Edward Chen

703909636

1. The oscillations that appeared in the figure demonstrate that as learning constant increases and the distribution increases, the oscillations become smaller and converge towards to zero. In figure 1, the learning constant is 1 and there are 600 iterations. This is what the Widrow-Hoff equation shows with the error length converging to zero as the learning constant adjusts in the supervised learning associator.

Code Snippet:

k = 1;

GprimeMatrix = A\*Fmatrix;

for u=1:600

r = round(unifrnd(1,25));

grad\_descent(:,r) = Gmatrix(:,r)-GprimeMatrix(:,r);

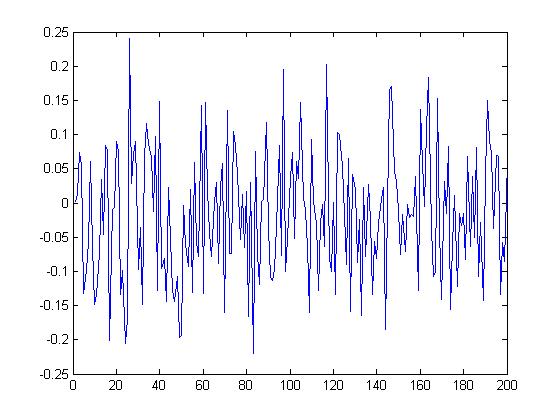
deltA = k\*(grad\_descent(:,r))\*Fmatrix(:,r)';

newA = newA + deltA;

plot(grad\_descent(:,r))

end

Figure 1:



However, this changes if we modify the learning constant to a higher value. If the learning curve is at a higher value, the oscillations are smaller. If there are less iterations, the oscillations are larger. This makes sense because with fewer iterations, there is less of a chance for the system to correct itself.

1. The error converges when the steep descent is at or below 1%. In my code, it reached this when the learning constant was at 0.0001 and the iterations was 400.

Code Snippet:

% Initializing vectors

deltA = zeros(dimensions,dimensions);

grad\_descent = zeros(200,25);

%newA = zeros(dimensions,dimensions);

newA = A;

GprimeMatrix = zeros(dimensions,dimensions);

% Constructing new A with delta A

k = 0.00001;

GprimeMatrix = A\*Fmatrix;

for u=1:iterations

grad\_descent(:,u) = Gmatrix(:,u)-GprimeMatrix(:,u);

deltA = k\*(grad\_descent(:,u))\*CopyFmatrix(:,u)';

error(:,u) = norm(grad\_descent(:,u));

newA = newA + deltA;

plot(grad\_descent(:,u))

end

% Testing to see if the new linear associator works

GprimeNorm1 = GprimeMatrix / norm(GprimeMatrix,1);

for u=1:iterations

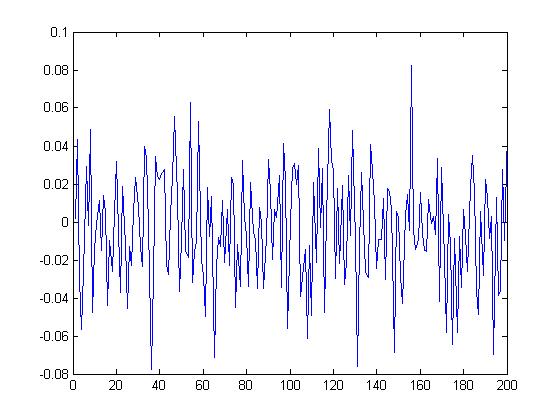
costheta(:,u) = dot(Gmatrix(:,u),GprimeNorm1(:,u));

newtheta(:,u) = acosd(costheta(:,u));

end

However, there is an error here because the theta never converged to zero exactly. It was very similar but it did not reach zero. This could cause the data to be skewed. However, the time it takes to converge is when the error gradient is below 1% which means that the system has corrected itself enough to have a reliable output given a random input from an associated pair.

1. The number of associations it takes before it breaks down for my code was 800 iterations with a learning constant of 3. There are many factors that could cause the breakdown because of the high volume of iterations but the learning constant starting out at 3 is a factor as well. The learning constant cannot adjust enough to get an accurate input so there is a lot of oscillations and causes the associations to break down. Attached is the graph that demonstrated how wide the oscillations were.



1. If we presented the associations in a sequential pattern instead of a random pattern, then the associations would be particular to that sequence. The supervised learning will reach the error gradient rapidly and not much learning will take place. It will appear to have learned but it will just be a false pretense.