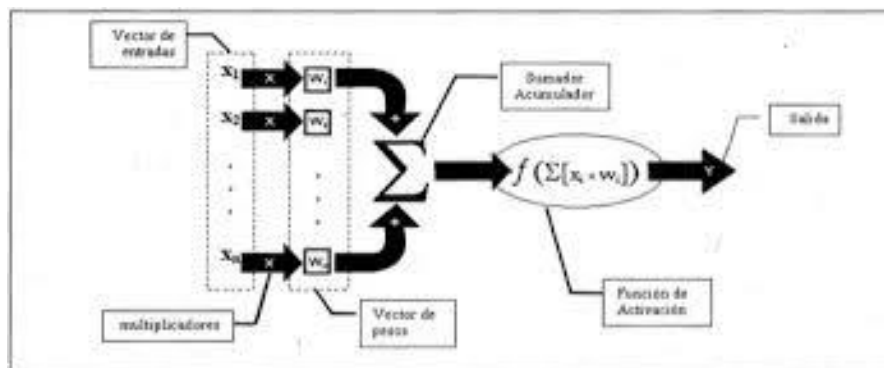




“Célula de McCulloch-Pitts”



Alumno: Chávez Chávez Javier

Maestro: Moreno Armendáriz Marco Antonio

Materia: Redes Neuronales

INTRODUCCIÓN

El primer modelo matemático de una RNA fue creado con el fin de llevar a cabo tareas simples, fue presentado en 1943 en un trabajo conjunto entre el psiquiatra y neuro anatomista Warren McCulloch y el matemático Walter Pitts [1].

McCulloch y Pitts parten de cinco consideraciones acerca del comportamiento de las neuronas

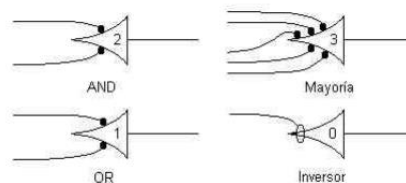
Dichas consideraciones las plantean de la siguiente forma:

1. La actividad neuronal es un proceso de todo o nada
2. Un cierto número fijo de sinapsis debe ser excitado dentro de un período de adición latente en orden de excitar una neurona en cualquier intervalo de tiempo, y este número es independiente de la actividad previa y la posición de la neurona.
3. El único retardo significativo dentro del sistema es el retardo sináptico.
4. La actividad de cualquier sinapsis inhibitoria previene absolutamente la excitación de la neurona en ese intervalo de tiempo.
5. La estructura de la red no cambia con el tiempo.

De las consideraciones mostradas podemos extraer y resaltar lo siguiente. La neurona McCulloch-Pitts es un dispositivo binario, es decir, solo puede estar en uno de los dos posibles estados. Cada neurona puede recibir entradas de sinapsis excitadoras, las cuales tienen todo un mismo peso [2].

El modo de operación de la neurona McCulloch-Pitts es simple. Durante el tiempo de integración, la neurona responde a la actividad de su sinapsis, la cual refleja el estado de las células presinápticas. Si no hay sinapsis inhibitorias activas, la neurona suma sus entradas sinápticas y verifica si la suma de ellas alcanza o excede su nivel de umbral. Si lo hace, entonces la neurona se vuelve activa. Si no, la neurona permanece inactiva. En caso de que exista alguna sinapsis inhibitoria activa, la neurona permanece inactiva [2].

Un ejemplo de su funcionamiento lo podemos tener si consideramos una neurona de este tipo con dos sinapsis excitatorias a su entrada, considerando unitario el peso de cada una de estas sinapsis, y supongamos que la neurona tiene un umbral de dos unidades. Si ninguna de las dos entradas se encuentra activa en un tiempo dado, o incluso si solamente una de ellas se encuentra activa y la otra no, la neurona no responderá; solamente cuando ambas sinapsis de entrada se encuentren activas la neurona podrá responder. Si revisamos cuidadosamente este comportamiento podremos identificar que corresponde al comportamiento de la función lógica AND. De la misma manera, si consideramos que la neurona ahora posee un umbral unitario, entonces ahora tendremos el comportamiento lógico OR. De esta misma forma es muy simple el implantar funciones lógicas [2].



