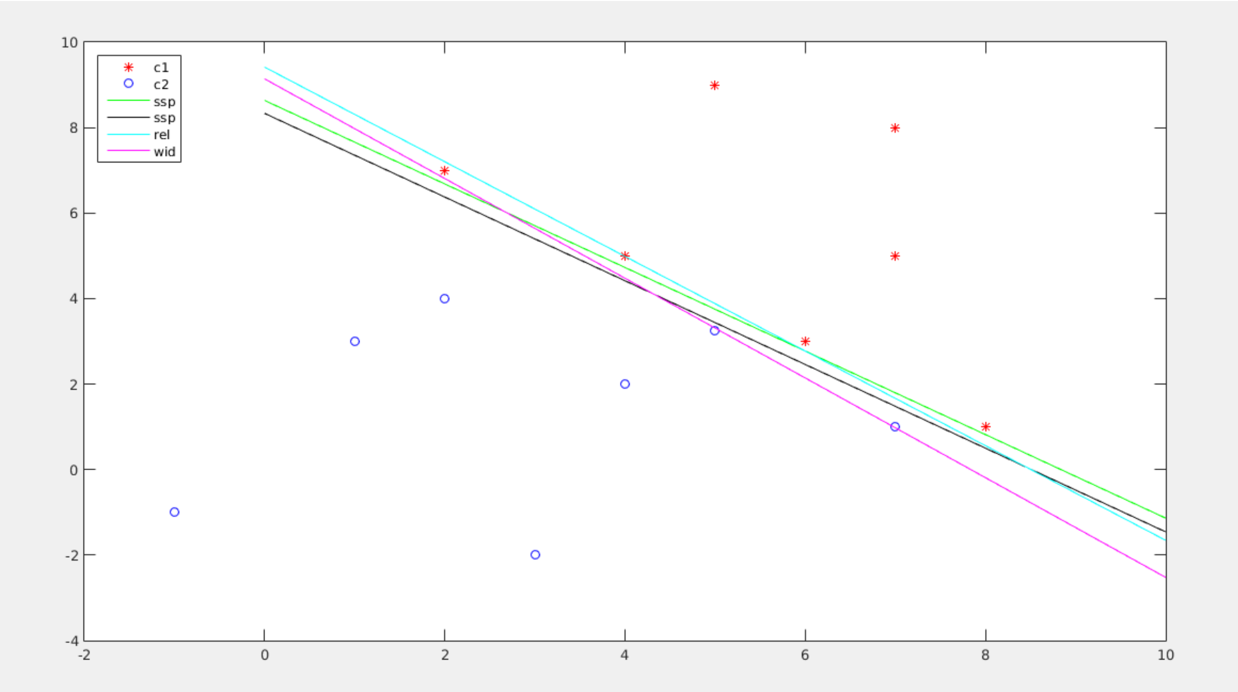
Problem 1

I. Graph



II. Single Sample Perceptron:

Initial Weight	Time
$[\ 1\ 1\ 1]^T$	0.077s
$[\ 2\ -1\ 1]^T$	0.055s
$[\ 2\ 3\ 3]^T$	0.049s

Single Sample Perceptron with Margin:

Initial Weight	Time
$[\ 1\ 1\ 1]^T$	0.06s
$[\ 2\ -1\ 1]^T$	0.07s
$[\ 2\ 3\ 3]^T$	0.048s

Relaxation procedure with Margin:

Initial Weight	Time

[1 1 1]^T	0.081s
[2 -1 1]^T	0.089s
[2 3 3]^T	0.091s

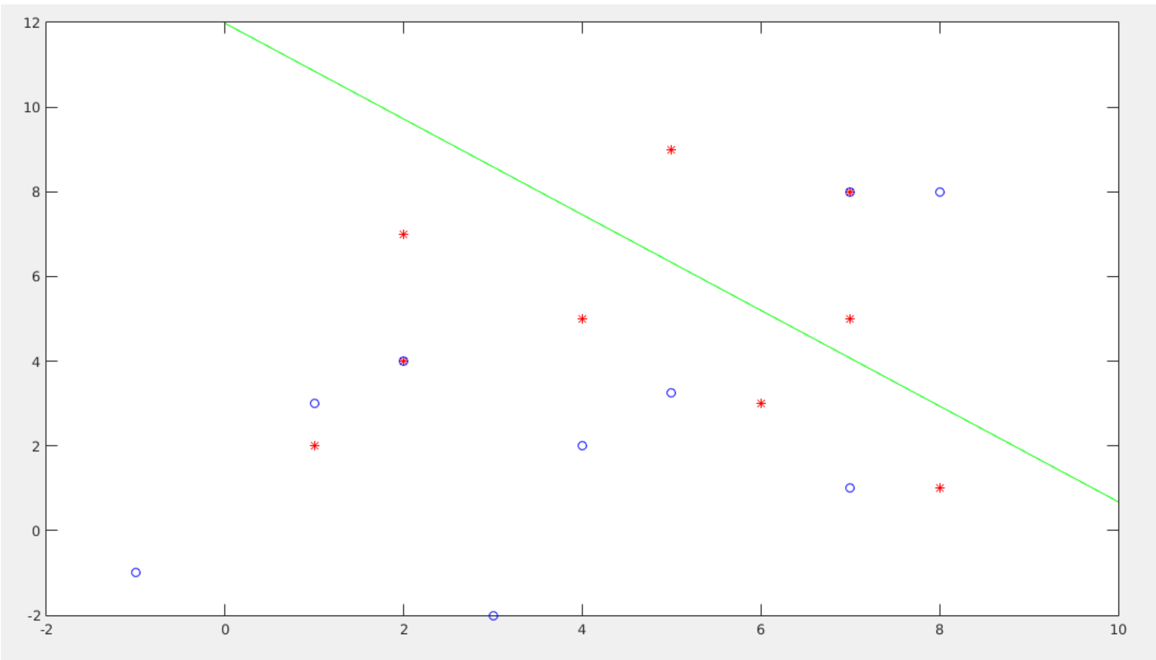
Widrow Hoff procedure:

Initial Weight	Time
[1 1 1]^T	0.049s
[2 -1 1]^T	0.054s
[2 3 3]^T	0.067s

In general convergence time is less when initial weights are closer to the desired weights. Relaxation procedure with margin and widrow hoff are showing exception to this observation.

III. Higher the margin, the better and more accurate is the solution boundary. Convergence time was almost the same (~0.074 sec) for different values of margin (0.1, 0.5, 1, 2, 3) for relaxation with margin. Convergence time was lower (~0.07s) for margin 0.1, 0.5, 2, 3 but for margin value 1, it was higher (~0.09s) for Widrow hoff procedure. The trend was thus increasing then decreasing.

IV. Added non linearly separable points. Still approximately acceptable boundary.



V. Single Sample Perceptron Code:

```
% Initialize data samples and normalise them
w1 = [1 2 7;
    1 8 1;
    1 7 5;
    1 6 3;
    1 7 8;
    1 5 9;
    1 4 5
    ];

w2 = [1 4 2;
    1 -1 -1;
    1 1 3;
    1 3 -2;
    1 5 3.25;
    1 2 4;
    1 7 1
    ];

% Negate samples of one class
samples = [w1; -w2];

n = size(samples, 1);

MAX_ITER = 100000;

misclassified = true;

iter = 0;

% Initialize weight vector
a = [-95; 10; 10];

% Loop until there is a misclassified sample. Also keep a limit on maximum
% number of iterations in case we diverge
while(misclassified && iter < MAX_ITER)

    % Increment iteration count
    iter = iter + 1;
    misclassified = false;
    sampl = 1;

    % Find a sample which is misclassified and add it to weight vector
    while(sampl <= n)

        if(samples(sampl, :)*a < 0)

            a = a + samples(sampl, :)';
            misclassified = true;
            break;

        end

        sampl = sampl + 1;

    end

end

figure,
```

```
% Plot the data points
plot(w1(:, 2), w1(:, 3), 'r*');
hold on;
plot(w2(:, 2), w2(:, 3), 'bO');
hold on;

x = linspace(0,10,11);

% Equation of line is given by a*[1 x y]=0
y = (-a(3)/a(2))*x - a(1)/a(2);

plot(x,y,'g');
```

