COMP3330 Project 1.b)

1. Variations of the Two-Spiral Task

ANN Training

The two-spiral problem has two real-valued inputs corresponding to the x and y coordinates of the point, and one binary target output that classifies the input points as belonging to either of the two spirals coiling three times around the origin. The aim of the following sections is to train a multilayer perceptron (MLP) to correctly classify (x,y) coordinates. The generalisation ability of the trained classifier will be tested by evaluating it on a test set of all pixels in a section of the (x,y)-plane. The generalisation result is then visualised in images displaying blue and yellow regions as labelled by the classifier. Ideally, the image should be divided into two equally thick spiral shaped regions. The convergence times will be highlighted through the use of error curves which plot the error as a function of the epoch number.

Part A

The original spiral dataset of Lang and Witbrock was read from the file provided on Blackboard and variations of MLPs were built to see how fast and well the network can solve the task. The number of hidden layers and units, and the use of momentum were the two main aspects explored to obtain a well-trained artificial neural network (ANN). The sigmoid activation function as used for all ANNs in this experiment.

Initially, the Gradient Descent optimiser was used at a learning rate of 0.5 to achieve the two-spiral task. However, even with increased hidden layers and units the number of iterations needed to form two distinct spirals was quite high. With a 2-50-50-40-1 MLP, around 200,000 iterations was required to form two distinct spirals as shown in Figure 1.

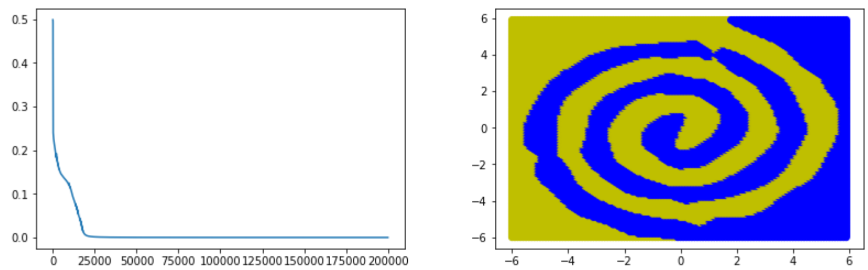


Figure 1: 2-50-50-40-1 MLP with Gradient Descent Optimiser (learning rate = 0.5) at 200,000 iterations

To reduce the iterations required, momentum was introduced into the system by using the Momentum Optimiser of the TensorFlow library. The MLP used with the Gradient Descent Optimiser was kept the same with the Momentum Optimiser, with the momentum set to 0.8. The two spirals were clearly defined after 5,000 iterations, as shown in Figure 2. This is significantly faster compared to the 200,000 iterations required with the Gradient Descent Optimiser, and with momentum it generalised better as shown by the smoother spiral bands.

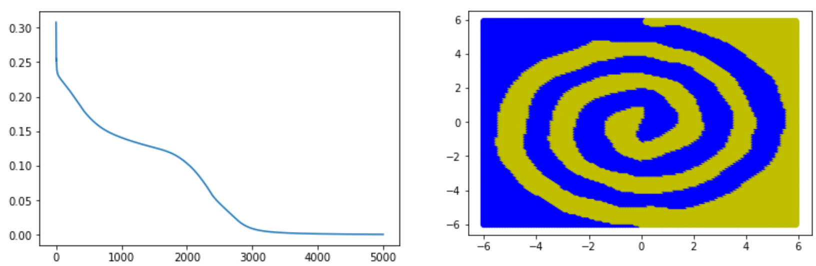


Figure 2: Resulting error curve (left) and generalisation image (right) obtained from 2-50-50-40-1 MLP with Momentum Optimiser (learning rate = 0.5, momentum = 0.8) at 5,000 iterations.

Part B

The variations of the original 2-spiral dataset were obtained using the C-routine provided on Blackboard. The radius was maintained at the default value of 6.5, as in the original dataset, but the density of points was varied. The datasets are shown below in Figure 3 for the different densities and thus number of data points. Note at a density of 1, the original 194-point dataset is obtained.

