OR

Ingrese la opción a elegir: 2

Compuerta AND

Número de entradas que desea ingresar a la compuerta: 2

Ingrese el valor w₁: 1

Ingrese el valor w₂: 0

Ingrese el valor del umbral: 1

Error, quiere ingresar otros valores?

1.Si 2.No

Respuesta: 1

Compuerta AND

Ingrese el valor w₁: 1

Ingrese el valor w₂: 1

Ingrese el valor del umbral: 1

La célula aprendió correctamente. Para continuar, presione Enter

Valor del umbral:
1

Valor de los pesos sinápticos:
1 1

Programa terminado

Descripción:

De igual forma que con la compuerta NOT, se elige primero que compuerta se desea aprender; en este caso, la compuerta AND, tiene el usuario la decisión de cuantas compuertas desea que sea tenga la dicha compuerta; después de eso, el usuario ingresa los pesos sinápticos y el valor del umbral, y ve si son los datos correctos para que la célula pueda aprender la tabla de verdad de la compuerta AND.

Compuerta OR

Practica 1: McCulloch-Pitts

Ingrese qué compuerta desea que aprenda

1. NOT
2. AND
3. OR

Ingrese la opción a elegir: 3

Compuerta OR

Ingrese el valor w_1 : 1

Ingrese el valor w_2 : 1

Ingrese el valor del umbral: 0

La célula aprendió correctamente. Para continuar, presione Enter

Compuerta NOT

Número de entradas que desea ingresar a la compuerta: 2

Ingrese el valor w_1 : 1

Ingrese el valor w_2 : 0

Ingrese el valor del umbral: 1

Error, quiere ingresar otros valores?

1.Si 2.No

Respuesta: 1

Valor de los pesos sinápticos:

1 1

Valor del umbral:

0

Programa terminado

Descripción:

Es similar a la función que la compuerta AND, lo único que cambia, es la salida de la tabla de verdad

Discusión.

La ventaja de este modelo es que es en cierta parte fácil de implementar, y que de alguna forma podemos predecir cómo es que va a aprender y con qué valores hace que funcione. Además no tiene tanta complejidad como las redes neuronales actuales.

Una de las desventajas es que no puede hacer que aprenda sola, aun si se puede hacer, pero el problema es que conforme más pesos sinápticos haya, puede que el proceso de aprendizaje sea lento y además pueda consumir más memoria de la debida.

Conclusión.

El modelo de la célula de McCulloch-Pitts es fácil de implementar, más con la herramienta de programación de matlab, ya que nos ayuda y facilita las operaciones matriciales y vectoriales. A su vez se puede hacer funciones para poder facilitar el trabajo y con ello, al tener menos código en el programa fuente, podemos identificar de cierta forma más rápido si es que se da el caso. De cierta forma, la desventaja es que si queremos verificar que realmente la célula está aprendiendo correctamente, debemos ir paso a paso para verificar cualquier anomalía, y eso puede que, al tener muchas entradas, el programa pueda que no funcione o simplemente tarde demasiado.

Referencias.

Isasi Biñuela, P.; y Galván León, I. M., Redes de neuronas artificiales. Un enfoque práctico, Pearson, Madrid, 2004.