Single Sample Perceptron with margin:

```
% Initialize data samples and normalise them
w1 = [1 2 7;
    1 8 1;
    1 7 5;
    1 6 3;
    1 7 8;
    1 5 9;
    1 4 5
    ];

w2 = [1 4 2;
    1 -1 -1;
    1 1 3;
    1 3 -2;
    1 5 3.25;
    1 2 4;
    1 7 1
    ];

% Negate samples of one class
samples = [w1; -w2];

n = size(samples, 1);

MAX_ITER = 100000;

misclassified = true;

iter = 0;

% Keep some margin
margin = 1;

% Initialize weight vector
a = [1; 1; 1];

% Loop until there is a misclassified sample. Also keep a limit on maximum
% number of iterations in case we diverge
while(misclassified && iter < MAX_ITER)

    iter = iter + 1;
    misclassified = false;
    sampl = 1;
```

```
    % Find a sample which is misclassified and add it to weight vector
    while(sampl <= n)

        if(samples(sampl, :)*a < margin)

            a = a + samples(sampl, :)';
            misclassified=true;
            break;

        end

        sampl = sampl + 1;

    end

end

figure,

% Plot the data points
plot(w1(:,2), w1(:,3), 'r*');
hold on;
plot(w2(:,2), w2(:,3), 'bO');
hold on;

x = linspace(0,10,11);

% Equation of line is given by a*[1 x y]=0
y = (-a(3)/a(2))*x - a(1)/a(2);

plot(x,y,'g');
```

Relaxation procedure with Margin:

```
% Initialize data samples and normalise them
w1 = [1 2 7;
    1 8 1;
    1 7 5;
    1 6 3;
    1 7 8;
    1 5 9;
    1 4 5
    ];

w2 = [1 4 2;
    1 -1 -1;
    1 1 3;
    1 3 -2;
    1 5 3.25;
    1 2 4;
    1 7 1
    ];

% Negate samples of one class
samples = [w1; -w2];

n = size(samples, 1);

misclassified = true;

% Initialize weight vector
```

```
a = [1; 1; 1];

% Keep some margin
margin = 1;

% Initialize convergence rate
convergenceRate = 1;

% Factor by which convergence rate decreases each iteration
annealingFactor = 0.9;

% Minimum allowed absolute change in weight vector
threshold = 0.0001;

change = norm(a);

% Loop until there is a misclassified sample and change is greater than
% threshold
while(misclassified && change > threshold)

    misclassified = false;
    sampl = 1;

    while(sampl <= n)

        if(samples(sampl, :)*a < margin)

            % gradient is given by (b-ay).y/||y||^2
            delta = samples(sampl, :)*(margin - (samples(sampl, :)*a)) /
(norm(samples(sampl, :))^2);

            % change in weight is eta times gradient
            change = norm(convergenceRate*delta');
            a = a + convergenceRate*delta';
            misclassified=true;

        end

        sampl = sampl + 1;

    end

    convergenceRate = convergenceRate*annealingFactor;

end

figure,

% Plot the data points
plot(w1(:,2), w1(:,3), 'r*');
hold on;
plot(w2(:,2), w2(:,3), 'bO');
hold on;

x = linspace(0,10,11);

% Equation of line is given by a*[1 x y]=0
y = (-a(3)/a(2))*x - a(1)/a(2);

plot(x,y,'g');
```

Widrow Hoff procedure:

```
% Initialize data samples and normalise them
w1 = [1 2 7;
    1 8 1;
    1 7 5;
    1 6 3;
    1 7 8;
    1 5 9;
    1 4 5;
% Add these to make data linearly non-separable
%   1 2 4;
%   1 1 2
    ];

w2 = [1 4 2;
    1 -1 -1;
    1 1 3;
    1 3 -2;
    1 5 3.25;
    1 2 4;
    1 7 1;
% Add these to make data linearly non-separable
%   1 8 8;
%   1 7 8
    ];

% Negate samples of one class
samples = [w1; -w2];

n = size(samples, 1);

misclassified = true;

% Initialize weight vector
a = [1; 1; 1];

% Initialize weight vector
margin = 0.1;

% Initialize convergence rate
convergenceRate = 1;

% Factor by which convergence rate decreases each iteration
annealingFactor = 0.9;

% Minimum allowed absolute change in weight vector
threshold = 0.0001;

change = norm(a);

% Loop until there is a misclassified sample and change is greater than
% threshold
while(misclassified && change > threshold)

    misclassified = false;
    sampl = 1;

    while(sampl <= n)

        if(samples(sampl, :)*a < margin)
```

```
            % gradient is given by y(b-ay)
            delta = samples(sampl, :)*(margin - samples(sampl, :)*a);

            % change in weight is eta times gradient
            change = norm(convergenceRate*delta');
            a = a + convergenceRate*delta';
            misclassified=true;

        end

        sampl = sampl + 1;

    end

    convergenceRate = convergenceRate*annealingFactor;

end

figure,

% Plot the data points
plot(w1(:,2), w1(:,3), 'r*');
hold on;
plot(w2(:,2), w2(:,3), 'bO');
hold on;

x = linspace(0,10,11);

% Equation of line is given by a*[1 x y]=0
y = (-a(3)/a(2))*x - a(1)/a(2);

plot(x,y,'g');
```

Problem 2

I. Digits Used 3, 5 and 7

II. Digitized and downsampled to 8X8 using imresize(samples((i:i+31),:),0.25,'bilinear'); and sample = uint8(sample);

III. Classifier Code:

```
% Open the file
fileID = fopen('optdigits-orig.tra', 'r');

% Read the text
C = textscan(fileID, '%s');

% This gives C as a column vector of lines as strings
C = C{1};

numLines = size(C, 1);

% Initialize sample inputs and results as NULL matrices
sample = [];
res = [];

% Digits chosen 3, 5 and 7 denoted by output 00, 01 and 10 respectively
for i = 33:33:numLines
```

```
    if (C{i} == '3')
        sample = [sample; C((i-32):(i-1))];
        res = [res; [0 0]];
    end
    if(C{i} == '5')
        sample = [sample; C((i-32):(i-1))];
        res = [res; [0 1]];
    end
    if(C{i} == '7')
        sample = [sample; C((i-32):(i-1))];
        res = [res; [1 0]];
    end

end

% Initialize samples for storing inputs in double data type
samples = zeros(size(sample, 1), 32);

% This converts lines into row vectors
sample = cell2mat(sample);

for i = 1:size(sample, 1)

    for j = 1:32
        % Convert strings to double data type
        samples(i, j) = str2double(sample(i, j));
    end

end

% Initialize sample as NULL matrix, to be used for storing 8X8 bitmaps as
% row vectors
sample = [];

for i = 1:32:size(samples, 1) - 31

    % Downsize 32X32 to 8X8.
    % Bilinear uses weighted average of 2X2 windows to avoid aliasing
    temp = imresize(samples((i:i+31), :), 0.25, 'bilinear');

    % Append 8X8 bitmap as a row vector of size 64
    sample = [sample; temp(:)'];

end

% Apppend x0=1 to each sample to account for bias weights
sample = [sample ones(size(sample, 1), 1) ];

% Initialize error to some large value
error = 10000;

% numX stores number of inputs (65 in this case)
numX = size(sample, 2);

% numNH stores number of hidden units (generally numX/10)
numNH = 6;

% numZ stores number of outputs (2 in this case)
numZ = 2;

% eta is the convergence rate. Chosen optimally by hit and trial
```

```
eta = 0.1;

% dimensionality of input data is 64
dim = 64;

lim = 1 / sqrt(numX);

% w1 stores the weights from input to hidden layer. Chosen optimally between
% -1/root(dim) to +1/root(dim)
w1 = -lim + 2 * lim * rand(numX, numNH);

lim = 1 / sqrt(numNH);

% w2 stores the weights from hidden to output layer. Chosen optimally
% between -1/root(nH) to +1/root(nH)
w2 = -lim + 2 * lim * rand(numNH + 1, numZ);

% keep a limit on maximum number of iterations just in case we diverge
MAX_ITER = 10000;

iter = 0;

while(error > 0.1 && iter < MAX_ITER)

    % Increment iteration count
    iter = iter + 1;

    % Calculate net for each hidden unit for each input
    netJ = sample * w1;

    % Apply activation function sigmoid to netJ
    yJ = 1./(1 + exp(-netJ));

    % Apppend y0=1 to each output to account for bias weights for hidden to
    % output layer
    yJ = [yJ ones(size(yJ,1),1) ];

    % Calculate net for each hidden unit for each output
    netK = yJ*w2;

    % Apply activation function sigmoid to netK
    zK = 1./(1 + exp(-netK));

    % Compute error as 0.5*||t-z||^2
    error = 0.5.*((norm(res - zK))^2)

    % Compute delta2 as (t-z)*f'(netK) and we know for sigmoid f'=f(1-f)
    delta2 = (res - zK).*(zK.*(1 - zK));

    % Compute delta1 as summation over k(delta2*w2)*f'(netJ)
    delta1 = (delta2*w2').*(yJ.*(1 - yJ));

    % Discard the last column as it is extra, due to bias weights from
    % hidden to output layer
    delta1(:,size(delta1,2)) = [];

    % Change in weights is given by eta.delta.x and eta.delta.y
    dw1 = eta.*(sample'*delta1);
    dw2 = eta.*(yJ'*delta2);

    % Compute new weights
    w1 = w1 + dw1;
```

```
  w2 = w2 + dw2;

end

w1;
w2;
```

