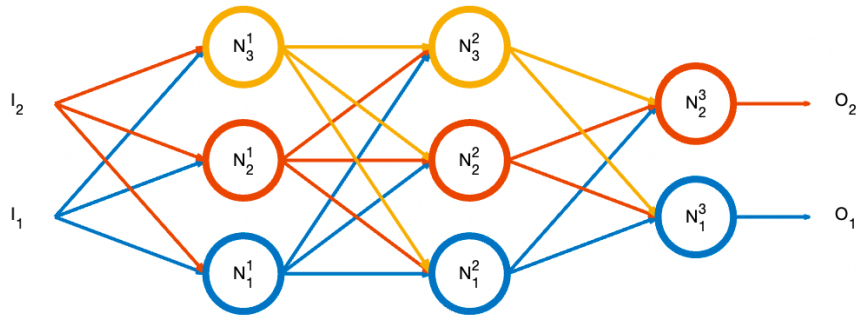
Name: Elena Pan
Date: Oct 2nd, 2022

Project 1

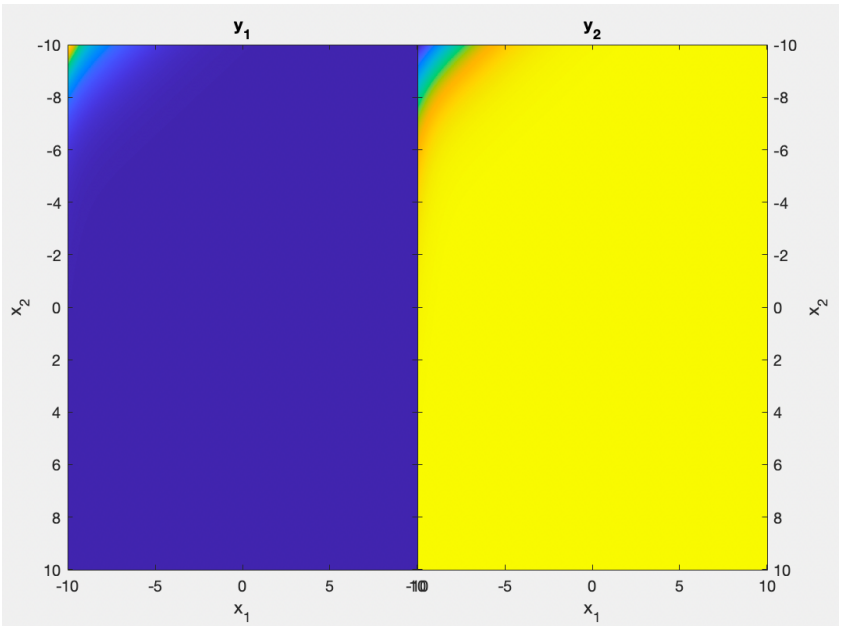
1. Construct a neural network with structure shown in Figure 1 using the routines provided and by selecting randomly generated weights. Generate plots of the level curves for the output y as function of x_1 and x_2. Example is provided in Matlab script 'test_Basic_NNTools.m'.

   - **Neural Network (2 inputs, 2 hidden layers, 2 outputs):**



The Neural Network has 2 input vectors, 2 hidden layers, and will generate binary outputs.
Code: `[network]=createNetwork(2,[3,3,2])`