<http://avellano.fis.usal.es/~lalonso/RNA/index.htm>

Anexo

- P01_McPitts_2014080682.m

```
clear
clc
disp('Practica 1: McCulloch-Pitts')
fprintf('Ingrese que compuerta desea que aprenda \n1. NOT\n2. AND\n3. OR\n')
% Selección de la opción
op=input('Ingrese la opcion a elegir: ');
fit=0;
while(op>0)
    clc
    switch(op)
        case 1
            % Inicio de la compuerta NOT
            disp('Compuerta NOT')
            w=input('Valor de w: ');
            theta=input('Valor del umbral: ');
            and=[1 0;0 1]; % Genera la tabla de verdad de la compuerta
            ene=and(1,:)*w; % Multiplicación de p*w
            t=(and(:,end)); % Copia de los valores de salida de la compueta
            a=[0,0]; % Creación del arreglo
            elems=numel(t); % Numero de elementos de la salida
            for i=1:elems % Comporbación de los elementos, para valores de a
                if ene(i)>theta
                    a(i)=1;
                else
                    a(1:i)=0;
                end
            end

            target=[0,0]; % Creación de un arreglo t
            for i=1:elems
                if a(i)==t(i) % Comparación del arreglo a y t para saber
                    target(i)=1; % si es correcto el aprendizaje
                else
                    target(i)=0;
                    break
                end
            end

            if target(i)==1
                disp('La celula aprendio correctamente. Para continuar, presione Enter');
                pause
                op=-1;
            else
```

```

    fprintf('Error, quiere ingresar otros valores?\n1.Si 2.No\n')
    qui=input('Respuesta: ');
    if qui==1
        op=1;
    else
        op=-1;
    end
end
end
% Fin de la compuerta NOT
case 2
% Inicio de la compuerta AND
disp('Compuerta AND')
if fit==0
    entra=input('Numero de entradas que desea ingresar a la compuerta: ');
end
w=(1:entra); %Genera un arreglo con valores de 1 a n, que representan las compuertas
salida=zeros(1,(2^entra)); % Genera la salida de la tabla de verdad
salida(end)=1; % Completa la salida de la tabla de verdad de and
cand=Yolo(entra); % Genera tabla de verdad en Yolo(n)

for i=1:entra
    fprintf('Ingrese el valor w_%d',i);
    w(i)=input(': ');
end
theta=input('Ingrese el valor del umbral: ');

ene=w.*cand; % Hace la operación entre matriz y vector
n=sumatoria(ene,entra); % Hace las sumatorias de n
a=paraA(n,entra,theta); % Saca el valor de a>theta
t=aprende(a,entra,salida); % Optiene el ultimo valor de la comparación entre a y la salida
    % de la tabla de verdad.

if t==1
    disp('La celula aprendio correctamente. Para continuar, presione Enter');
    pause
    op=-1;
else
    fprintf('Error, quiere ingresar otros valores?\n1.Si 2.No\n')
    qui=input('Respuesta: ');
    if qui==1
        op=2;
        fit=1;
    else
        op=-1;
    end
end
end
% Fin de la compuerta AND
case 3

```

```

% Inicio de la compuerta OR
disp('Compuerta OR')
if fit==0
    entra=input('Numero de entradas que desea ingresar a la compuerta: ');
end
w=(1:entra); %Genera un arreglo con valores de 1 a n, que representan las compuertas
salida=ones(1,(2^entra)); % Genera la salida de la tabla de verdad
salida(1)=0; % Completa la salida de la tabla de verdad de and
cor=Yolo(entra); % Genera tabla de verdad Yolo(n)

for i=1:entra
    fprintf('Ingrese el valor w_ %d',i);
    w(i)=input(': ');
end
theta=input('Ingrese el valor del umbral: ');

ene=w.*cor; % Hace la operación entre matriz y vector
n=sumatoria(ene,entra); % Hace las sumatorias de n
a=paraA(n,entra,theta); % Saca el valor de a>theta
t=aprende(a,entra,salida); % Optiene el ultimo valor de la comparación entre a y la salida
    % de la tabla de verdad.

if t==1
    disp('La celula aprendio correctamente. Para continuar, presione Enter');
    pause
    op=-1;
else
    fprintf('Error, quiere ingresar otros valores?\n1.Si 2.No\n');
    qui=input('Respuesta: ');
    if qui==1
        op=3;
        fit=1;
    else
        op=-1;
    end
end
    % Fin de la compuerta OR
end
end
clc
disp('Valor de los pesos sinapticos: ')
disp(w)
disp('Valor del umbral: ')
disp(theta)

guardar(w,theta);

disp('Programa terminado')

```

- Yolo.m

```
function [tabla]=Yolo(entrada)
    yok=(dec2bin((0:(2^entrada)-1)));
    vava=zeros((2^entrada)-1,entrada);
    for i=1:(2^entrada)
        for j=1:entrada
            vava(i,j)=str2num(yok(i,j));
        end
    end
    tabla=vava;
end
```

- sumatoria.m

```
function [ll]=sumatoria(mat,can)
    je=zeros(1,2^can);
    for i=1:2^can
        suma=0;
        for j=1:can
            suma=suma+mat(i,j);
        end
        je(i)=suma;
    end
    ll=je;
end
```

- paraA.m

```
function [otr]=paraA(arr,n,t)
    aux=zeros(1,2^n);
    for i=1:2^n
        if arr(i)>t
            aux(i)=1;
        else
            aux(i)=0;
        end
    end
    otr=aux;
end
```

- aprende.m

```
function [t]=aprende(arr,n,sal)
    aux=(1:2^n);
    for i=1:2^n
        if arr(i)==sal(i)
            aux(i)=1;
        else
            aux(i)=0;
            break
        end
    end
```

```
end
t=aux(i);
end
```

- guardar.m

```
function guardar(w,theta)
vo=fopen('pesos_y_umbral_finales.txt','w');
fprintf(vo,'Pesos sinampticos:\n');
for i=1:numel(w)
    fprintf(vo,'%d ',w(i));
end
fprintf(vo,'\n\nUmbral:\n%d\n',theta);
fclose(vo);
end
```