Weights for number of hidden units as 6:

```
w1 =


   -0.1056    0.0999   -0.0815    0.0848    0.0305
    0.1012    0.1022   -0.1095    0.0018   -0.1062
   -0.0546    0.0259   -0.0233   -0.0966    0.1167
    0.0285   -0.0337   -0.0092   -0.0024    0.0275
    0.0405    0.0246   -0.0743    0.0249   -0.0291
   -0.0750    0.0421    0.0924   -0.1024   -0.1174
    0.1150    0.0986    0.0245    0.1196    0.0894
    0.0413   -0.1032   -0.1192    0.0383    0.0259
    0.0103    0.0098    0.0999   -0.0097    0.0870
   -0.0368   -0.6527    0.5044    0.3765    0.2023
   -0.4348   -0.1305   -0.0669   -0.2295    0.3840
   -0.3669   -0.0348   -0.1057   -0.0336    0.3751
    0.1082   -0.0814    0.1593    0.0292   -0.1472
    0.1629   -0.1080    0.1169    0.1647   -0.0345
    0.0590   -0.1225    0.0288   -0.0640   -0.0872
    0.0203   -0.1356   -0.0954    0.1304    0.0788
   -0.0035   -3.4548    2.9167    2.0068    0.2511
   -0.7493   -4.9180    4.1010    2.3519    1.3865
   -2.2367   -0.9678    0.2344   -0.6482    2.5767
   -1.7879   -0.6291    0.2127   -0.6136    2.0109
    0.6869   -0.3823    0.5292    0.5977   -0.6232
    0.7569   -0.0768    0.2292    0.3381   -0.7297
    0.0464   -1.9151    1.7165    1.0580   -0.0049
    0.2473   -3.8707    3.2062    2.3377    0.0862
    0.6790   -5.7834    5.0524    3.6540   -0.1857
```

0.0420	-4.3438	3.7922	2.5519	0.1188
-1.1358	-1.0243	0.7336	0.0069	1.1645
-0.6256	-4.0067	3.3687	2.0716	0.9307
1.4908	-2.9228	2.7873	2.5000	-1.3778
2.1503	-0.3757	0.9709	1.5416	-2.3375
1.2575	-2.5215	2.5226	2.0876	-1.2537
0.2657	-5.8696	5.0134	3.3120	0.4634
0.5226	-5.4342	4.9549	3.3340	-0.0772
1.0486	-5.3030	4.7575	3.5232	-0.6490
1.2805	-4.4175	4.0266	3.2483	-1.0569
1.3424	-5.6169	5.1136	3.8892	-1.0347
1.6253	-5.1121	4.8243	3.8132	-1.4234
0.9165	-1.6400	1.4914	1.4783	-0.9826
-1.1393	-3.1766	2.6299	1.2188	1.4143
-1.2739	-5.0191	3.8990	1.9258	1.9874
0.6508	-2.0833	1.9657	1.6334	-0.3865
1.7384	-4.0513	3.9021	3.3531	-1.7506
2.6468	-3.2717	3.3231	3.3356	-2.6361
1.9389	-2.2429	2.4657	2.6048	-2.0933
1.4908	-4.6954	4.2414	3.4357	-1.2421
-0.7538	-5.1748	4.2667	2.3749	1.3694
-1.1060	-5.2187	4.3677	2.4245	1.6349
-0.1936	-2.4688	1.9915	1.3379	0.4985
0.8282	-0.2452	0.3076	0.5238	-0.7544
1.1225	-0.2258	0.4581	0.7400	-1.0451
0.4828	-0.1644	0.2063	0.5072	-0.7107
0.4163	0.0447	0.1479	0.3318	-0.6365
0.3935	-0.3498	0.3499	0.2818	-0.4728
0.0626	-1.6006	1.3620	0.9515	0.2361
0.0109	-1.3720	1.0947	0.7085	0.2597
0.0221	-0.1592	0.2503	0.0429	-0.0885
-0.0463	-0.1286	-0.0302	-0.0094	-0.1359
-0.0636	-0.1141	-0.0119	-0.0059	-0.0643

0.0940	-0.0443	0.0044	0.0737	0.0661
-0.0199	-0.1004	-0.1095	-0.1153	0.0061
-0.0031	-0.0825	-0.0672	0.0567	0.0636
-0.0099	-0.0322	-0.0954	0.0932	-0.0826
0.0039	-0.1151	-0.1003	-0.0535	0.0432
-0.0570	0.0523	0.0976	0.0392	0.0297
0.6278	-5.8847	5.1192	3.6721	-0.3885

w2 =

-12.3852	-13.4197
-12.8251	-15.1118
-12.5723	-17.0805
-14.3059	-13.6774
-16.5653	-17.3229
-27.5850	-31.8622

Weights for number of hidden units as 10:

w1 =

0.0166	0.0128	-0.0093	-0.1180	-0.0619	-0.0936	0.0297	0.0038	0.1066
0.0635	-0.1121	-0.0926	0.0524	0.1125	0.0364	-0.1033	-0.1117	0.0035
0.0049	-0.0481	0.0123	0.1013	0.0312	-0.0157	-0.0311	0.0859	-0.0767
0.0215	0.1164	0.1174	0.0915	-0.0733	0.0282	-0.0782	0.0237	0.0257
-0.0773	0.1079	-0.0145	-0.0952	-0.0976	0.0594	0.0776	0.0478	-0.0755
0.0009	-0.0305	0.0227	0.1138	0.0165	-0.0492	0.0558	0.0368	-0.0224
-0.1123	0.0294	-0.0973	-0.0148	-0.0564	-0.1143	0.0437	0.0309	0.0883
-0.1110	0.0156	-0.0750	0.0939	-0.1072	0.0839	0.1035	0.1135	-0.0177
-0.0412	0.0826	-0.0843	0.0912	-0.0854	-0.0327	0.0802	-0.0808	0.1064
-0.0900	-0.0663	0.2644	0.2278	-0.3292	-0.2637	-0.1135	-0.3548	-0.3128