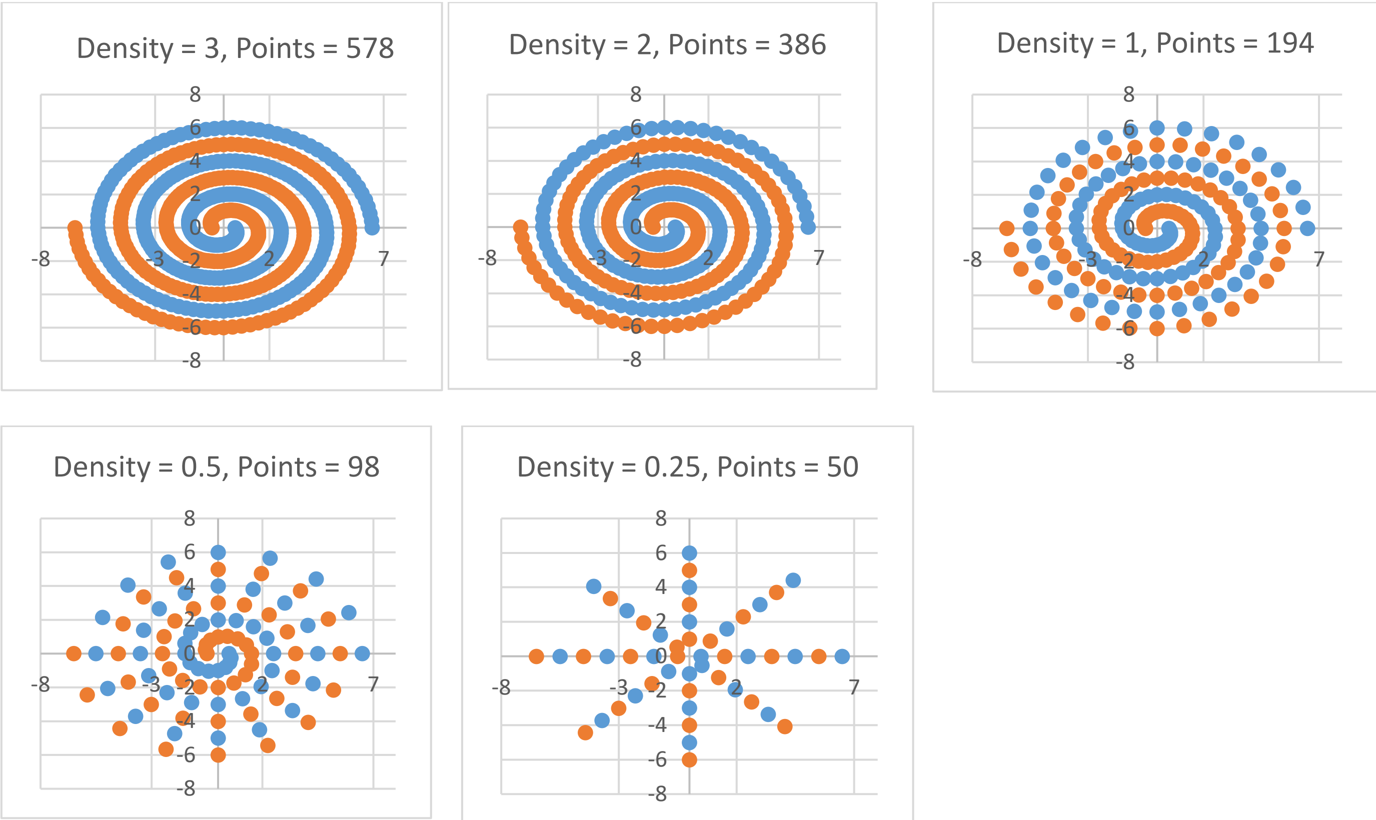


Figure 3: Variations of the two-spiral dataset with 578 points (top left), 386 points (top right), 194 points (middle left), 98 points (middle right), and 50 points (bottom)

As shown in part (a), solving the two spirals problem is easier with the introduction of momentum, and more difficult with just the gradient descent method. Thus, for this section, only the momentum optimiser is used, and the value of the momentum is varied as well as the network structure. Note that the learning rate was maintained at 0.5 for each experiment, and all artificial neural networks used have a sigmoid activation function.

