ASSIGNMENT 5

Question 1

	Predicted Yes	Predicted No	Total
Actual Yes	223	43	266
Actual No	87	453	540
Total	310	496	806

Fig. Confusion Matrix

- Specificity = TN/ N = TN / (TN + FP)
  = 453 / (453 + 87) = 453 / 540 = **0.83889**
- Sensitivity = TP / P = TP / (TP + FN)
  = 223 / (223 + 43) = 223 / 266 = **0.83835**
- Accuracy = (TP + TN) / (TP + FP + FN + TN)
  = (223 + 453) / (223 + 87 + 43 + 453) = 676 / 806 = **0.83871**
- Prevalence = (TP + FN) / (TP + FP + FN + TN)
  = (223 + 43) / (223 + 87 + 43 + 453) = 266 / 806 = **0.33002**
- Recall rate = TP / P = TP / (TP + FN)
  = 223 / (223 + 43) = 223 / 266 = **0.83835**
  (Same as Sensitivity)
- Precision = TP / (TP + FP)
  = 223 / (223 + 87) = 223 / 310 = **0.71935**

- Sometimes, the cost of errors is not constant. For example, if we were predicting whether a patient has cancer, a false negative could be considered much worse than a false positive. If that were the case, which of the above statistics would you want to optimize? Briefly explain why.
  Ans: If we were predicting whether a patient has cancer, then
  False negative: We are predicting the patient doesn't have cancer even if he has.
  False positive: We are predicting the patient has cancer even if he doesn't.
  In this case, false positive is better than false negative as it is not tolerable to have more false negative. It would then be a problem as the patients having cancer would be told that they are safe even though they are not.

  As false negative i.e. Type II error is what we want to be as less as possible, we wish to optimize **Sensitivity**. Sensitivity tells us the proportion of actual positives that were predicted that way. That is, in the ROC curve, if we achieve 100% sensitivity, that means there are no false negative values as

  Sensitivity = TP / P = TP/ (TP + FN)

  If sensitivity is more, it means that false negatives are less. For the best case, if sensitivity is 100%, then TP = TP + FN, which makes FN = 0.

Question 2: ID3

	Blood Type	Gender	Family History	Doing Sports	Smoke	Age	Having Disease
01	A	M	Y	N	Y	60	+
02	A	F	Y	Y	Y	40	+
03	A	M	N	N	N	12	-
04	B	M	Y	Y	N	36	+
05	B	F	N	N	N	12	-
06	AB	M	Y	N	Y	68	+
07	AB	M	Y	Y	N	47	-
08	AB	M	Y	Y	N	47	+
09	O	F	N	N	Y	72	+
10	O	F	N	Y	N	15	-

- Training set T = {1, 2, 3, …, 10}
- **I(T)** = $-\frac{6}{10}log_2\frac{6}{10} - \frac{4}{10}log_2\frac{4}{10}$ = **0.9709505944546686**

- **Test Blood Type:**
  1. $I(T_{\{\text{Blood Type<--A}\}}) = -\frac{2}{3}log_2\frac{2}{3} - \frac{1}{3}log_2\frac{1}{3}$ = 0.9182958340544896333333333333333334
  2. $I(T_{\{\text{Blood Type<--B}\}}) = -\frac{1}{2}log_2\frac{1}{2} - \frac{1}{2}log_2\frac{1}{2} = 1$
  3. $I(T_{\{\text{Blood Type<--AB}\}}) = -\frac{2}{3}log_2\frac{2}{3} - \frac{1}{3}log_2\frac{1}{3}$ = 0.9182958340544896333333333333333334
  4. $I(T_{\{\text{Blood Type<--O}\}}) = -\frac{1}{2}log_2\frac{1}{2} - \frac{1}{2}log_2\frac{1}{2} = 1$
  5. $I(\text{Blood Type, T}) = \frac{3}{10} * I(T_{\{\text{Blood Type<--A}\}}) + \frac{2}{10} * I(T_{\{\text{Blood Type<--B}\}}) + \frac{3}{10} * I(T_{\{\text{Blood Type<--AB}\}}) + \frac{2}{10} * I(T_{\{\text{Blood Type<--O}\}})$ = **0.95097750043269378**
  6. **Gain**: G(Blood Type, T) = I(T) - I(Blood Type, T) = **0.01997309402197482**