-0.0157	-0.0679	0.3399	0.3912	-0.1258	-0.0251	-0.5855	-0.2678	-0.4381
-0.0134	0.1323	0.1337	0.2789	-0.0126	-0.1061	-0.3609	-0.1799	-0.2409
0.0649	-0.0028	-0.1028	-0.1754	0.1034	0.1357	0.0996	0.1258	-0.0039
0.0627	-0.0899	-0.1428	-0.1497	-0.0388	0.0923	0.1833	-0.0350	0.0495
0.0750	-0.0696	0.1246	-0.0308	0.0990	0.0728	0.0858	0.0479	0.1062
0.0365	0.0422	0.0137	-0.0824	0.0201	0.0274	0.1115	0.0940	-0.1152
-0.5060	-1.0387	1.4688	0.8485	-1.0537	-1.4654	0.0962	-1.4565	-0.8169
-0.9803	-1.3665	2.8078	2.1208	-1.6755	-2.3201	-0.9921	-2.6380	-2.1441
-0.4857	0.2628	1.9945	1.7603	-0.8807	-0.5998	-2.3540	-1.9155	-2.4853
-0.5372	0.1896	1.5970	1.5594	-0.6661	-0.4724	-1.8660	-1.5046	-2.0053
0.0332	-0.2711	-0.0429	-0.2022	0.0181	-0.0101	0.3287	0.1779	0.4469
0.1684	-0.1927	-0.3971	-0.3057	0.2326	0.1868	0.6751	0.4457	0.5619
-0.3324	-0.7005	0.8072	0.5977	-0.5853	-0.7374	0.1577	-0.7569	-0.3006
-0.4751	-1.2755	1.5207	1.1356	-0.9096	-1.5466	-0.0323	-1.6247	-0.8820
-0.7863	-1.8168	2.3021	1.3815	-1.3918	-2.4515	0.2924	-2.0017	-0.9844
-0.5754	-1.2877	2.0643	1.3279	-1.2586	-1.7981	-0.1002	-1.8363	-1.0059
-0.4272	-0.1797	1.3184	1.0096	-0.5873	-0.6679	-1.0568	-1.0471	-1.4077
-0.6275	-1.2143	2.2479	1.5461	-1.3353	-1.8956	-0.6244	-2.0857	-1.6767
-0.0784	-1.0869	0.4826	0.0679	-0.5306	-1.0366	1.1370	-0.3465	0.5130
0.3455	-0.3083	-1.0737	-1.1431	0.4627	0.0382	1.6886	0.9443	1.8436
-0.2477	-0.9304	0.2472	0.0451	-0.2587	-0.7841	1.0271	-0.2800	0.3882
-0.9257	-1.7213	2.7059	1.7516	-1.5502	-2.5822	-0.2403	-2.4557	-1.4094
-0.8109	-1.7766	2.1142	1.2844	-1.2817	-2.2477	0.3144	-2.0386	-1.0478
-0.6765	-1.7293	1.5912	1.0099	-1.2345	-2.0730	0.8278	-1.4829	-0.3798
-0.4916	-1.5404	0.9854	0.3084	-0.9102	-1.7537	1.2673	-0.9780	0.1843
-0.7077	-1.9928	1.4991	0.7041	-1.2432	-2.1738	1.1142	-1.4879	-0.1743
-0.4098	-1.8405	1.2784	0.6910	-1.0758	-2.1164	1.3382	-1.2197	0.1482
-0.0849	-0.6553	0.0094	0.0242	-0.1721	-0.5159	0.7334	-0.1502	0.4661
-0.7063	-0.8923	2.2103	1.5965	-1.2180	-1.4627	-1.2560	-2.0625	-1.9480
-1.0886	-1.4728	3.2183	2.2698	-1.8822	-2.4801	-1.4843	-2.8361	-2.5756
-0.1227	-0.7491	0.7423	0.4522	-0.5545	-0.8707	0.3048	-0.5912	-0.2705
-0.1504	-1.4474	0.6500	0.0388	-0.6480	-1.5247	1.5841	-0.6778	0.5373
0.1229	-1.4174	-0.2740	-0.7502	-0.0954	-0.9631	2.3623	0.3285	1.6187

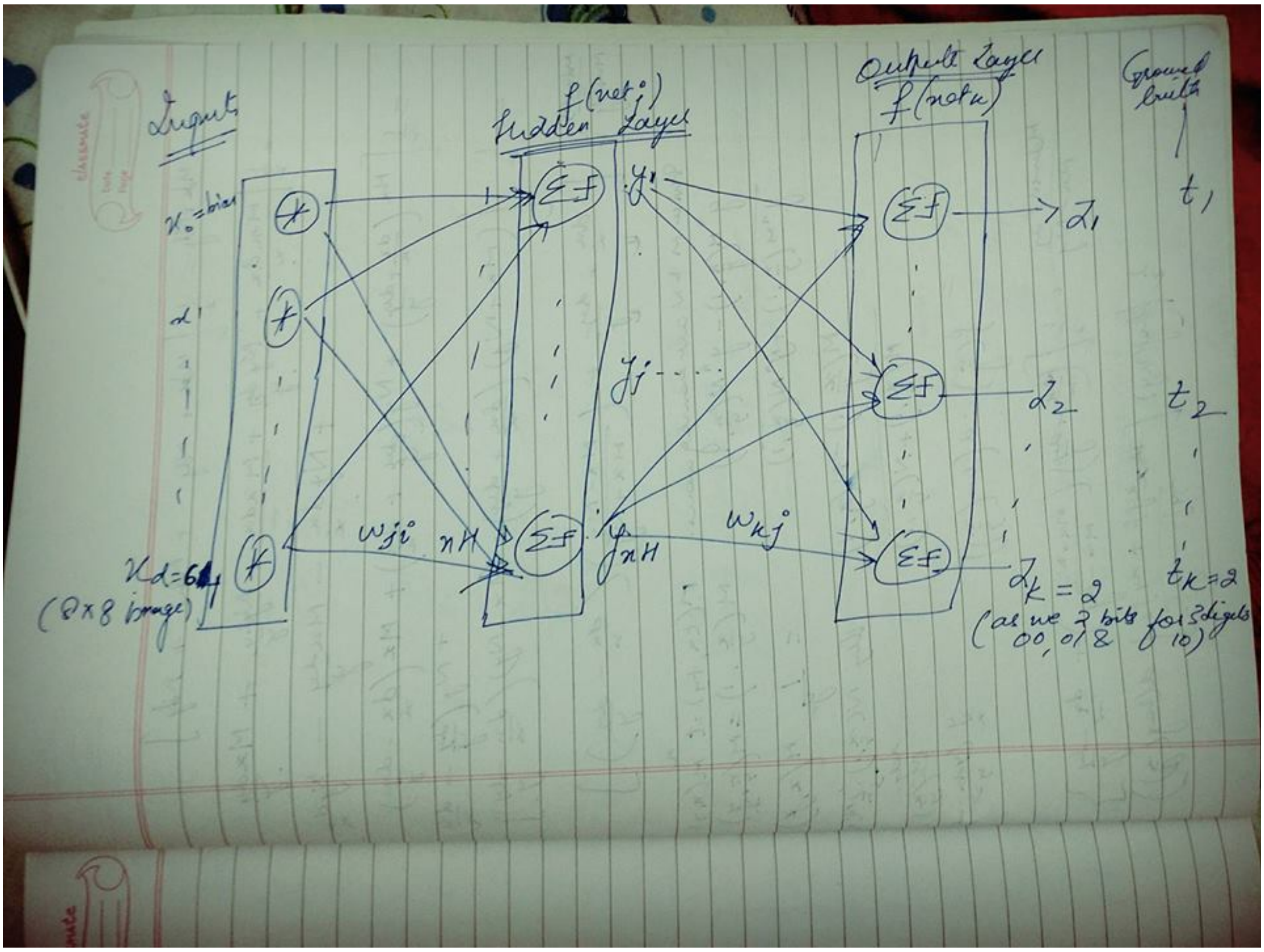
0.0442	-0.9522	-0.2534	-0.4346	-0.1979	-0.7678	1.7372	0.2435	1.2125
-0.3914	-1.5347	1.2549	0.5113	-0.9061	-1.7843	1.2379	-1.1784	0.0850
-0.9822	-1.4478	2.9854	2.0412	-1.7570	-2.4728	-0.8105	-2.7044	-1.9267
-1.0024	-1.4080	3.1920	2.1413	-1.6779	-2.5977	-0.8764	-2.8733	-2.3320
-0.4656	-0.6825	1.2121	0.8831	-0.6704	-1.1608	-0.1437	-1.1439	-0.6592
0.1486	-0.0398	-0.2585	-0.3102	0.1484	-0.0463	0.5439	0.3302	0.4686
0.2185	-0.1567	-0.5903	-0.6116	0.1812	0.1466	0.8460	0.4474	0.8185
-0.0254	-0.1870	-0.2675	-0.3811	0.1026	0.1215	0.5851	0.3417	0.4933
0.0165	-0.0268	-0.2935	-0.3392	0.2194	-0.0617	0.4080	0.3486	0.4361
-0.0339	-0.0904	-0.0764	-0.2027	0.1433	-0.0279	0.2851	0.0870	0.3222
-0.2186	-0.4458	0.6840	0.5130	-0.4589	-0.8358	0.1054	-0.6531	-0.4385
-0.1545	-0.3620	0.6638	0.3054	-0.3787	-0.6215	0.0208	-0.6758	-0.2899
0.0530	-0.1044	0.1177	0.0036	0.0463	-0.0793	0.1108	-0.0423	-0.1113
-0.1143	-0.0048	0.0186	0.0194	0.0275	0.0873	-0.0437	-0.0199	0.0695
-0.0671	0.0614	-0.0726	-0.0745	0.0790	0.0357	0.1191	0.0652	0.1418
-0.0077	0.0182	-0.0421	-0.0445	0.0403	-0.0612	-0.0015	-0.0512	-0.0423
-0.0304	0.1135	-0.0780	-0.0861	-0.0076	0.1012	0.0227	0.0451	0.0784
-0.0399	-0.0821	-0.0998	-0.0281	-0.0701	0.0668	-0.1138	-0.0266	0.0719
-0.1090	0.1018	-0.0534	0.0989	0.0256	0.0157	0.0161	-0.0270	-0.0939
0.0653	0.0631	-0.0363	0.0969	-0.0789	0.0993	0.0334	-0.0733	-0.0857
-0.0243	-0.0535	0.0089	-0.0266	-0.0756	-0.0330	-0.0005	0.0561	-0.0213
-0.7590	-1.8510	2.4211	1.5084	-1.3985	-2.5154	0.2365	-2.0399	-0.9278

