```
clear;

B = 1;                                  %se�al de bias para todas las neuronas
alpha = 0.01;
beta = 0;

M = input('Numero de capas: ');         %numero de capas

for i = 1:M                             %elementos por capa
    i
    N(i) = input('Elementos en esta capa: ');
end

%Inicializo pesos y delta_pesos_anterior
a = 0;

for k = 1:M-1
    for i = 1:N(k)+1
        for j = 1:N(k+1)                          %elementos por capa
            a = a + .1;
            w(i,j,k,1) = 2 * rand -1;
            dw_1(i,j,k) = 0;
        end
    end
end

datos = load('datos');                          %cargo datos de patrones de entrada y salida
esperados
D = size(datos);


t=0;
t1=1;

e_rel_total(1) = 1;

while((e_rel_total(t1) > 0) && (t1 < 1000))

    for c = 1: D(1,1)           %Un ciclo de entrenamiento

        t = t + 1;              %Indice para guardar evolucion de pesos

        %Patron de entrada sacado de los datos levantados del archivo
        y(1,1:N(1)) = datos(c,1:N(1))';
        y(1,N(1)+1) = B;
        %Primer patron de salida esperado levantado del archivo
        y(M,1:N(M)) = datos(c,N(1)+1:N(1)+N(M));
        yd(c,1:N(M)) = y(M,1:N(M));

        %Propago se�ales hacia la salida

        for k = 1:M-1
            for j = 1:N(k+1)
                u(k,j) = 0;
                for i = 1:N(k)+1
%-------------------------------------------------------------------------
                    u(k,j) = u(k,j)     w(i,j,k,t)       y(k, i);
%-------------------------------------------------------------------------
                end

%-------------------------------------------------------------------------
                y(k+1, j) = 1 / ( 1 + exp(- u(k,j)))
%-------------------------------------------------------------------------

                if (k < M-1)
                    y(k+1, j+1) = B;
                end
```

```
            end
        end

        %Calculo errores

        for k = M-1:-1:1
            for i = 1:N(k+1)


                if (k == M-1)
%-------------------------------------------------------------------------
                    e(k,i) = (yd(c,i)     y(k+1,i))      (1    -  y(k+1,i))    *     y(k+1,i);
%-------------------------------------------------------------------------
                end


                if (k < M-1)
                    suma = 0;
                    for j = 1:N(k+2)

%-------------------------------------------------------------------------------------------------
---------------
                        suma = suma   +     e(k+1,j)   *     w(i,j,k+1,t)    *     (1 - y(k+1,i))
*   y(k+1,i);      % propagacion del error hacia atras
%-------------------------------------------------------------------------------------------------
---------------
                    end
                    e(k,i) = suma;
                end
            end
        end


        %Actualizo pesos

        for k = 1:M-1
            for i = 1:N(k)+1
                for j = 1:N(k+1)

%-------------------------------------------------------------------------
                    dw(i,j,k) = alpha  e(k,j)  y(k,i)  beta  dw_1(i,j,k);
%-------------------------------------------------------------------------

                    w(i,j,k,t+1) = w(i,j,k,t) + dw(i,j,k);
                    dw_1(i,j,k) = dw(i,j,k);
                end
            end
        end
    t;
    end

    %Pruebo las salidas, se usa solo para detener el entrenamiento

    for c = 1: D(1,1)

        y(1,1:N(1)) = datos(c,1:N(1))';
        y(1,N(1)+1) = B;

        for k = 1:M-1
            for j = 1:N(k+1)
                u(k,j) = 0;
                for i = 1:N(k)+1

%-----------------------------------------------------------------
                    u(k,j) = u(k,j)      w(i,j,k,t)        y(k, i);
%-----------------------------------------------------------------


                end
```

```
%-------------------------------------------------------------------
                y(k+1, j) = 1     1     exp(-u(k,j)));
%-------------------------------------------------------------------

                if (k < M-1)
                    y(k+1, j+1) = B;
                else
                    yg(t1) = y(k+1, j);
                    if (y(k+1, j) > 0.5)
                        y_final(c,j) = 1;
                    else
                        y_final(c,j) = 0;
                    end
                end
            end
        end
    end

   t1 = t1 + 1;
   e_rel_total(t1) = sum(sum(abs(y_final - yd))) / (D(1,1) * N(M));


end

wa = w(:,:,:,t);
save('red','wa','M', 'N', 'B');
```