Density = 3

The densest dataset consisted of 578 data points for which the ANN was trained as a 2-40-40-1 MLP with momentum 0.9. This resulted in the error curve and generalisation image below. The error curve is quite smooth and reach an error value of <0.005 after 5000 iterations. A significant portion of the points from the dataset were classified correctly and resulted in a clearly defined image of two intertwined spirals as shown in Figure 4.

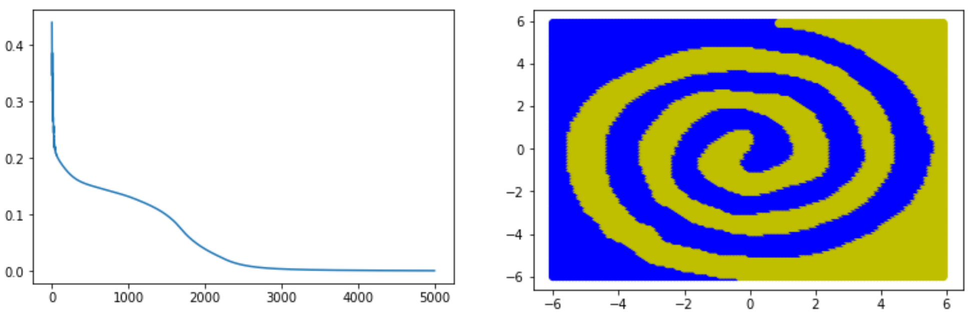


Figure 4: Resulting error curve (left) and generalisation image (right) obtained from 2-40-40-1 MLP with Momentum Optimiser (learning rate = 0.5, momentum = 0.9) at 5,000 iterations. Generated from density = 3 dataset.

Density = 2

The same ANN architecture and method was trained using the next dataset 386 data points. The 2-40-40-1 MLP with momentum 0.9 provided the error curve and generalisation image shown below. Similar to the previous case, the error curve is smooth and had an error value of <0.009 after 5000 iterations. The generalisation image shows two well-defined spirals as expected. However, multiple trials of training this network can result in higher error rates at 5000 iterations of 0.015, with defects in the two spirals. An example of this is shown in Figure 5. 6000 iterations will more consistently acquire correct classifications. The error curve from the previous case reaches the smaller error values faster than for this dataset. Due to the decrease in data points, it takes the network more iterations to achieve the two spirals.

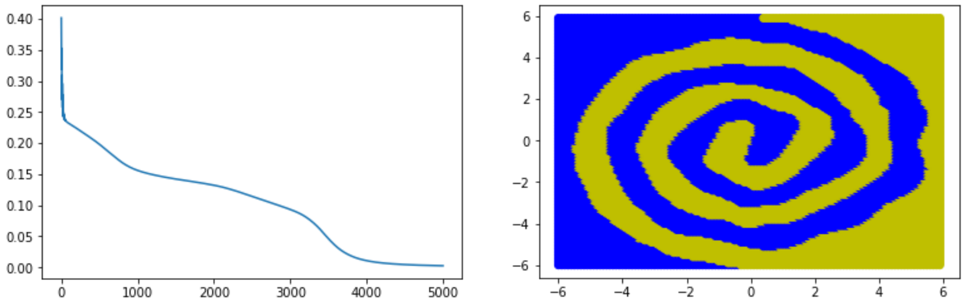


Figure 5: Resulting error curve (left) and generalisation image (right) obtained from 2-40-40-1 MLP with Momentum Optimiser (learning rate = 0.5, momentum = 0.9) at 5,000 iterations. Generated from density = 2 dataset.