- **Test Gender:**
  1. $I(T_{\{\text{Gender<--M}\}}) = -\frac{4}{6}log_2\frac{4}{6} - \frac{2}{6}log_2\frac{2}{6}$ = 0.9182958340544896333333333333333334
  2. $I(T_{\{\text{Gender<--F}\}}) = -\frac{2}{4}log_2\frac{2}{4} - \frac{2}{4}log_2\frac{2}{4} = 1$
  3. $I(\text{Gender, T}) = \frac{6}{10} * I(T_{\{\text{Gender<--M}\}}) + \frac{4}{10} * I(T_{\{\text{Gender<--F}\}})$ = **0.95097750043269378**
  4. **Gain**: G(Gender, T) = I(T) - I(Gender, T) = **0.01997309402197482**

- **Test Family History:**
  1. $I(T_{\{\text{Family History<--Y}\}}) = -\frac{5}{6}log_2\frac{5}{6} - \frac{1}{6}log_2\frac{1}{6} =$ 0.650022421648354166666666666666667
  2. $I(T_{\{\text{Family History<--N}\}}) = -\frac{1}{4}log_2\frac{1}{4} - \frac{3}{4}log_2\frac{3}{4} =$ 0.81127812445913285
  3. $I(\text{Family History, T}) = \frac{6}{10} * I(T_{\{\text{Family History<--Y}\}}) + \frac{4}{10} * I(T_{\{\text{Family History<--N}\}}) =$ **0.71452470277266564**
  4. **Gain**: G(Family History, T) = I(T) - I(Family History, T) = **0.25642589168200296**

- **Test Doing Sports:**
  1. $I(T_{\{\text{Doing Sports<--Y}\}}) = -\frac{3}{5}log_2\frac{3}{5} - \frac{3}{5}log_2\frac{3}{5} =$ 0.9709505944546686
  2. $I(T_{\{\text{Doing Sports<--N}\}}) = -\frac{3}{5}log_2\frac{3}{5} - \frac{3}{5}log_2\frac{3}{5} =$ 0.9709505944546686
  3. $I(\text{Doing Sports, T}) = \frac{5}{10} * I(T_{\{\text{Doing Sports<--Y}\}}) + \frac{5}{10} * I(T_{\{\text{Doing Sports<--N}\}}) =$ **0.9709505944546686**
  4. **Gain**: G(Doing Sports, T) = I(T) - I(Doing Sports, T) = **0**

- **Test Smoke:**
  1. $I(T_{\{\text{Smoke<--Y}\}}) = -\frac{4}{4}log_2\frac{4}{4} - \frac{0}{4}log_2\frac{0}{4} = 0$
  2. $I(T_{\{\text{Smoke<--N}\}}) = -\frac{2}{6}log_2\frac{2}{6} - \frac{4}{6}log_2\frac{4}{6} =$ 0.9182958340544896333333333333333334
  3. $I(\text{Smoke, T}) = \frac{5}{10} * I(T_{\{\text{Smoke<--Y}\}}) + \frac{5}{10} * I(T_{\{\text{Smoke<--N}\}}) =$ **0.55097750043269378**
  4. **Gain**: G(Smoke, T) = I(T) - I(Smoke, T) = **0.41997309402197482**

- **Test Age:**
  1. $I(T_{\{\text{Age<--(0, 20]}\}}) = -\frac{0}{3}log_2\frac{0}{3} - \frac{3}{3}log_2\frac{3}{3} = 0$
  2. $I(T_{\{\text{Age<--(20, 40]}\}}) = -\frac{2}{2}log_2\frac{2}{2} - \frac{0}{2}log_2\frac{0}{2} = 0$
  3. $I(T_{\{\text{Age<--(40, 60]}\}}) = -\frac{2}{3}log_2\frac{2}{3} - \frac{1}{3}log_2\frac{1}{3} =$ 0.9182958340544896333333333333333334
  4. $I(T_{\{\text{Age<--(60, 100]}\}}) = -\frac{2}{2}log_2\frac{2}{2} - \frac{0}{2}log_2\frac{0}{2} = 0$
  5. $I(\text{Age, T}) = \frac{3}{10} * I(T_{\{\text{Age<--(0, 20]}\}}) + \frac{2}{10} * I(T_{\{\text{Age<--(20, 40]}\}}) + \frac{3}{10} * I(T_{\{\text{Age<--(40, 60]}\}}) + \frac{2}{10} * I(T_{\{\text{Age<--(60, 100]}\}}) =$ **0.27548875021634689**
  6. **Gain**: G(Age, T) = I(T) - I(Age, T) = **0.69546184423832171**