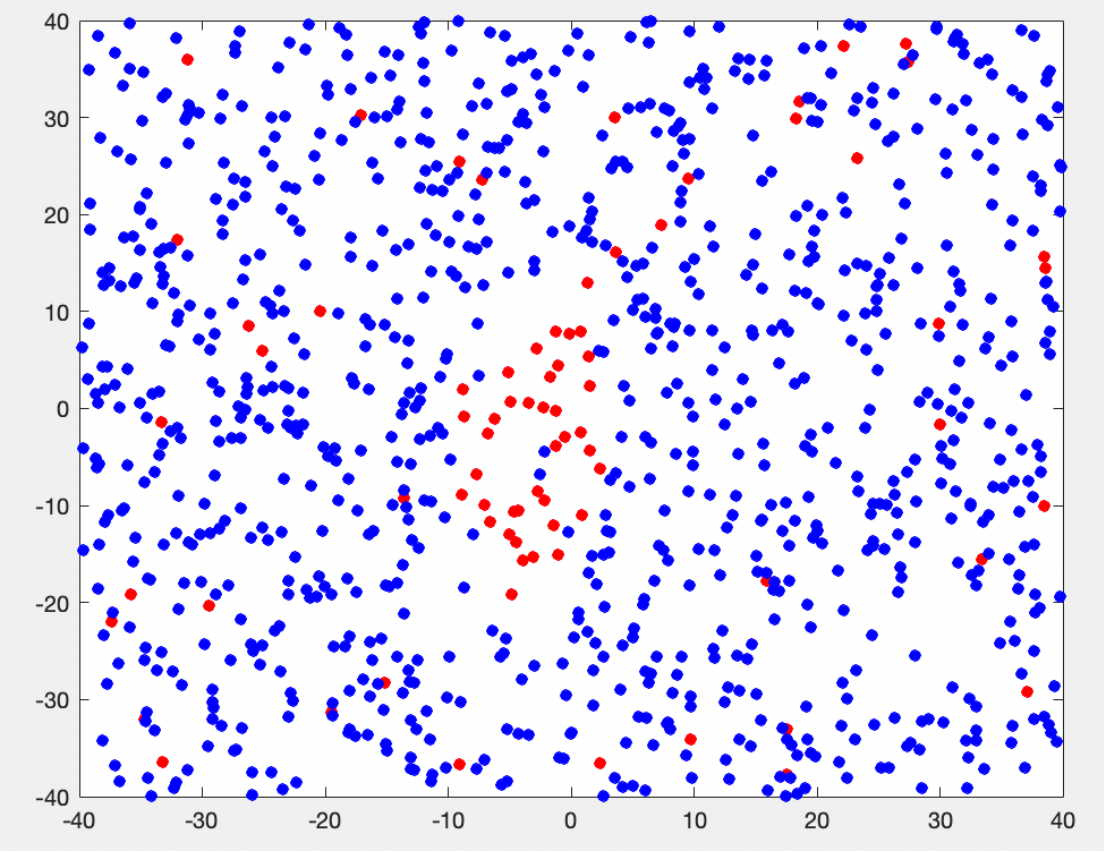
   - **Level Curves for the output y as function of x1 and x2 with random weights**



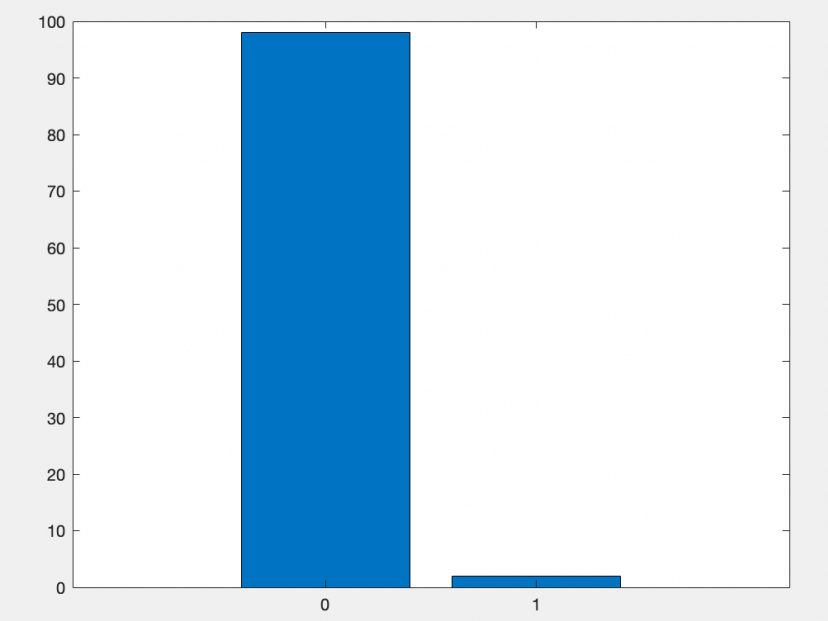
2. Generate a set of training data $\{(x_k, y_k)\}_{k=1}^{n}$ using the provided routines for synthetic data generation. Example is given in Matlab script 'test_getData.m'. When calling the script 'getData.m', you should use your own 10 digit student ID as the last input argument which is used to seed for the random number generator.