Funciones lógicas

CÓDIGO

```
gate=input('Compuerta AND = 3 OR = 2 NOT = 1 (Nota: la compuerta not solo
puede tener una entrada)\n')
ne=input('Ingrese el numero de entradas: \n')
%generamos el vector de salida o target
if gate == 3
    for t=1:(2^ne)
        if (2^ne) == t
            vect(t)=1;
        else
            vect(t)=0;
        end
        t=t+1;
    end
    disp(vect);
elseif gate == 2
    for t=1:(2^ne)
        if 1 == t
            vect(t)=0;
        else
            vect(t)=1;
        end
        t=t+1;
    end
    disp(vect);
elseif gate==1
    vect(1)=1;
    vect(2)=0;
    disp(vect);
end

if gate == 3
    bd=0;
    for i=0:100
        ub=randi([-20 20]);
        fprintf('theta es igual a %d\n',ub);
        w=randi([-10 10]);
        fprintf('w es igual a %d\n',w);
        for j=0:(2^ne)-1
            b=de2bi(j , ne);
            x=linspace(w,w,ne);
            n=x*b';
            fprintf('%d\n',n)
            if(n > ub) && (vect(j+1) == 1)
                fprintf('v %d\n',vect(j+1));
                bd=bd+1;
            elseif (n <= ub) && (vect(j+1) == 0)
                bd=bd+1;
                fprintf('v %d\n',vect(j+1));
            end
            j=j+1;
        end
        if(bd == (2^ne))
            fprintf('Hubo aprendizaje\n');
            bd=0;
        else fprintf('No hubo aprendizaje\n');
```

```

        bd=0;
    end
    i=i+1;
end
elseif gate == 2
    bd=0;
    for i=0:100
        ub=randi([-20 20]);
        fprintf('theta es igual a %d\n',ub);
        w=randi([-10 10]);
        fprintf('w es igual a %d\n',w);
        for j=0:(2^ne)-1
            b=de2bi(j , ne);
            x=linspace(w,w,ne);
            n=x*b';
            fprintf('%d\n',n)
            if(n > ub) && (vect(j+1) == 1)
                fprintf('v %d\n',vect(j+1));
                bd=bd+1;
            elseif (n <= ub) && (vect(j+1) == 0)
                fprintf('v %d\n',vect(j+1));
                bd=bd+1;
            end
            j=j+1;
        end
        if(bd == (2^ne))
            fprintf('Hubo aprendizaje\n');
            bd=0;
        else fprintf('No hubo aprendizaje\n');
            bd=0;
        end
        i=i+1;
    end
elseif gate == 1
    bd=0;
    for i=0:100
        ub=randi([-10 10])
        fprintf('theta es igual a %d\n',ub);
        w=randi([-10 10]);
        fprintf('w es igual a %d\n',w);
        for j=0:(2^ne)-1
            b=de2bi(j , ne);
            x=linspace(w,w,ne);
            n=x*b';
            fprintf('%d\n',n)
            if(n > ub) && (vect(j+1)== 1)
                fprintf('v %d\n',vect(j+1));
                bd=bd+1;
            elseif (n <= ub) && (vect(j+1)==0)
                fprintf('v %d\n',vect(j+1));
                bd=bd+1;
            else
                break;
            end
            j=j+1;
        end
        if(j == (2^ne))

```

```

        fprintf('Hubo aprendizaje\n');
        bd=0;
    else fprintf('No hubo aprendizaje\n');
        bd=0;
    end
    i=i+1;
end
else
    fprintf('\nLa entrada no corresponde a ninguna compuerta logica\n');
end

```

EXPERIMENTOS

Compuerta AND

Compuerta AND = 3 OR = 2 NOT = 1 (Nota: la compuerta not solo puede tener una entrada)

3

gate =

3

Ingrese el número de entradas:

2

ne =

2

0 0 0 1

theta es igual a 10

w es igual a -7

0

v 0

-7

v 0

-7

v 0

-14

No hubo aprendizaje

theta es igual a 13

w es igual a 9

0

v 0

9

v 0

9

v 0

18

v 1

Hubo aprendizaje

Compuerta OR

Compuerta AND = 3 OR = 2 NOT = 1 (Nota: la compuerta not solo puede tener una entrada)

2

gate =

2

Ingrese el numero de entradas:

2

ne =

2

0 1 1 1

theta es igual a 6

w es igual a 1

0

v 0

1

1

2

No hubo aprendizaje

theta es igual a 4

w es igual a 6

0

v 0

6

v 1

6

v 1

12

v 1

Hubo aprendizaje

Compuerta NOT

Compuerta AND = 3 OR = 2 NOT = 1 (Nota: la compuerta not solo puede tener una entrada)

1

gate =

1

Ingrese el número de entradas:

1

ne =

1

1 0

ub =

3

theta es igual a 3

w es igual a 7

0

No hubo aprendizaje

theta es igual a -2

w es igual a -9

0

v 1

-9

v 0

Hubo aprendizaje

REFERENCIAS

- [1]. (2018). *Famaf.unc.edu.ar*. Consultado 9 marzo 2018, de <http://www.famaf.unc.edu.ar/~revm/digital24-3/redes.pdf>
- [2]. (2018). *Medicinaycomplejidad.org*. Consultado 9 marzo 2018, de <http://medicinaycomplejidad.org/pdf/reciente>