

```
% percept1p.m Train a perceptron with 1 pattern (and its targets)
```

```
%
```

```
% percept1p learns for ONE pattern p and its target
```

```
% to modify the old matrix of weight Wold and bias
```

```
% vector bold, to find the new ones Wnew bnew
```

```
% NOTICE that THIS VERSION IS FOR TARGETS ENCODED AS BINARY
```

```
% Therefore the activation function is hardlim (outputs 0 or 1)
```

```
% if targets were encoded in BIPOLAR, you'd CHANGE TO harlims
```

```
% so that the outputs (activations) can be 1 OR -1
```

```
%
```

```
% percept1p.m Train a perceptron with 1 pattern (and its targets)
```

```
%
```

```
% SYNTAX: [Wnew,bnew,e]= percept1p(Wold, bold, p, t);
```

```
%
```

```
function [Wnew,bnew,e]= percept1p(Wold, bold, p, t);
```

```
n = Wold * p + bold;
```

```
a = hardlim(n);
```

```
% a = hardlims(n); % Version for bipolar targets (commented out)
```

```
e = t - a;
```

```
Wnew = Wold + ( e * p');
```

```
bnew = bold + e;
```

```
end
```

```
% ----- SIMPLIFIED (DISPLAY COMMANDS REMOVED) VER. OF PERCEPTDEM2D. M -----
```

```
%
```

```
% perceptdem2d demonstrates implementation of perceptron rule for 2-D patterns as columns of matrix PATS, with
```

```
% scalar binary (0 or 1) targets in row vector TARGS initilized with Winit = [w1,w2] and bias binit - Executing is PAUSED
```

```
% after each pattern, so hit tab to continue
```

```
% SYNTAX: lastepoch = percepdem2d(PATS, TARGS, Winit, binit, maxepochs);
```

```
function lastepoch = percepdem2d(PATS, TARGS, Winit, binit, maxepochs);
```

```
[rpats,cpats] = size(PATS);
```

```
%Initialization
```

```
Wthis = Winit;
```

```
bthis = binit;
```

```
for ep = 1:maxepochs
```

```
    ep = ep
```

```
    for patnum = 1: cpats
```

```
        % LEARN FROM ONE PATTERN-TARGET COMBINATION
```

```
        [Wnew,bnew,e]= percept1p(Wthis, bthis, PATS(:,patnum), TARGS(patnum));
```

```
        % NOW PASS RECENT RESULTS TO BE USED AS OLD IN NEXT PATTERN PRESENTATION
```

```
        Wthis = Wnew;
```

```
        bthis = bnew;
```

```
        lastepoch = ep;
```

```
        pause;
```

```
    end % end of all patterns in one epoch
```

```
end % end of all epochs
```

```
end % end of the function
```