   Use own 10-digit student ID: 2471729356 as the last input argument
   Code: `[x,y]=getData(1000,2,2471729356);`

   - **A set of 1000 training data**
     - y = 0 → blue dots
     - y = 1 → red dots



- **Output y distribution**

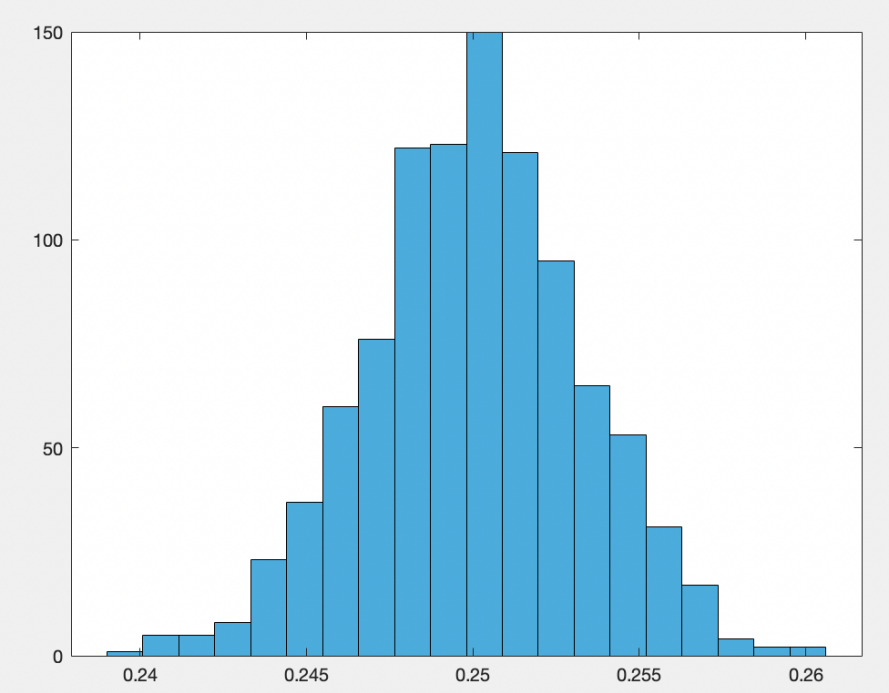


3. Write a routine that evaluate the least square cost function for a given data set and a given weights for the neural network.

```
[Weight]=getNNWeight(network);
Weight=squeeze(0.01*randn([size(Weight),nTrials]));
MSE=NaN(nTrials,1);
for iTrial=1:nTrials
    [network]=setNNWeight(network,Weight(:,iTrial));
    [yVal,~]=networkFProp(xData,network);
    MSE(iTrial)=mean((yData-yVal).^2);
end
figure;
histogram(MSE,20);
return
end
```

4. Using randomly generated weights for the neural network to estimate the distribution of the performance function for a fixed set of training data generated with 'getData.m'.

`Experiment_Basic_NNTools(1000)` → use 1000 training data



The distribution is approximately a normal distribution with mean of 0.250048 and standard deviation of 0.00321783

5. By increasing the random trials of weight find the best performance among the random weights as the size of the random samples increases. This is equivalent to optimization by random trails.

nTrials	Best Performance (minimize MSE)
1	0.2482
10	0.2475
100	0.2420
1000	0.2398
10,000	0.2391
100,000	0.2346
1,000,000	0.2346

As the nTrails increases from 1 to 1*10^6, the best performance of minimal MSE among the random weights will decrease, and eventually converge to one point, 0.2346.