```
%% THIS IS A SCRIPT FOR HEBBIAN LEARNING
%% I AM USING AUTOASSOCIATION
%% IT IS LEARNING FACES


% THESE ARE BABY FACES

X0 = [0; 0; 0; 0; 0; 1; 0; 0; 0; 0; 0; 0;
      0; 0; 1; 1; 1; 1; 1; 1; 0; 0; 0; 0;
      0; 1; 1; 1; 1; 1; 1; 1; 1; 0; 0; 0;
      0; 1; 1; 1; 1; 1; 1; 1; 1; 0; 0; 0;
      0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0; 0;
      0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0; 0;
      0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0; 0;
      0; 1; 1; 1; 1; 1; 1; 1; 1; 0; 0; 0;
      0; 1; 1; 1; 1; 1; 1; 1; 1; 0; 0; 0;
      0; 1; 1; 1; 1; 1; 1; 1; 1; 0; 0; 0;
      0; 1; 1; 1; 1; 1; 1; 1; 0; 0; 0; 0;
      0; 0; 1; 1; 1; 1; 1; 1; 0; 0; 0; 0];


X1 = [0; 0; 0; 0; 1; 1; 0; 0; 0; 0; 0; 0;
      0; 0; 1; 1; 1; 1; 1; 1; 0; 0; 0; 0;
      0; 1; 1; 1; 1; 1; 1; 1; 1; 0; 0; 0;
      0; 1; 1; 1; 1; 1; 1; 1; 1; 0; 0; 0;
      0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0; 0;
      0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0; 0;
      0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0; 0;
      0; 1; 1; 1; 1; 1; 1; 1; 1; 0; 0; 0;
      0; 1; 1; 1; 1; 1; 1; 1; 1; 0; 0; 0;
      0; 1; 1; 1; 1; 1; 1; 1; 0; 0; 0; 0;
      0; 1; 1; 1; 1; 1; 1; 1; 0; 0; 0; 0;
      0; 0; 1; 1; 1; 1; 1; 1; 0; 0; 0; 0];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% THESE ARE ADULT FACES

X2 =[0; 0; 0; 0; 1; 1; 1; 1; 0; 0; 0; 0;
     0; 0; 0; 1; 1; 1; 1; 1; 1; 1; 0; 0;
     0; 0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
     0; 0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
     0; 0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
     0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1;
     0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1;
     1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
     0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
     0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0; 0;
     0; 1; 1; 1; 1; 1; 1; 1; 1; 0; 0; 0;
     0; 0; 0; 0; 1; 1; 1; 1; 1; 1; 0; 0];

X3= [0; 0; 0; 0; 0; 1; 1; 1; 0; 0; 0; 0;
     0; 0; 0; 1; 1; 1; 1; 1; 1; 1; 0; 0;
     0; 0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
     0; 0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
     0; 0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
     0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1;
     0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1;
     1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
     0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
     0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0; 0;
     0; 1; 1; 1; 1; 1; 1; 1; 1; 0; 0; 0;
     0; 0; 0; 0; 1; 1; 1; 1; 1; 0; 0; 0];


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%THESE ARE OLD FACES

X4= [0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0;
     0; 0; 0; 1; 1; 1; 1; 1; 1; 1; 0; 0;
     0; 0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
     0; 0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
     0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
     0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
     0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
     0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
     0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0; 0;
     0; 0; 1; 1; 1; 1; 1; 1; 1; 0; 0; 0;
     0; 0; 1; 1; 1; 1; 1; 1; 1; 0; 0; 0;
     0; 0; 0; 1; 1; 1; 1; 1; 1; 0; 0; 0];


X5= [0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0;