w2 =

-15.9522	-13.2000
-16.4580	-12.1466
-15.7398	-12.8080
-14.1719	-6.8688
-15.0634	-11.0636
-14.1170	-11.0320
-15.8345	-12.9936

-14.0170	-12.1743
-11.8233	-12.0016
-31.0210	-22.8075

Architecture:



For digit 3:

netJ after learning is over:

netJ =

-20.1909	-10.6518	-6.5516	3.3833	-45.5422	-14.1243	10.8588	-7.2598	10.7733
-18.6678	-9.9293	-6.2250	3.0808	-41.8349	-13.3444	10.4309	-6.7445	9.9801
-20.2052	-10.7234	-6.8046	3.3180	-45.6702	-14.2886	11.1848	-7.4907	11.1719
-19.2020	-10.2771	-6.4855	3.2430	-43.5406	-13.6414	10.4893	-6.9936	10.4764
-18.2196	-9.8182	-6.0973	2.8126	-41.2494	-12.9229	10.1732	-6.8448	9.9141
-20.7613	-11.0210	-6.6557	3.4996	-46.7613	-14.7731	11.4444	-7.7184	10.9999

-20.4237	-10.8073	-6.5408	3.2289	-46.0019	-14.3716	11.0984	-7.5988	11.2264
-21.2443	-11.3042	-6.9319	3.4590	-47.9144	-15.1821	11.8322	-7.8649	11.3207
-18.2471	-9.5996	-5.8952	2.8618	-41.6945	-13.2435	10.0362	-6.8090	9.6546
-21.1422	-11.4010	-7.1454	3.5041	-48.6568	-15.2717	11.8842	-7.9283	11.5826
-17.7296	-9.3144	-5.9009	2.9109	-39.9429	-12.4911	9.8697	-6.5680	9.5597
-19.8611	-10.5864	-6.7629	3.4267	-45.4748	-14.3751	11.0026	-7.3519	10.9859
-18.7448	-10.0525	-6.6486	3.1037	-43.6378	-13.8006	10.5153	-6.9943	10.3650
-20.9891	-10.9089	-6.5680	3.3091	-46.9218	-14.8202	11.4150	-7.7831	11.0512
-16.7818	-8.6562	-5.5048	2.6181	-37.4853	-12.0649	9.4308	-5.9882	8.9910
-16.2348	-8.6387	-5.4532	2.8642	-36.7417	-11.5729	8.7890	-6.0350	8.9552
-18.8899	-10.1230	-6.1659	3.0671	-42.4993	-13.3829	10.3521	-7.0233	10.2698
-21.2112	-11.2273	-6.8086	3.5966	-48.1203	-15.2170	11.8228	-7.8104	11.3058
-17.9507	-9.3306	-5.8109	3.0027	-40.5697	-12.8205	9.7372	-6.5930	9.2831
-19.2462	-10.1679	-6.1542	3.2345	-43.1050	-13.8466	10.7630	-6.9692	10.4295
-22.2998	-11.9324	-7.3194	3.5773	-50.5113	-16.0846	12.4797	-8.1616	11.8471
-18.8772	-10.1058	-6.5137	3.3799	-43.1300	-13.5605	10.4024	-7.1394	10.5643
-22.3822	-11.9739	-7.5315	3.6646	-51.1223	-16.1588	12.5665	-8.2680	12.0626
-20.8829	-10.9692	-6.6985	3.6566	-47.4917	-14.8301	11.6718	-7.7544	11.3187
-19.6015	-10.5007	-6.5589	3.3761	-45.1146	-14.1958	10.9758	-7.3457	10.5823
-17.9544	-9.7350	-5.9730	2.9540	-40.2470	-12.6514	9.7962	-6.7908	9.9884
-21.6725	-11.3128	-6.7499	3.6702	-48.7702	-15.1440	11.9155	-7.9762	11.6079
-18.2303	-9.4144	-6.1464	2.7606	-41.2084	-12.8757	9.7025	-6.7750	9.4941
-20.6867	-10.9489	-6.6020	3.8762	-47.2310	-14.4929	11.1460	-7.7212	11.5820
-20.8006	-11.2217	-6.9539	3.3010	-47.1111	-14.9891	11.7083	-7.6356	11.3051
-18.2802	-9.7886	-6.3154	3.0586	-42.0651	-13.1559	10.1514	-6.9484	10.1100
-18.5463	-9.5000	-5.9207	3.0603	-42.8189	-13.4840	10.5011	-6.9008	9.8941
-21.0894	-11.4099	-7.0753	3.5998	-47.1685	-14.6646	11.6799	-7.8652	11.4266
-18.8835	-10.0602	-6.3468	3.2777	-43.5829	-13.4352	10.4014	-7.2245	10.4690
-20.6681	-11.0387	-7.0766	3.5823	-47.4198	-14.8570	11.3492	-7.5284	11.3447
-17.7066	-9.4485	-6.0836	3.0775	-40.5307	-12.6870	9.8421	-6.5233	9.4685
-18.8581	-10.1852	-6.4905	3.0766	-43.0103	-13.4879	10.4527	-6.7972	10.0398
-18.7746	-9.9790	-6.3167	3.3786	-42.9314	-13.5468	10.3494	-7.1060	10.2452
-20.2996	-10.8282	-6.7363	3.3334	-45.7026	-14.1980	11.3070	-7.1252	10.8516