Density = 1

This dataset has the same 194 points as the original dataset used in part (a). The exact same method and results were obtained. A 2-50-50-40-1 MLP with momentum 0.8 was trained with 5000 iterations to produce the error curve and generalisation image shown in Figure 6. The error value at 5000 epochs was approximately 0.001. Again, two well-defined spirals formed in the generalisation image. This dataset has significantly less data points than the other two cases, and thus an extra hidden layer of 50 hidden units was added which increased the network complexity.

2-50-50-40-1, 5000, MomentumOptimiser, learning_rate=0.5, momentum=0.8

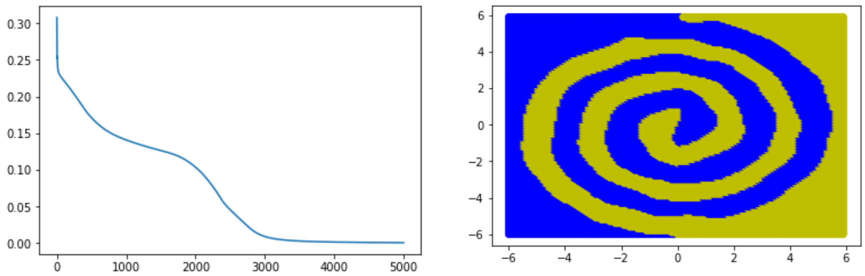


Figure 6: Resulting error curve (left) and generalisation image (right) obtained from 2-50-50-40-1 MLP with Momentum Optimiser (learning rate = 0.5, momentum = 0.8) at 5,000 iterations. Generated from density = 1 dataset.

Density = 0.5

This dataset has 98 points and a network with 4 hidden layers was used to solve the classification task. For a 2-60-70-60-40-1 MLP with momentum 0.8, the error curve and generalisation image shown in Figure 7 was obtained. Despite the error value of 0.0001, at epoch 5000, the image is not of two perfectly distinct spirals. Even if the iterations were to be increased to 10,000, there is no significant improvement in the image despite the improved error of 0.00006, as shown in Figure 8.

The spirals are forming but due to the sparser data the network does not perfectly form the two spirals. The image uses a much larger number of input points than the network was trained on and thus the areas between the training data points can be incorrectly classified.

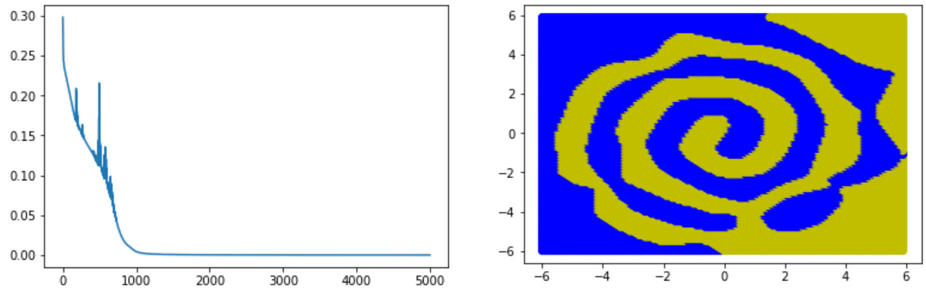


Figure 7: Resulting error curve (left) and generalisation image (right) obtained from 2-60-70-60-40-1 MLP with Momentum Optimiser (learning rate = 0.5, momentum = 0.8) at 5,000 iterations. Generated from density = 0.5 dataset.