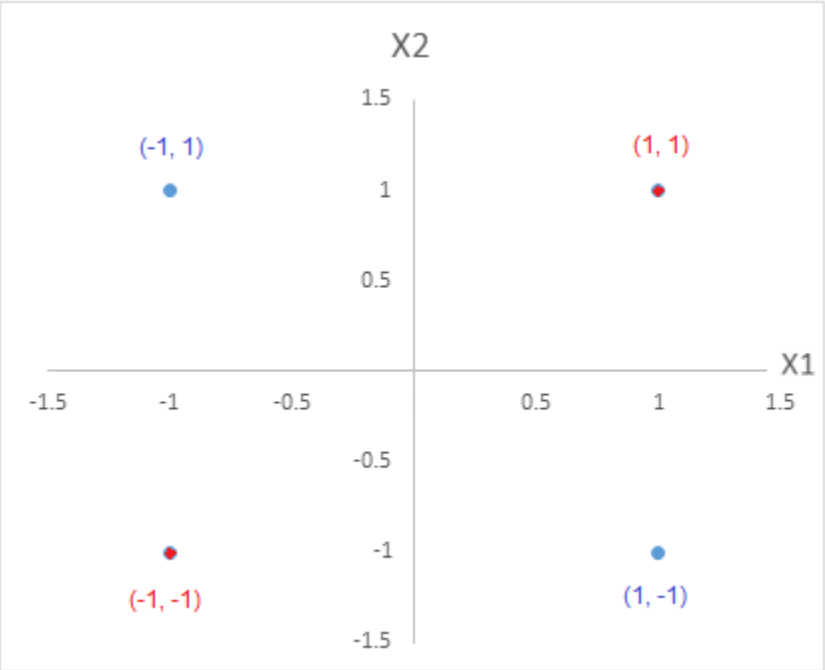
0.69546184423832171 > 0.41997309402197482 > 0.25642589168200296 > 0.01997309402197482 >= 0.01997309402197482 > 0.
Gain of attribute 'Age' is maximum. **Thus, 'Age' should be chosen as root.**

Question 3: SVM

The following table shows the XOR values of all the four data points:
[-1, -1], [-1, +1], [+1, -1], [+1, +1]

Point No.	x1	x2	x1 XOR x2
1	-1	-1	-1
2	-1	+1	+1
3	+1	-1	+1
4	+1	+1	-1

This data is not linearly separable.

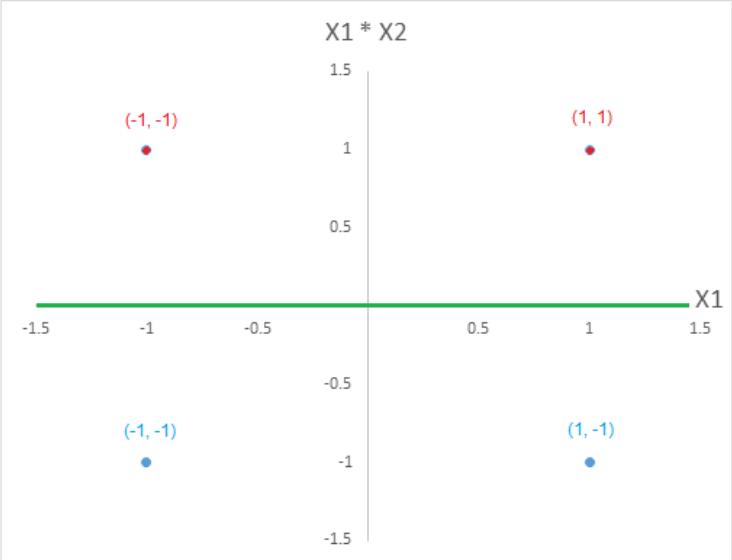


(Fig. Old space)

Red dots indicate points with x1 XOR x2 equal to -1, Blue dots indicate points with x1 XOR x2 equal to 1.

Let us add a new feature x1 * x2 and plot these four points into new space.