-17.9013	-9.5490	-5.8626	3.4240	-40.8746	-12.7738	9.8801	-6.7410	10.0633
-19.8354	-10.7124	-6.9193	3.4024	-45.5447	-14.2802	10.9837	-7.2286	10.7655
-20.0900	-10.3226	-6.4818	3.1172	-45.0880	-14.5132	11.1615	-7.0737	10.6718
-20.1263	-10.3274	-6.0807	3.5499	-44.7676	-13.8840	11.1148	-7.4670	10.6562
-21.1275	-11.4242	-7.0361	3.6128	-47.7011	-15.2215	11.8617	-7.6758	11.4603
-17.5046	-9.0408	-5.6765	3.0176	-39.9946	-12.7710	10.0759	-6.4199	9.3966
-19.4559	-10.0830	-6.5341	3.1168	-44.3708	-14.1820	10.9708	-7.0381	10.4829
-19.0957	-9.8867	-6.1353	3.5134	-42.4340	-13.1970	10.6012	-6.9276	10.2730
-20.1842	-10.8576	-6.9447	3.4229	-45.9165	-14.7168	11.4042	-7.3957	10.8629
-17.2221	-9.0001	-5.7100	2.8261	-38.9623	-12.1817	9.6096	-6.1146	9.3937
-22.0872	-11.6732	-7.2630	3.7092	-50.6340	-15.9560	12.3116	-8.0686	12.0045
-18.9380	-10.0438	-6.1441	3.3750	-42.6964	-13.4586	10.2507	-6.9290	10.3413
-18.1453	-9.5113	-5.8078	3.4082	-40.9026	-12.9794	9.9318	-6.7138	9.6383
-23.7843	-12.7475	-7.8551	3.7665	-54.3973	-17.0741	13.3949	-8.7423	12.8802
-22.6677	-12.0960	-7.3096	3.6295	-51.0905	-16.0314	12.6318	-8.1422	12.1795
-22.8755	-12.2510	-7.6893	3.7825	-52.2714	-16.3745	12.9115	-8.4917	12.6156
-21.5817	-11.5656	-7.1348	3.7721	-49.1150	-15.3023	11.9159	-7.7358	11.9713
-18.0056	-9.1752	-5.6969	2.9326	-40.2093	-12.5977	9.8357	-6.4327	9.4736
-19.5577	-10.2974	-6.2481	3.5032	-44.1167	-13.9775	10.7985	-7.1309	10.5624
-16.9931	-8.8380	-5.5835	3.2108	-38.6809	-12.0294	9.4563	-6.4034	9.4137
-15.8722	-8.1884	-5.0530	3.0673	-35.7306	-11.1149	8.6079	-5.7197	8.5969
-17.3614	-8.9883	-5.7146	3.1648	-39.4131	-12.4928	9.7234	-6.3926	9.3864
-18.9243	-9.9827	-6.0068	3.3811	-42.5416	-13.2204	10.2763	-7.1319	10.3725
-22.4397	-12.0904	-7.1010	3.9337	-50.3546	-15.6730	12.1520	-8.2026	12.0185
-21.2316	-11.5108	-7.1821	3.3427	-48.1954	-15.0755	11.5459	-7.9741	11.9074
-19.5489	-10.4880	-6.4050	3.3543	-44.7612	-13.9812	11.0222	-7.1752	10.8645
-22.8978	-12.5757	-7.7003	3.8170	-51.9669	-16.2108	12.6614	-8.4644	12.5571
-21.7680	-11.7570	-6.8929	3.1586	-48.7299	-15.2559	12.1507	-7.8928	11.7718
-18.2363	-9.6648	-5.3975	3.0479	-40.5483	-12.6605	9.9334	-6.8850	9.8165
-22.7268	-12.2472	-7.2381	3.4666	-51.0811	-16.1872	12.5614	-8.3497	12.1413
-21.0467	-11.0081	-6.8279	3.2801	-47.7130	-14.9323	11.4806	-7.7596	11.5322
-21.4854	-11.3978	-6.9140	3.6247	-48.8763	-15.1759	11.6961	-8.0536	11.8703
-17.7899	-9.2597	-5.4901	3.0995	-39.8405	-12.5016	9.6341	-6.5709	9.4565

-19.5579	-10.3725	-6.3348	3.0054	-44.0460	-13.7907	10.5028	-7.3244	10.6836
-22.8045	-12.2556	-7.4577	3.8125	-51.4672	-16.2344	12.5311	-8.3607	12.5016
-22.9492	-12.3462	-7.4949	3.8439	-52.6521	-16.4068	12.7813	-8.4417	12.7853
-22.0813	-11.9996	-7.4082	3.5605	-49.4434	-15.5042	12.2703	-7.8139	11.9537
-21.7592	-11.6388	-6.9245	3.3351	-48.2321	-15.0406	12.2719	-7.9380	11.7589
-20.2857	-10.8629	-6.7818	3.3801	-45.2489	-14.1046	11.3083	-7.1562	10.9777
-22.3947	-11.9891	-7.3351	3.6053	-51.1153	-16.1097	12.6347	-8.2156	12.1490
-22.8786	-12.2148	-7.2949	3.8188	-51.9906	-16.1448	12.6596	-8.5215	12.5351
-20.5432	-11.1599	-6.8785	3.2466	-46.2794	-14.2152	11.5202	-7.3118	11.5468
-17.9095	-9.3998	-5.7379	3.0629	-40.0850	-12.6025	9.5779	-6.7912	9.5860
-17.3362	-9.0786	-5.7893	3.1553	-39.3255	-12.5591	9.4182	-6.4200	9.3445
-20.2699	-10.7412	-6.8148	3.2634	-46.0517	-14.2633	11.3872	-7.7428	11.1512
-17.0472	-9.1969	-5.6678	2.7731	-38.4789	-12.1385	9.1603	-6.4905	9.2202
-22.3215	-12.0832	-7.8853	3.6654	-51.5906	-16.1142	12.5052	-8.1287	12.5399
-18.4751	-10.0034	-6.1472	3.0132	-41.3402	-12.9306	10.1436	-6.8484	9.8695
-19.6263	-10.0767	-6.3569	3.2281	-44.3904	-13.8265	10.8713	-7.4273	10.6402
-20.7956	-11.0799	-7.0085	3.2588	-46.9299	-14.9196	11.6685	-7.7099	11.3331
-20.5558	-11.0243	-7.1407	3.3776	-47.4100	-14.8463	11.4080	-7.7030	11.5784
-21.2092	-10.9566	-6.5725	3.3271	-47.4353	-14.9063	11.5301	-7.7006	11.0508
-20.2085	-10.8053	-6.6768	3.1471	-45.6553	-14.2725	10.9995	-7.5629	11.0528
-19.0597	-10.1776	-6.0856	2.8527	-42.2787	-13.3852	10.3911	-7.1791	10.1589
-20.5834	-10.9762	-6.9175	3.4538	-46.6948	-14.9333	11.4682	-7.4223	11.3761
-20.3140	-10.5157	-5.9986	3.5382	-44.7995	-13.8883	10.8759	-7.5503	10.6830
-20.2460	-10.8462	-6.9972	3.5525	-46.7126	-14.6318	11.2362	-7.5589	11.3705
-15.8255	-8.1748	-5.2226	2.4356	-35.0961	-11.0291	8.3673	-5.9129	8.4704
-18.4256	-10.2266	-6.0371	2.8574	-40.8705	-13.0360	10.1679	-6.9246	9.9974
-22.2317	-11.8931	-7.5139	3.5497	-50.9480	-16.0918	12.6787	-8.1884	11.9989
-22.6948	-12.0513	-7.3764	3.7964	-51.7254	-16.1661	12.4442	-8.4699	12.1374
-21.0347	-11.1905	-6.7132	3.2317	-47.3850	-14.7666	11.6296	-7.6542	11.0918
-19.3785	-10.2502	-6.5533	3.0138	-44.1862	-14.0582	10.7883	-6.9845	10.2377
-21.0359	-11.0411	-6.6939	3.4432	-47.4673	-15.0215	11.7400	-7.6673	11.2519
-21.9146	-11.7340	-7.4000	3.5617	-50.0710	-15.9372	12.2501	-8.0585	11.8908
-19.1091	-10.3580	-6.3397	2.9808	-42.9109	-13.5716	10.3411	-7.1788	10.4796