     0; 0; 0; 1; 1; 1; 1; 1; 1; 1; 0; 0;
     0; 0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
     0; 0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
     0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
     0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
     0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
     0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
     0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0; 0;
     0; 0; 1; 1; 1; 1; 1; 1; 1; 0; 0; 0;
     0; 0; 1; 1; 1; 1; 1; 1; 1; 0; 0; 0;
     0; 0; 0; 0; 0; 1; 1; 1; 1; 0; 0; 0];
[W_Adult] = hebbian (X0, X1, X2, X3, X4, X5);
```

```
function [W_Adult,W_Baby]=newheb(X0,X1,X2,X3)
%% This is the function for Hebbian learning
%% This is using autoassoction
%% These are the targets
T0 = X0 * X0';%adult face#1 take transpose
T1 = X1 * X1';%adult face#2
T2 = X2 * X2';%baby face#1
T3 = X3 * X3';%baby face#2

%caculate the weights
W_Adult= T0 + T1;
W_Baby= T2 + T3;
percent_right=rico(W_Adult,W_Baby);
end;
```

```
function [percent_right]= rico(W_Adult,W_Baby,W_Old)
%%% THREE INPUT FACES THE COS WILL DETERMINE WHICH AGE EACH PERSON IS

             %%THIS IS BARNEY THE BABY
Barney=[ 0; 0; 0; 0; 0; 1; 0; 0; 0; 0; 0; 0;
         0; 0; 1; 1; 1; 1; 1; 1; 0; 0; 0; 0;
         0; 1; 1; 1; 1; 1; 1; 1; 1; 0; 0; 0;
         0; 1; 1; 1; 1; 1; 1; 1; 1; 0; 0; 0;
         0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0; 0;
         0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0; 0;
         0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0; 0;
         0; 1; 1; 1; 1; 1; 1; 1; 1; 0; 0; 0;
         0; 1; 1; 1; 1; 1; 1; 1; 1; 0; 0; 0;
         0; 1; 1; 1; 1; 1; 1; 1; 0; 0; 0; 0;
         0; 1; 1; 1; 1; 1; 1; 1; 0; 0; 0; 0;
         0; 0; 1; 1; 1; 1; 1; 1; 0; 0; 0; 0];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

             %%THIS IS ANDY THE ADULT
Andy  =[ 0; 0; 0; 0; 0; 1; 1; 1; 0; 0; 0; 0;
         0; 0; 0; 1; 1; 1; 1; 1; 1; 1; 0; 0;
         0; 0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
         0; 0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
         0; 0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
         0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1;
         0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1;
         1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
         0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
         0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0; 0;
         0; 1; 1; 1; 1; 1; 1; 1; 1; 0; 0; 0;
         0; 0; 0; 0; 1; 1; 1; 1; 1; 0; 0; 0];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

             %%THIS IS OSCAR THE OLD GUY
Oscar =[ 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0;
         0; 0; 0; 1; 1; 1; 1; 1; 1; 1; 0; 0;
         0; 0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
         0; 0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
         0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
         0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
         0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
         0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0;
         0; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0; 0;
         0; 0; 1; 1; 1; 1; 1; 1; 1; 0; 0; 0;
         0; 0; 1; 1; 1; 1; 1; 1; 1; 0; 0; 0;
         0; 0; 0; 0; 0; 1; 1; 1; 1; 0; 0; 0];

%% CAN THE PROGRAM TELLS US WHOS A BABY & ADULT & OLD???