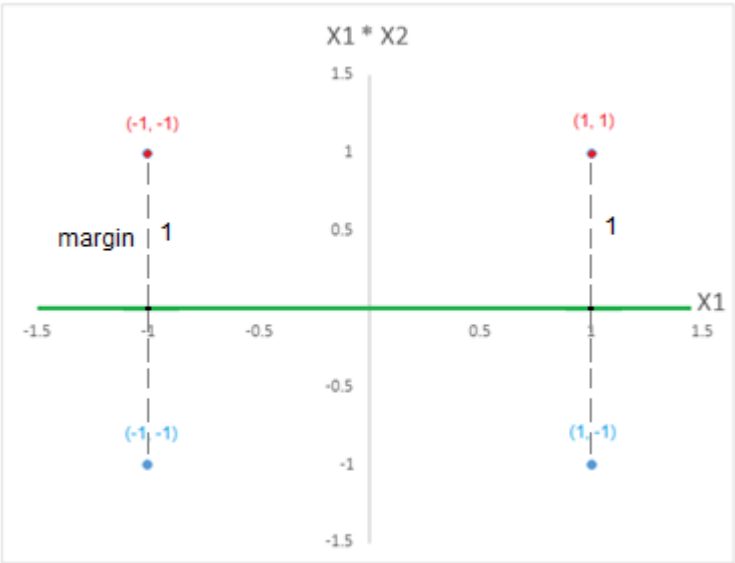
Point No.	x1	x1* x2	x1 XOR x2
1	-1	+1	-1
2	-1	-1	+1
3	+1	-1	+1
4	+1	+1	-1



(Fig. New space with maximal margin separator)

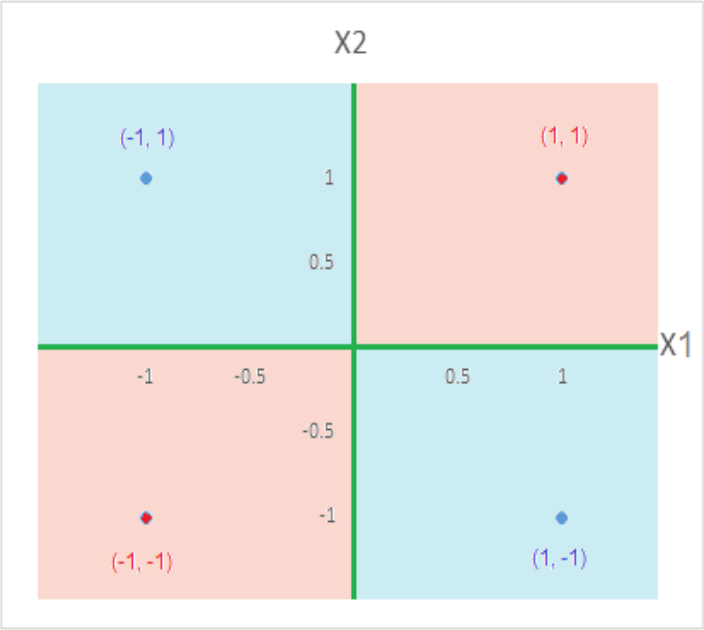
This data is linearly separable as the line overlapping with axis X1 is separating the four input points correctly per their XOR values. The separator line is the green one in the figure above. Here, it can be observed that the separator maximizes the 'margin' and so is the maximal margin separator.

Margin = geometric distance from a point to the separating hyperplane = 1
Equation of the separator in the new Euclidean space is x1 * x2 = 0.



(Fig. New space with maximal margin separator and margin)

Going back to the original space, this separator can be drawn as follows:



(Fig. Old space with maximal margin separator)

Comparison between representation of points in the old space and new space:

Point No.	Old Space ([x1, x2], x1 ⊻ x2)	New Space ([x1, x1 * x2], x1 ⊻ x2)
1	([-1, -1], -1)	([-1, +1], -1)
2	([-1, +1], 1)	([-1, -1], 1)
3	([+1, -1], 1)	([+1, -1], 1)
4	([+1, +1], -1)	([+1, +1], -1)