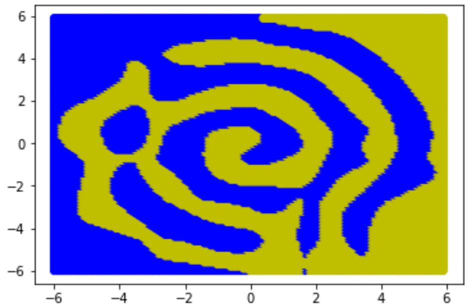


Figure 8: Generalisation image obtained from 2-60-70-60-40-1 MLP with Momentum Optimiser (learning rate = 0.5, momentum = 0.8) at 10,000 iterations. Generated from density = 0.5 dataset.

Density = 0.25

The final dataset has 50 points with the results shown in Figure 9. A 2-60-100-80-80-1 MLP with momentum 0.8 was trained to solve the task. The trained network didn't separate the (x,y)-plane into two perfect spiral regions but did fit all the training points, as indicated by the minute error value of 4e-06. Despite this, the two spirals are not distinct throughout the image, however they can be differentiated the clearest near the centre. This confirms Lang and Witbrock's initial findings (1988) of networks approximating the spiral decision boundaries better at the spiral centres [1]. This is due to the point density of the original training data being the highest at the centre, as can be seen from Figure 3.

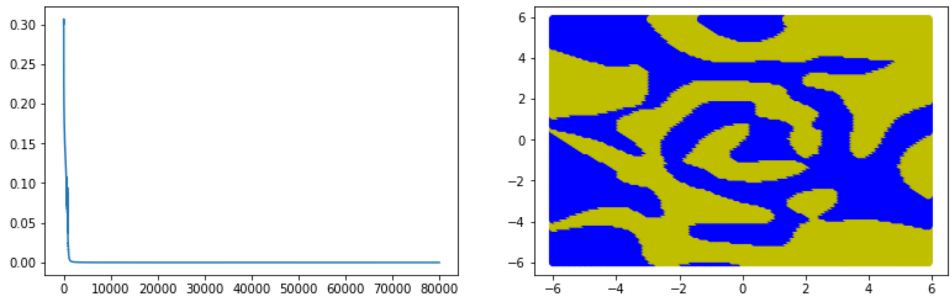


Figure 9: Resulting error curve (left) and generalisation image (right) obtained from 2-60-100-80-80-1 MLP with Momentum Optimiser (learning rate = 0.5, momentum = 0.8) at 5,000 iterations. Generated from density = 0.25 dataset.

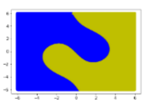
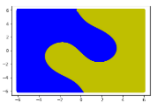
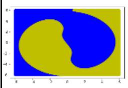
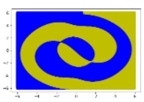
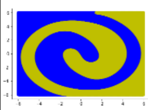
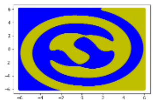
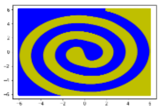
Reducing the training dataset increased the ANN architecture complexity required to solve the task. Increasing the density of the training set can result in better approximations of the spiral decision boundary [1] as does increasing the iterations. With 578 points, 5,000 iterations are enough to train a feed-forward network with two hidden layers and solve the two-spiral task.

Only two aspects were focused on in these experiments; number of hidden layers and units, and the use of momentum. The extent to which both features can be exploited is greater than what has been explored. Thus, there most likely is an ideal balance between the two resulting in a spiral image, or extra features that need to be utilised to achieve the task with sparser data. In the cases explored, the number of training points required to successfully solve the two-spiral task lies between 194 and 98.