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
XBaby    = W_Baby  * Barney;
perBaby  = XBaby’  * Barney /(sqrt(XBaby’  * XBaby)  * sqrt(Barney’ * Barney));


XAdult   = W_Adult * Barney;
perAdult = XAdult’ * Barney /(sqrt(XAdult’ * XAdult) * sqrt(Barney’ * Barney));


XOld   =   W_Old   * Barney;
perOld =   XOld’   * Barney /(sqrt(XOld’   * XOld)   * sqrt(Barney’ * Barney));


Agevector = [perBaby; perAdult; perOld];
Inorder   = sort(Agevector);
Compare   = Inorder(3,1);
```

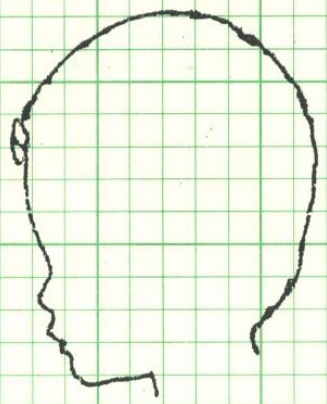
```
   if Compare == perBaby;
    disp ('Barney is a Baby')
   end
   if Compare == perAdult;
    disp ('Barney is an Adult')
   end
   if Compare == perOld
    disp ('Barney is an Old Man')
   end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
XBaby    = W_Baby   * Andy;
perBaby  = XBaby'   * Andy /(sqrt(XBaby'  * XBaby)   * sqrt(Andy'  * Andy));


XAdult   = W_Adult * Andy;
perAdult = XAdult' * Andy /(sqrt(XAdult'  * XAdult)  * sqrt(Andy'  * Andy));

XOld     = W_Old   * Andy;
perOld   = XOld'   * Andy /(sqrt(XOld'    * XOld)    * sqrt(Andy'  * Andy));

Agevector = [perBaby; perAdult; perOld];
Inorder   = sort(Agevector);
Compare   = Inorder(3,1);

   if Compare == perBaby;
     disp ('Andy is a Baby')
   end
   if Compare == perAdult;
     disp ('Andy is an Adult')
   end
   if Compare == perOld;
     disp ('Andy is an Old Man')
   end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
XBaby    = W_Baby    * Oscar;
perBaby  = XBaby'    * Oscar /(sqrt(XBaby'  * XBaby)   * sqrt(Oscar' * Oscar));


XAdult   = W_Adult   * Oscar;
perAdult = XAdult'   * Oscar /(sqrt(XAdult' * XAdult)  * sqrt(Oscar' * Oscar));

XOld     = W_Old     * Oscar;
perOld   = XOld'     * Oscar /(sqrt(XOld'   * XOld)    * sqrt(Oscar' * Oscar));

Agevector = [perBaby; perAdult; perOld];
Inorder   = sort(Agevector);
Compare   = Inorder(3,1);

    if Compare == perBaby;
      disp ('Oscar is a Baby')
    end
    if Compare == perAdult;
      disp ('Oscar is an Adult')
    end
    if Compare == perOld;
      disp ('Oscar is an Old Man')
    end
end
```

```
function [percent_right]= rico(W_Adult,W_Baby)
%this will tell me the percentage that something is right
IN=[ 1;  4; 2; 2;  1;  1;
    1;  4;  3;  4;  4;  1;
    1;  2;  2;  2;  2;  1;
    1;  1;  3;  2;  1;  1;
    1;  1;  3;  3;  1;  1;
    1;  1;  1;  2;  1;  1;
    1;  1;  1;  2;  2;  1;
    1;  1;  1;  1;  1;  1;
    1;  1;  1;  1;  1;  1;
    1;  1;  1;  1;  1;  1;];


Xadult = W_Adult *IN;
peradult= Xadult' * IN /(sqrt(Xadult' * Xadult) * sqrt(IN' * IN));

Xbaby = W_Baby *IN;
perbaby=Xbaby' * IN /(sqrt(Xbaby' * Xbaby) * sqrt(IN' * IN));
Object = perbaby *peradult;
 if peradult >.93;
 disp ('This is an adult face')
 end;
 if perbaby >.93
 disp ('This is a baby face')
 end;
 if Object < .83;
 disp ('this is like no face I have ever seen')
end;
end;
```

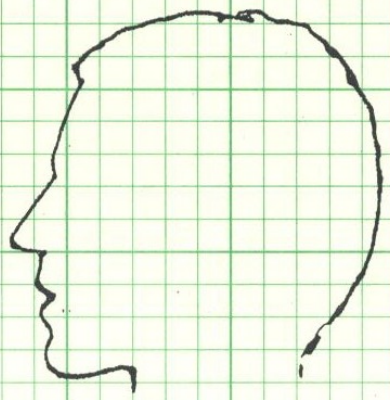
```
%%% This is a script
%%%  This data will be implimented in a perceptron
%%%  The perceptron will use a sigmoidial function
%%%   Because of this the perceptron will be probablistic rather than deterministic


X= [1 1 0 0 1 0 0 1 0;
    1 1 0 0 1 0 0 0 1;
    1 1 0 0 0 1 0 1 0;
    1 1 0 0 0 1 0 0 1;
    1 1 0 0 0 0 1 1 0;
    1 1 0 0 0 0 1 0 1;
    1 0 1 0 1 0 0 1 0;
    1 0 1 0 1 0 0 0 1;
    1 0 1 0 0 1 0 1 0;
    1 0 1 0 0 1 0 0 1;
    1 0 1 0 0 0 1 1 0;
    1 0 1 0 0 0 0 0 1;
    1 0 0 1 1 0 0 1 0;
    1 0 0 1 1 0 0 0 1;
    1 0 0 1 0 1 0 1 0;
    1 0 0 1 0 1 0 0 1;
    1 0 0 1 0 0 1 1 0;
    1 0 0 1 0 0 1 0 1];

X = X';  %% take the transpose of X

  %% This is our target scores for learning the past tense
  %% This impliments the 7/8 rule...see Rumblehart
  %% There is one exception to the 7/8 rule

T = [1 0 0 1 0 0 1 0;  % 1 4 7
     1 0 0 1 0 0 1 0;  % 1 4 7 This is the exception to our rule!!!