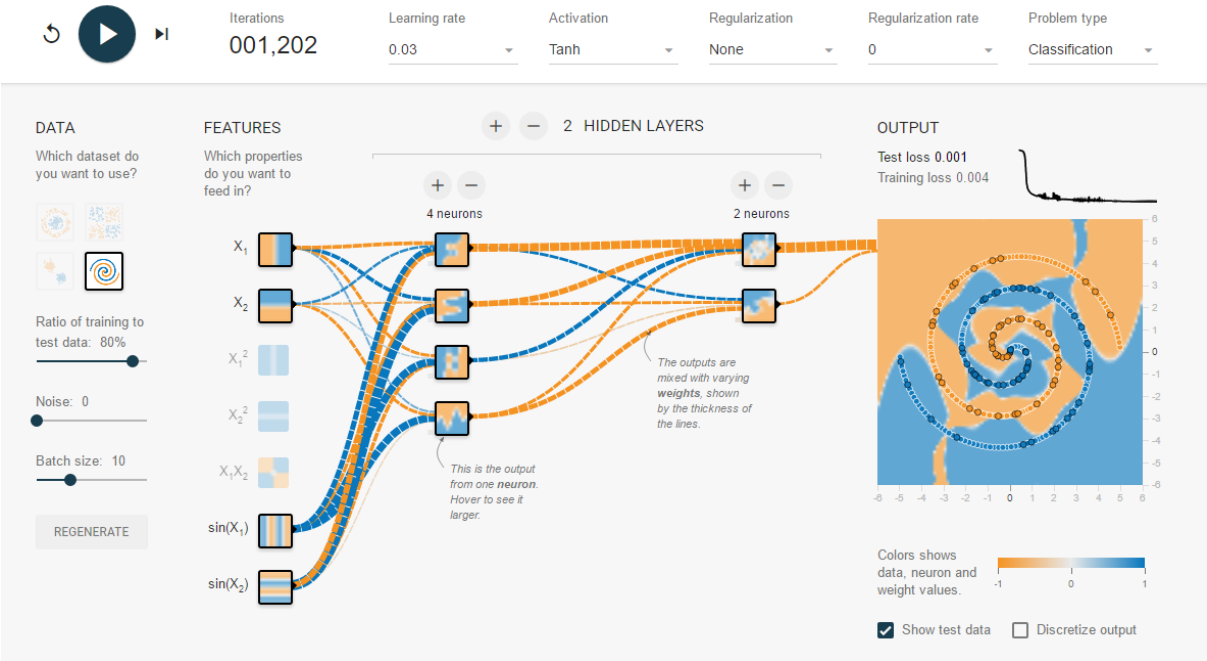
In the old space, input points were not linearly separable because of the features x1 and x2. The points with XOR value 1 were in 2nd and 4th quadrant, while points with XOR value -1 were in 1st and 3rd quadrant. Without adding any additional feature, it was impossible to separate these points, thus x1*x2 feature was added. Thus, points with x1*x2 = 1 were above the line x1 and those with x1*x2 = -1 were below the line x1. This made x1 axis, the maximal margin separator with equation x1*x2 = 0. Putting back this separator in the old space, axis x1 and axis x2 became the separator as x1*x2 = 0 is the equation of two intersecting lines which overlap with both the axes.

Separator in the old space: x1*x2 = 0
Separator in the new space: x1 = 0, x2 = 0

Bonus Problem:

1. Why does it seem to be easier for it to predict the third picture (two clumps of points by color) than the fourth (the spiral)?
   Ans:
   Third picture (two clumps of points) contains linearly separable points because we can easily find a maximal margin separator between those to clumps which can separate those classes with almost zero test loss (i.e. perfect classification). This separation can be done using training neural network using X1 and X2 features only.
   Whereas, the spiral contains both the types of points which are not linearly separable and need other features like sin(X1), sin(X2). Using only X1 and X2 is not sufficient as they only come up with the linear classifier. Thus, it requires more training and hidden layers with more neurons. So, it requires too many iterations to train it with as low test loss as possible.
   Hence, it is easier to predict the third picture as compared to the fourth one.

