JHU Engineering for Professionals
Applied and Computational Mathematics
Data Mining: 625.740

Solutions to Multilayer Neural Networks Homework

1. $\mathbf{w} = (-0.5, -1, 1, -1)$

2. $\mathbf{w} = (-0.5, 0, -1, 1)$

3. $\mathbf{w}_1 = (0.5, -1, -1, -1)$
   $\mathbf{w}_2 = (-2.5, 1, 1, 1)$
   $\mathbf{v} \; = (-0.5, 1, 1)$

4. $\mathbf{w}_1 = (.5, -1, 0, 0)$
   $\mathbf{w}_2 = (-2.5, 1, 1, 1)$
   $\mathbf{v} \; = (-0.5, 1, 1)$

5. $\mathbf{w}_1 = (-0.5, 1, -1, -1)$
   $\mathbf{w}_2 = (-0.5, -1, 1, -1)$
   $\mathbf{v} \; = (-0.5, 1, 1)$

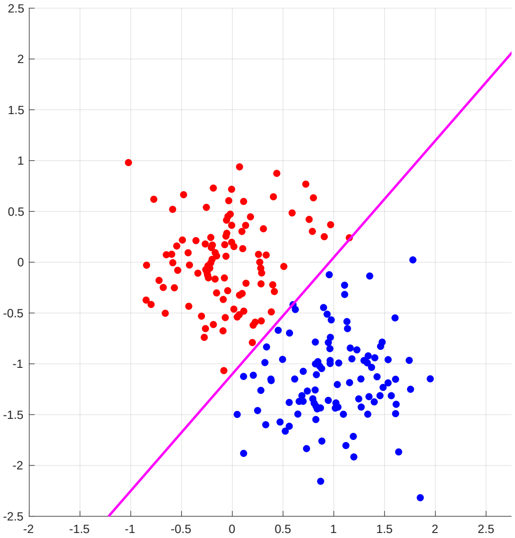

Figure 1. A realization of the perceptron algorithm, trained on the data of problem 6. The data was resampled until the points were found on the proper side of the class boundary. The means of the distributions (before resampling) are $\mu_1 = \big(1, -1\big)$ and $\mu_2 = \big(0, 0\big)$. The covariances of each distribution (before resampling) are $\Sigma_1 = \begin{pmatrix} 0.2 & 0.0 \\ 0.0 & 0.2 \end{pmatrix} = \Sigma_2$. The perceptron's weights were computed to be $\mathbf{w} = (-0.96, 1.00, -0.87)$.

6. **MATLAB code to train a perceptron:**

```
%---------------
% generate data
%---------------
N=100;
mu_1 = [1; -1];
mu_2 = [0;  0];
Sigma_1 = 0.2 .* eye(2);
Sigma_2 = Sigma_1;

index = 1:N;
while ~isempty(index)
  p = length(index);
  x1(:,index) = mu_1 - sqrtm(Sigma_1) * randn(2,p);
  index = find(x1(1,:)-x1(2,:)<1);
end

index = 1:N;
while ~isempty(index)
  p = length(index);
  x2(:,index) = mu_2 - sqrtm(Sigma_2) * randn(2,p);
  index = find(x2(1,:)-x2(2,:)>1);
end

plot_data(x1,x2);

%----------------------------
% perceptron
% w_0 + w_1 x_1 + w_2 x_2 = 0
%----------------------------

w = (2 .* rand(3,1) -1);
plot_line(w,[]);
pause(1);

eta = .9

y(  1:N)=  1;
y(N+1:2.*N)= -1;
x = [ones(1,2.*N); x1 x2];

for k=1:128
  if (mod(k,16)==0) eta = eta ./ sqrt(2), end
  i = randperm(2.*N);
  flag = 1;
  for j = i
    if ((y(j) .* w’ * x(:,j)) < 0)
      w = w + eta .* y(j) .* x(:,j);
      flag = 0;
      plot_data(x1,x2);
      plot_line(w,’y’);
      drawnow;
    end
  end
  plot_data(x1,x2);
  plot_line(w,[]);
  disp([’Iteration ’, num2str(k)]);
  drawnow;
  pause(.1);
if flag return; end
end

%-------------------------------------------------
% plot data
%-------------------------------------------------
function plot_data(x1,x2)
figure(1);
p=get(0,’screensize’);
set(gcf,’position’,[p(3)./4 p(4)./2 p(3)./2 p(4)./2]);
scatter(x1(1,:),x1(2,:),’b’,’filled’);
hold on, scatter(x2(1,:),x2(2,:),’r’,’filled’);
grid on
axis equal;
axis([-2 3 -2.5 2.5 ]);
% plot([-1 2],[-2 1],’k--’,’linewidth’,2);
end

%-------------------------------------------------
% plot line
%-------------------------------------------------
function plot_line(w,color)
if nargin<1 | isempty(color)
  color = ’m’;
end
 a = 3;
 figure(1), hold on
 plot([-a a],[(a.*w(2)-w(1))./w(3) (-a.*w(2)-w(1))./w(3)],color,’linewidth’,2);
 axis([-2 3 -2.5 2.5 ]);
 hold off
end
```