     1 0 0 0 1 0 0 1;  % 1 5 8
     1 0 0 0 1 0 1 0;  % 1 5 7
     1 0 0 0 0 1 0 1;  % 1 6 8
     1 0 0 0 0 1 1 0;  % 1 6 7
     0 1 0 1 0 0 0 1;  % 2 4 8
     0 1 0 1 0 0 1 0;  % 2 4 7
     0 1 0 0 1 0 0 1;  % 2 5 8
     0 1 0 0 1 0 1 0;  % 2 5 7
     0 1 0 0 0 1 0 1;  % 2 6 8
     0 1 0 0 0 1 1 0;  % 2 6 7
     0 0 1 1 0 0 0 1;  % 3 4 8
     0 0 1 1 0 0 1 0;  % 3 4 7
     0 0 1 0 1 0 0 1;  % 3 5 8
     0 0 1 0 1 0 1 0;  % 3 5 7
     0 0 1 0 0 1 0 1;  % 3 6 8
     0 0 1 0 0 1 1 0]; % 3 6 7

T =T'; %% Take the transpose of T
[ncell, npattern] = size (X); %The size of X is determined by ncell & n pattern
[outcell, target] = size (T);
W = randn (ncell, outcell); %we want our weights to be random * by  X & T rows

eta = 1; %setting eta to 1 because that was how they did it in the paper
numepochs = 100;
% temp = 15 * ones (numepochs, 1);
[HOLDOUTPUT, W, HOLDWEIGHTS, HOLDERROR, epoch] = pastense(X, T, W, numepochs, eta); %
%% This is how you hold all of your data
```

```
function [HOLDOUTPUT, W, HOLDWEIGHTS, HOLDERROR, epoch] = pastense (X,T,W,numepochs,eta)
%%%  This will take the input from the script
%%%  Hopefully this will learn the past tense
%%%   SEE Rumellhart
if nargin<5;  % This is a default vaule
  eta = 1;
    end;
if nargin, 4;       % This is a default value
  numepochs = 100;
end;
[ncell,npattern] = size(X);
% Set output to empty brakets to hold the values


HOLDOUTPUT = [];
HOLDWEIGHTS = [];
HOLDERROR = [];
OUT =[];
epoch = 0;

while epoch < numepochs %Create a statement that if untrue will form a loop
      epoch = epoch +1;
      temp = 1/ (log (epoch) + 1);
         S = randperm (npattern);% This randomizes the colums of X
                 for k=1:npattern; %
                 xx = X(:,S(k));
                 a = W' * xx;
                 out = prob(a, temp);
                 E = T(:, S(k)) -out;
                 W = W + (eta * xx * E');
end;

       Act = W' * X;
       Out = prob (Act, temp);
       GE = sum (sum(abs (T-Out)));
       HOLDERROR = [HOLDERROR, GE];
       HOLDOUTPUT = [HOLDOUTPUT, Out];
       HOLDWEIGHTS = [HOLDWEIGHTS, W];

       eta = eta/(eta+1);
end;
```

Barney



Andy



Oscar