2. Pick some parameters that predict the spiral well and take a screen shot to show the settings and the results. Note that the value for "test loss" is the percent of points in the test data that are mispredicted. Lower test loss is better.
   Ans:



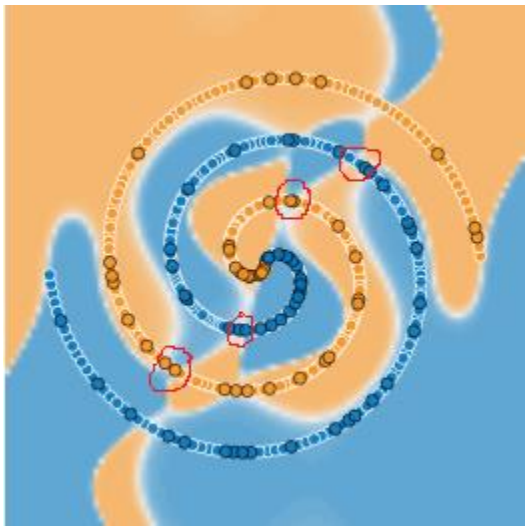
- Features: X1, X2, sin(X1), sin(X2)
- Hidden Layers: 2 (1st contains 4 neurons, 2nd contains 2 neurons)
- Learning rate: 0.03
- Activation function: tanh
- Noise: 0

- Ratio of training to test data: 80%
- # of iterations: 1202
- Batch Size: 10
- **Training Loss: 0.004**
- **Test Loss: 0.001**

3. Explain why the edges connecting the sine function outputs end up weighted more than the others.
   Ans: Thick line in the experiment indicates that the neuron has very strong impact on the next layer neurons, whereas thin line indicates that it has very low impact. In this example, as other weights come from the features like X1, X2 which are mostly used for linear classification, the spiral won't be having too much impact of them. On the other hand, sinusoidal function has curvy nature which is very useful for the spiral classification. Thus, they have a huge impact on the output. Therefore, they end up weighted more than others.

4. For each change listed below, start from the settings you found in part b and make that one change. Note the changes in test loss and the kinds of points it will mispredict. Explain why that change would have that effect on the net.
   a. Add a node in the first hidden layer.
      Ans:
      Test Loss: 0.001
      Details: If a node is added in the test loss doesn't change. It will have an extra neuron for computation. This will only increase the size of neural network, but at the time of pruning we can easily detect this extraneous neuron and just remove it. It takes less number of iterations to achieve the test loss.

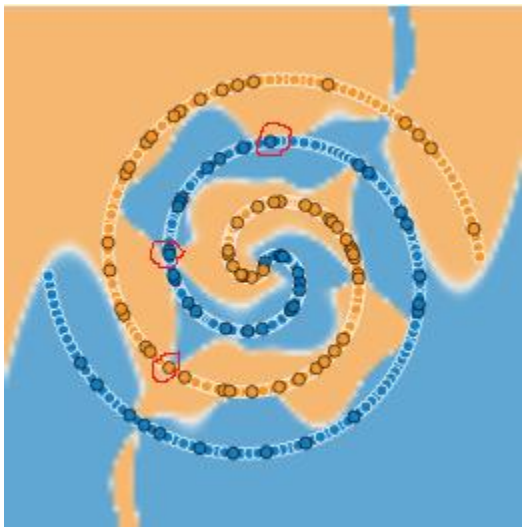
   b. Remove a node in the first hidden layer.
      Ans:
      Test Loss: 0.125



Details: If one neuron is removed from the first hidden layer, then test loss increases. The reason behind that is there are less computations going on in that hidden layer, so it becomes difficult to get lower test loss with the same number of iterations. It needs more training to get lower test loss, but even then, also it won't be reaching to the minima. The points where the hyperplane narrows, those are mispredicted.

c. Add a node in the second hidden layer.
Ans:
Test Loss: 0.001
Details: This is same as that of adding a neuron in the first hidden layer.

d. Remove a node in the second hidden layer.
Test Loss: 0.037

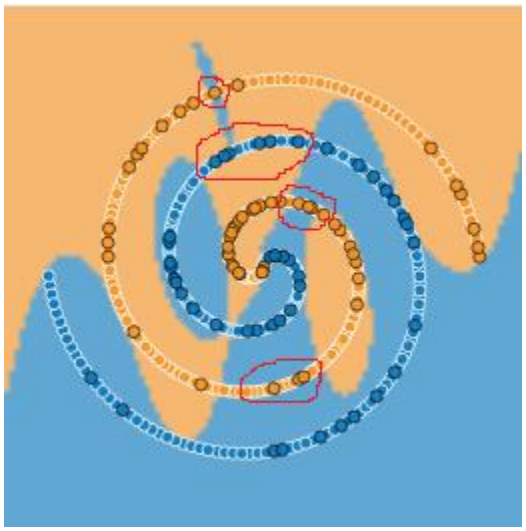


Details: If one neuron is removed from the first hidden layer, then test loss increases. The reason behind that is there are less computations going on in that hidden layer, so it becomes difficult to get lower test loss with the same number of iterations. It needs more training to get lower test loss, but even then, also it won't be reaching to the minima. The points which are at the peak of the separator hyperplane will be misclassified.