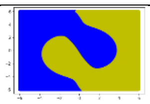
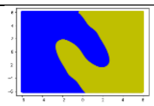
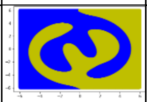
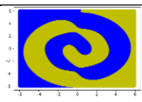
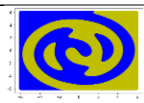
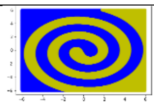
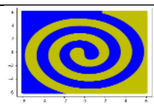
ANN vs SVM

Part C

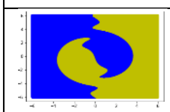
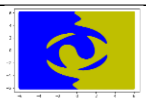
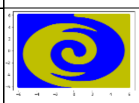
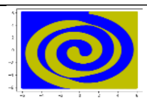
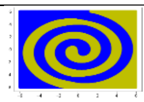
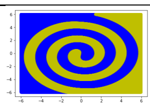
To solve task a, we sought to explore the effects of varying C against specifically chosen values for gamma for an RBF kernel Support Vector Machine. Some examples of the findings are shown below:

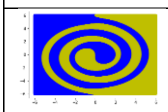
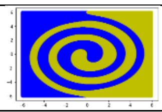
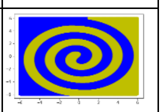
C = 0.1, gamma = 0.1	C = 0.001, gamma = 0.1 [1]	C = 25, gamma = 0.1	C = 75, gamma = 0.1	C = 1000, gamma = 0.1	C = 25000, gamma = 0.1	C = 1000000, gamma = 0.1
						

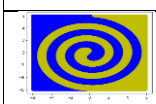
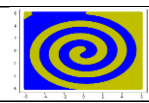
[1] note that here, reducing C produces no discernible visual change. This was repeated for all values of gamma and the same result was observed.

C = 0.1, gamma = 0.3	C = 1, gamma = 0.3	C = 2.5, gamma = 0.3	C = 5, gamma = 0.3	C = 25, gamma = 0.3	C = 50, gamma = 0.3	C = 100, gamma = 0.3 [2]
						

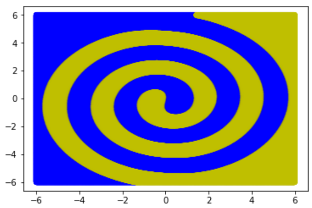
[2] notice here that the inner part of the curve is more pronounced than at C=50, however the C value has become quite high.

C = 0.1, gamma = 0.5	C = 0.5, gamma = 0.5	C = 2.5, gamma = 0.5	C = 4, gamma = 0.5	C = 5, gamma = 0.5	C = 10, gamma = 0.5
					

C = 0.1, gamma = 0.75	C = 0.5, gamma = 0.75	C = 10, gamma = 0.75
		

C = 0.1, gamma = 2	C = 0.5, gamma = 2
	

In general, from these images we can see that smaller values of gamma lead to less accurate results, as well as a higher level of fine-grain variation as C is increased from 0.1. This produces a much slower rate of change for the image as C is incrementally varied. Reducing C lowers the visible effect of outliers on the data, which can be observed when comparable values are examined across tables, while higher C implies the model has greater complexity, and thus computation time increases. Conversely, larger values of gamma lead to much better results for relatively small values of C. We can observe dramatic changes in output for very small adjustments made to C.



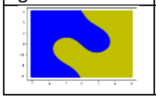
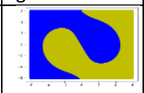
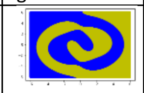
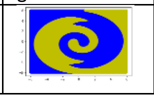
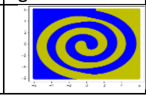
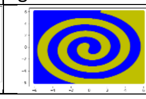
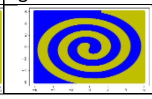
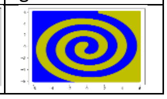
While keeping both parameters relatively low, an appropriate solution of high quality to choose from those shown above would be that displayed to the right, with C = 10 and gamma = 0.5.

This following approach for part b involved the five datasets of varying point densities used in 1.b) We modified our approach from part a above by generally focusing on a higher value of gamma, 0.5, as C only had to be increased a relatively small amount to produce noticeably improved images. Thus complexity was somewhat minimised, and computation time lessened. Any increase in C corresponds to a reduction in the level of bias, but increased the variance of observations.

Given these considerations, a specific set of C and gamma parameter pairings were allocated for testing in this experiment. However, some variation in the values chosen was provided as required to show the distinct critical ranges where the output approached an optimal 2-spiral result.