-23.1662	-12.4290	-7.5906	3.8662	-52.8321	-16.6120	12.8650	-8.5349	12.4648
-18.1787	-9.6909	-5.9528	3.1307	-41.3777	-13.0737	10.2572	-6.8080	10.0596
-22.0322	-11.7628	-6.8963	3.7020	-49.5198	-15.3341	12.0561	-8.2850	11.8978
-21.3124	-11.3684	-7.0878	3.3028	-48.0923	-14.9304	11.6461	-8.0594	11.8567
-22.2109	-11.7705	-7.1253	3.3855	-49.9691	-15.7623	12.1604	-8.2376	12.0126
-19.9092	-10.7241	-6.4508	3.4086	-44.3659	-13.7167	10.9593	-7.1447	10.7220
-22.8017	-12.3271	-7.6374	3.7858	-52.0850	-16.4990	12.5234	-8.4296	12.5273
-18.7195	-9.9608	-6.1431	3.2306	-42.6920	-13.4809	10.5190	-7.0355	9.8156
-21.0359	-11.0411	-6.6939	3.4432	-47.4673	-15.0215	11.7400	-7.6673	11.2519
-19.6345	-10.4487	-6.5748	3.5600	-45.2314	-14.1449	10.8066	-7.2470	10.9735
-21.5627	-11.5922	-7.2886	3.3811	-48.8888	-15.3116	11.8284	-8.2428	12.1243
-21.4863	-11.5999	-7.0028	3.5844	-48.5994	-15.2978	11.8536	-7.8958	11.6867
-20.4594	-10.9779	-6.7729	3.4260	-46.5063	-14.6390	11.4563	-7.2771	11.0816
-19.6766	-10.4733	-6.4740	3.2526	-44.1514	-13.9325	10.7442	-7.1982	10.8535
-19.1162	-10.3107	-6.3719	3.3068	-42.9865	-13.6304	10.6668	-6.9563	10.2689
-19.3456	-10.3947	-6.4648	3.2875	-44.2089	-14.0821	10.8573	-7.0704	10.5369
-19.8667	-10.5950	-6.4220	3.5188	-44.6367	-14.2294	10.8888	-7.2872	10.6818
-20.3326	-10.9483	-6.5735	3.6326	-46.0873	-14.2430	11.0518	-7.6596	11.2241
-18.7064	-10.0804	-6.4297	2.9663	-42.3422	-13.1970	10.0853	-7.0314	10.2938
-16.1477	-8.5756	-5.5658	2.9255	-36.7105	-11.3885	8.7409	-5.9415	8.9673
-18.7622	-10.2135	-6.3940	3.2070	-42.8743	-13.4987	10.6206	-7.0457	10.6042
-22.1921	-11.7334	-7.4044	3.6057	-50.3239	-15.8968	12.3515	-8.1339	12.0520
-17.1780	-8.7217	-5.6995	2.8331	-39.1727	-12.4548	9.5099	-6.0855	9.3644
-15.8367	-8.3209	-5.4124	2.7328	-36.2463	-11.4283	8.6019	-6.0321	8.8119
-19.2316	-10.1431	-6.4841	3.1358	-43.7156	-13.6820	10.8392	-7.0442	10.4837
-21.3587	-11.2778	-7.2854	3.4942	-48.5683	-15.2521	11.9820	-7.8407	11.8175
-15.1201	-7.5307	-5.0602	2.4044	-34.8079	-11.2586	8.4538	-5.4423	8.1975
-21.4961	-11.5543	-6.8198	3.4276	-48.0032	-15.0686	11.8041	-8.1564	11.6096
-22.5304	-12.2045	-7.5103	3.5916	-51.1810	-16.1648	12.8188	-8.2603	12.4233
-19.4127	-10.3040	-6.3880	3.0127	-43.4830	-13.6217	10.5403	-7.3460	10.9005
-19.9050	-10.6838	-6.2767	3.0862	-44.1056	-14.1077	11.1190	-7.2974	10.6142
-21.4475	-11.3622	-6.7844	3.5868	-48.1015	-15.1898	11.9752	-7.7840	11.4686
-19.4995	-10.3508	-5.9041	3.2614	-43.3835	-13.3991	10.7694	-7.4472	10.8254

-20.3374	-10.8586	-6.4565	3.1442	-45.2759	-14.3259	11.1491	-7.4803	10.9845
-17.5644	-9.1482	-5.7180	2.9793	-39.7771	-12.4280	9.7203	-6.5558	9.3037
-20.0294	-10.4275	-6.5483	3.0432	-44.9789	-14.2391	11.0292	-7.4408	10.7648
-17.8086	-9.2066	-5.9926	2.9113	-40.0855	-12.8231	9.5886	-6.7010	9.6893
-15.7663	-8.4649	-5.4411	2.4769	-35.3930	-11.1793	8.3847	-6.0001	8.6487
-19.4649	-10.4584	-6.8425	3.1631	-44.4485	-14.0319	10.5835	-7.2994	11.0827
-20.2480	-10.5004	-6.1099	3.4861	-45.0245	-14.0537	11.0314	-7.4978	10.6583
-20.1001	-10.8309	-6.7090	3.1857	-45.6242	-14.2219	10.9800	-7.5908	11.2572
-18.1872	-9.6095	-6.3291	2.7539	-41.8388	-13.2261	10.2681	-6.8230	10.0622
-19.7633	-10.5100	-6.7875	3.1396	-45.4240	-14.3148	11.0218	-7.1915	10.6222
-19.1146	-10.0372	-6.1729	3.2356	-43.0348	-13.7207	10.7575	-6.9771	10.1400
-20.1724	-10.7505	-6.7568	3.3745	-46.1088	-14.5038	11.3656	-7.2900	11.0006
-20.3131	-10.8904	-6.9804	3.4563	-46.6814	-14.8583	11.3848	-7.5491	11.0924
-18.6765	-10.0081	-6.4135	3.4146	-42.8886	-13.5346	10.2105	-6.9900	10.3700
-19.4120	-10.3562	-6.6159	3.2176	-44.3084	-13.9376	10.8297	-7.1402	10.5994
-22.7795	-12.3385	-7.7290	3.9156	-51.8168	-16.1341	12.5846	-8.2901	12.5438
-24.0933	-13.0451	-8.0290	3.7822	-54.9174	-17.3260	13.4851	-8.8986	12.9996
-21.1036	-11.5370	-7.0842	3.3068	-47.7521	-15.2453	11.7462	-7.8694	11.5666
-20.7873	-11.1033	-7.0286	3.0929	-47.3544	-15.0429	11.5601	-7.5146	11.0055
-19.8808	-10.4553	-6.5839	3.3146	-45.2727	-14.0810	11.0129	-7.4167	10.4798
-21.7629	-11.7313	-7.3494	3.2844	-49.1654	-15.5859	12.0685	-8.1055	11.8265
-18.1335	-9.6245	-6.1530	2.7744	-42.0593	-13.1891	10.2031	-6.5817	9.6521
-18.1137	-9.7338	-5.8374	3.1456	-40.7474	-12.8525	9.9439	-6.7845	9.6828
-19.5204	-10.4405	-6.3274	3.3533	-44.1057	-14.0010	10.8930	-7.2421	10.5140
-22.0653	-11.9723	-7.6977	3.5021	-50.5794	-15.7773	12.6614	-8.0573	12.1381
-24.9040	-13.4103	-8.2096	3.7825	-56.7199	-17.7209	14.0730	-9.1251	13.7658
-16.0992	-8.4491	-5.0642	2.6959	-35.8076	-11.1441	8.8479	-6.0648	8.7656
-15.9395	-8.5660	-5.5480	2.5960	-36.0355	-11.3445	9.0570	-5.6353	8.4891
-20.8729	-11.1308	-7.1392	3.6495	-48.1468	-15.0384	11.5683	-7.7291	11.8047
-21.8516	-11.6892	-7.5961	3.6121	-50.1085	-15.7244	12.2006	-8.1095	12.1374
-19.3272	-10.4769	-7.0204	2.8050	-44.2610	-14.2825	10.7899	-6.9728	10.6742
-18.1339	-9.7717	-6.1450	2.6777	-40.5668	-12.6662	9.9955	-6.6215	10.1116
-19.4209	-10.3699	-6.6699	3.2631	-44.7951	-14.1619	10.9426	-7.1783	10.6000