e. Make the noise in the data be 20%.
Ans:
Test Loss: 0.039

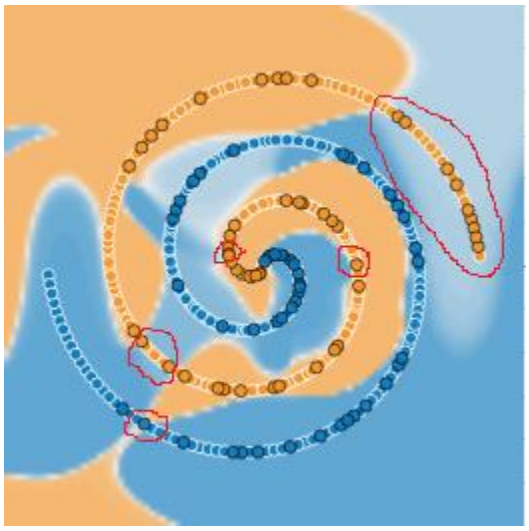
Details: As noise is increased from 0% to 20%, the test data points are no longer in the perfect spiral shape. They are scattered little a bit. So, those points which are not in the spiral trajectory will be mispredicted if we use the same settings as above.

f. Remove the sin(X2) input.
Test Loss: 0.237
Details: As sin(X2) is to be removed, the test loss increases by a greater extent.



This shows the effect of sin(X2) alone. The points which are there in the sine wave shape will be classified correctly. This is what is missing if we don't use sin(X2) as a feature. We will mispredict the points which come in that sinusoidal wave zone.

g. Add an input that you hadn't used before.
Test Loss: 0.147



Details: If a new feature x_2^2 is added, then the test loss increases. Adding this feature affects the classification as it predicts the top right region as blue instead of orange and that's why the orange points from the outer loop of the spiral get misclassified. Also, as compared to the previous settings, this one gives very much lower weights for blue class as compared to orange class.

REPORT

- **Approach:**
    - The basic approach which I followed is to start with a hidden layer with number of neurons equals to the number of input layer neurons (in this case 4) and the most basic input features.
    - Input features like X1, X2 were not sufficient to predict the points which are in the complex shapes like spiral. That is why I went for more complex features like sin(X1) and sin(X2). There I got the test loss of 0.35 which was still high.
    - Next, I tried increasing the ratio of training data to test data up to 80-20 proportion to have comparatively more test data than training data. The test loss decreased to 0.1.
    - As those features have curvy nature, those helped in getting better results. After a few iterations, it became hard to use only one hidden layer to get low test loss below 0.03.
    - Thus, I added another hidden layer with a couple of neurons and I achieved the test loss of 0.001 (least possible among all the settings).

- **What happened:**
    - Whenever number of neurons were increased the test loss went on decreasing, but as the number of iterations were also increasing, the problem of overfitting the data came into picture.
    - Very less number of hidden layers and less number of hidden layers also prevented the data from accurate classification because of less computation.
    - When the learning rate alpha was increased, it sometimes took very less time to converge. But, for most of the times, it failed to stop at the minima and it overshot the gradience (went on diverging).
    - Extraneous features also prevented the data from correct classification as they also had their respective impact on the output leading to mispredicting the points.
    - Adding noise to the data sometimes was useful to get a simple classifier instead of a very complex shaped one. But, it also decreased the test loss.

- **Conclusion:**
    - Neither too many not too less number of hidden layers and hidden neurons should be used. Trial and error method concludes that the optimal number of hidden neurons should be in between the sum of input neurons and output neurons. This will help to avoid the overfitting problem.
    - Regularization should be used to maintain the balance between accuracy (test loss) and overfitting avoidance. It would achieve the hyperplane with a less complex shape with less spikes in between.