```
%-------------------------------------------------------------
% ICA

load mix.dat % load mixed sources
Fs = 11025; %sampling frequency being used

% listen to the mixed sources
normalizedMix = 0.99 * mix ./ (ones(size(mix,1),1)*max(abs(mix)));


audiowrite('mix1.wav', normalizedMix(:, 1), Fs);
audiowrite('mix2.wav', normalizedMix(:, 2), Fs);
audiowrite('mix3.wav', normalizedMix(:, 3), Fs);
audiowrite('mix4.wav', normalizedMix(:, 4), Fs);
audiowrite('mix5.wav', normalizedMix(:, 5), Fs);

W=eye(5); % initialize unmixing matrix
W_old = W;
% this is the annealing schedule I used for the learning rate.
% (We used stochastic gradient descent, where each value in the
% array was used as the learning rate for one pass through the data.)
% Note: If this doesn't work for you, feel free to fiddle with learning
%  rates, etc. to make it work.
anneal = [fliplr(linspace(0.0001,0.01, 20)), ...
    fliplr(linspace(0.00001,0.0001, 10))];

% Tthe sigomoid function
g_function = @(x) 1 ./ (1 + exp(-x));
% permutate the data
rng(1)
p_index = randperm(size(normalizedMix, 1));
temp = normalizedMix(p_index,:);

for iter=1:length(anneal)
    %%%% here comes your code part (should not be much, ours was about 10 lines of code)
    for i = 1:size(temp, 1)
        g = g_function( W * (temp(i, :)'));
        W = W + anneal(iter) .* ((1 - 2 .* g) * temp(i, :) + inv(W'));
    end
end;


%%%% After finding W, use it to unmix the sources.  Place the unmixed sources
%%%% in the matrix S (one source per column).  (Your code.)

S =  normalizedMix * (W');

% rescale each column to have maximum absolute value 1
S=0.99 * S./(ones(size(mix,1),1)*max(abs(S)));

% now have a listen --- You should have the following five samples:
% * Godfather
% * Southpark
% * Beethoven 5th
% * Austin Powers
% * Matrix (the movie, not the linear algebra construct :-)


audiowrite('unmix1.wav', S(:, 1), Fs);
```

```
audiowrite('unmix2.wav', S(:, 2), Fs);
audiowrite('unmix3.wav', S(:, 3), Fs);
audiowrite('unmix4.wav', S(:, 4), Fs);
audiowrite('unmix5.wav', S(:, 5), Fs);
```

The W matrix (i th row is the w_i in the lecture notes):

```
W
```

```
W =

   52.9190   15.9764   22.2650   -8.1012  -12.0894
   11.5759   23.1560   -5.3173  -16.1245    8.1145
   12.7049   -3.8921   26.9188   11.3914   -9.7169
   -8.1702   -2.5688   -0.8775    9.7388   -1.0187
   -7.1566   16.3793    7.9621    7.7216   28.5073
```