-18.4212	-9.4189	-5.6560	3.1172	-41.3072	-12.9836	10.1489	-6.8543	9.6709
-21.3224	-11.4553	-7.0828	3.3980	-48.5308	-15.3132	11.8601	-7.9185	11.7665
-21.9769	-12.0351	-7.8626	3.2445	-50.2269	-15.9602	12.3282	-8.2663	12.3168
-19.7629	-10.5629	-6.7210	3.5931	-45.5595	-14.2228	10.8484	-7.4125	11.0497
-17.8703	-9.3912	-6.1494	2.5342	-40.7354	-12.8120	9.9925	-6.4964	9.4716
-17.4335	-9.1109	-5.7647	3.0668	-39.5676	-12.5190	9.3307	-6.5470	9.5067
-19.9283	-10.6307	-6.7461	3.3305	-45.4435	-14.6016	11.1512	-7.3421	10.7078
-19.3778	-10.1909	-6.6804	3.0265	-44.2449	-13.8568	10.8298	-7.0639	10.2988
-20.9704	-11.1898	-6.7298	3.7019	-47.5330	-14.7909	11.7668	-7.7092	11.4119
-21.6416	-11.6554	-6.9912	3.6617	-48.9063	-15.5104	12.0719	-7.9324	11.6462
-19.2351	-10.2091	-6.2087	3.2923	-43.3618	-13.4915	10.3013	-7.4069	10.3739
-21.8842	-11.7108	-7.1711	3.4662	-49.2032	-15.6485	12.1724	-8.0476	11.6388
-20.2099	-10.8407	-6.5934	3.6167	-45.5254	-14.1156	11.0938	-7.3266	11.1059
-18.7746	-9.9790	-6.3167	3.3786	-42.9314	-13.5468	10.3494	-7.1060	10.2452
-21.7596	-11.4531	-7.3397	3.5290	-49.9570	-15.7755	12.0039	-8.1059	11.9024
-21.4600	-11.4045	-6.9741	3.5534	-48.6723	-15.1754	11.7917	-8.1073	11.5470
-22.2998	-11.9324	-7.3194	3.5773	-50.5113	-16.0846	12.4797	-8.1616	11.8471
-22.2738	-11.9284	-7.4637	3.3942	-51.3721	-16.0637	12.7789	-8.1644	12.2858
-18.5146	-9.8323	-6.7186	2.8237	-42.4749	-13.6259	10.3018	-6.5213	10.0152
-18.0280	-9.3694	-5.9277	3.3901	-41.2125	-12.8262	9.8432	-6.6814	9.8719
-20.4503	-11.1252	-7.1062	3.4235	-46.5498	-14.6022	11.4100	-7.7341	11.4339
-19.2704	-10.3448	-6.5396	3.3278	-44.2060	-13.9920	10.7119	-7.2589	10.4129
-19.4430	-10.2180	-6.5809	2.9465	-44.5430	-14.1813	11.0249	-7.1486	10.6827
-18.4543	-10.0551	-6.3367	2.8200	-41.8182	-13.2320	10.2427	-6.7926	10.0785
-19.0980	-10.1409	-6.3283	3.1504	-42.5579	-13.3996	10.6488	-6.8986	10.2642
-20.3261	-10.6346	-6.4661	3.1548	-45.7464	-14.2780	11.1960	-7.4831	10.7654
-22.4880	-12.1588	-7.3038	3.7754	-50.4744	-15.9503	12.3565	-8.2483	12.1627
-20.4937	-10.8797	-6.5639	3.6159	-46.0710	-14.6360	11.2209	-7.4575	11.1160

For digit 3:

netK after learning is over:

netK =

-56.6060	-48.6977
-56.2848	-48.4132
-56.4166	-48.5307
-56.1579	-48.3028
-56.2861	-48.4181
-56.5871	-48.6800
-56.5931	-48.6863
-56.5995	-48.6906
-56.3437	-48.4706
-56.6160	-48.7016
-56.1407	-48.2896
-56.4728	-48.5783
-56.4624	-48.5705
-56.8543	-48.9116
-56.6138	-48.7034
-55.9920	-48.1640
-56.1953	-48.3403
-56.5905	-48.6800
-56.4058	-48.5282
-56.5600	-48.6512
-56.7225	-48.7956
-56.5517	-48.6486
-56.7126	-48.7869
-56.6470	-48.7286
-56.4570	-48.5639
-55.9288	-48.1097
-56.7383	-48.8080
-56.3196	-48.4506
-56.5159	-48.6143
-56.6812	-48.7585
-56.5082	-48.6111
-56.5385	-48.6387

-56.7237	-48.7960
-56.4951	-48.5975
-56.6756	-48.7526
-56.6215	-48.7093
-56.2680	-48.4022
-56.4501	-48.5598
-56.7057	-48.7818
-56.3431	-48.4672
-56.4903	-48.5930
-56.7156	-48.7898
-56.6501	-48.7343
-56.6661	-48.7442
-56.6626	-48.7464
-56.7633	-48.8313
-56.6826	-48.7602
-56.5993	-48.6868
-56.4138	-48.5309
-56.7237	-48.7950
-56.3169	-48.4429
-56.1696	-48.3122
-56.8068	-48.8673
-56.7731	-48.8389
-56.7567	-48.8231
-56.7186	-48.7900
-56.6146	-48.7053
-56.4365	-48.5438
-56.5227	-48.6240
-55.9621	-48.1317
-56.5528	-48.6483
-56.5348	-48.6331
-56.7145	-48.7872
-56.6751	-48.7567
-56.3728	-48.4923

-56.6838	-48.7623
-56.7133	-48.7901
-56.4491	-48.5641
-56.6643	-48.7462
-56.6869	-48.7646
-56.6180	-48.7044
-56.4551	-48.5703
-56.5339	-48.6354
-56.7174	-48.7878
-56.6400	-48.7211
-56.7039	-48.7777
-56.7973	-48.8635
-56.5732	-48.6634
-56.7339	-48.8044
-56.6736	-48.7522
-56.3813	-48.4971
-56.6698	-48.7556
-56.4427	-48.5563
-56.6804	-48.7588
-56.3644	-48.4918
-56.7816	-48.8447
-56.5102	-48.6159
-56.7858	-48.8521
-56.6246	-48.7105
-56.5951	-48.6837
-56.6558	-48.7386
-56.4594	-48.5664
-56.5623	-48.6604
-56.6143	-48.7004
-56.6079	-48.6983
-56.5246	-48.6232
-55.9814	-48.1560
-56.3926	-48.5102

-56.7245	-48.7971
-56.7324	-48.8030
-56.7277	-48.8017
-56.5749	-48.6700
-56.7184	-48.7917
-56.7329	-48.8046
-56.5780	-48.6737
-56.7135	-48.7871
-56.5570	-48.6554
-56.7477	-48.8186
-56.7257	-48.8015
-56.7035	-48.7797
-56.5530	-48.6481
-56.7669	-48.8322
-56.4209	-48.5335
-56.7184	-48.7917
-56.5353	-48.6319
-56.6704	-48.7509
-56.7508	-48.8194
-56.5218	-48.6199
-56.5387	-48.6367
-56.5559	-48.6509
-56.6354	-48.7191
-56.4061	-48.5189
-56.6575	-48.7398
-56.4380	-48.5549
-55.8456	-48.0330
-56.5371	-48.6379
-56.8058	-48.8649
-56.4145	-48.5336
-56.1582	-48.3097
-56.5095	-48.6126
-56.8258	-48.8825

-56.4518	-48.5652
-56.7675	-48.8352
-56.7401	-48.8099
-56.6005	-48.6913
-56.6242	-48.7108
-56.6874	-48.7641
-56.4046	-48.5230
-56.4893	-48.5952
-56.2080	-48.3500
-56.7145	-48.7889
-56.5393	-48.6358
-56.2880	-48.4245
-56.6598	-48.7407
-56.6574	-48.7420
-56.4878	-48.5937
-56.6809	-48.7620
-56.6094	-48.6995
-56.5134	-48.6142
-56.4999	-48.6001
-56.6248	-48.7098
-56.4769	-48.5835
-56.3746	-48.4940
-56.8068	-48.8663
-56.7727	-48.8387
-56.7087	-48.7831
-56.5990	-48.6904
-56.5367	-48.6356
-56.6675	-48.7499
-56.3541	-48.4797
-56.1860	-48.3343
-56.3298	-48.4549
-56.8146	-48.8757
-56.8284	-48.8859

-56.1927	-48.3470
-56.2851	-48.4218
-56.6221	-48.7068
-56.7484	-48.8172
-56.5080	-48.6095
-56.3927	-48.5130
-56.4570	-48.5636
-56.6502	-48.7357
-56.5237	-48.6225
-56.7979	-48.8592
-56.5047	-48.6058
-56.6602	-48.7468
-56.3027	-48.4356
-56.5806	-48.6716
-56.7364	-48.8097
-56.5988	-48.6871
-56.6807	-48.7578
-56.5410	-48.6402
-56.7335	-48.8052
-56.5990	-48.6889
-56.4501	-48.5598
-56.7439	-48.8125
-56.7060	-48.7821
-56.7225	-48.7956
-56.6343	-48.7181
-56.3466	-48.4681
-56.3414	-48.4654
-56.5487	-48.6441
-56.5069	-48.6084
-56.7277	-48.8001
-56.0521	-48.2160
-56.5747	-48.6654
-56.7480	-48.8205

-56.7286	-48.7987
-56.5188	-48.6158