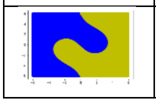
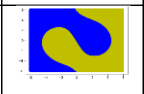
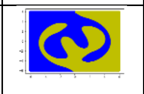
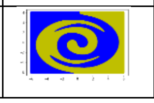
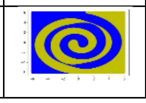
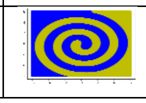
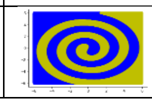
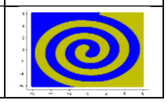
Density = 3

Being the most data-dense set of points, this model generates accurate models with the least amount of change to parameters. The 2-spiral appears as expected when C=2, gamma = 0.5.

C = 0.1, gamma = 0.1	C = 1, gamma = 0.1	C = 1, gamma = 0.3	C = 1, gamma = 0.5	C = 1.5, gamma = 0.5	C = 2, gamma = 0.5	C = 5, gamma = 0.5	C = 50, gamma = 0.5
							

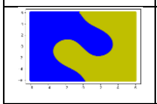
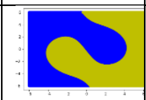
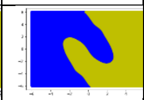
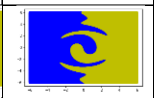
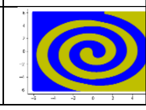
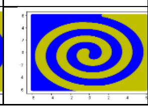
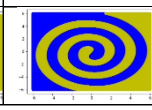
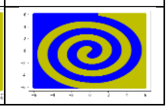
Density = 2

The 2-spiral image is best achieved at C=5, gamma = 0.5, which is somewhat later than in the previous iteration.

C = 0.1, gamma = 0.1	C = 1, gamma = 0.1	C = 1, gamma = 0.3	C = 1, gamma = 0.5	C = 2, gamma = 0.5	C = 5, gamma = 0.5	C = 10, gamma = 0.5	C = 50, gamma = 0.5
							

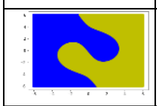
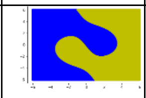
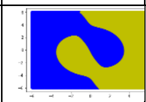
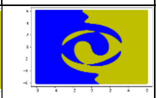
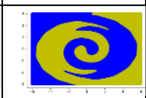
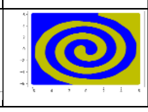
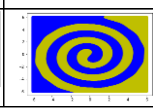
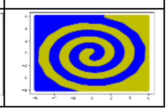
Density = 1

This is the original dataset, so many of these images were also generated for part a of the SVM question. We can see that the 2-spiral reaches an acceptable level of clarity at C=10, gamma = 0.5.

C = 0.1, gamma = 0.1	C = 1, gamma = 0.1	C = 1, gamma = 0.3	C = 1, gamma = 0.5	C = 5, gamma = 0.5	C = 10, gamma = 0.5	C = 50, gamma = 0.5	C = 50, gamma = 0.75
							

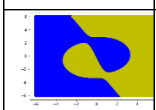
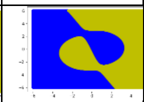
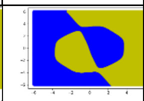
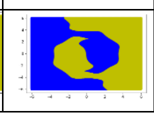
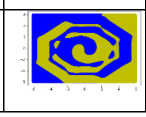
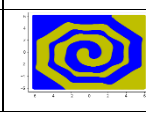
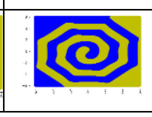
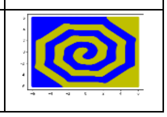
Density = 0.5

The data points being more sparse leads to a loss of fine-grain definition in the curves of the spiral images to follow, with the best plot appearing with parameters close to C = 50, gamma = 0.5.

C = 0.1, gamma = 0.1	C = 1, gamma = 0.1	C = 1, gamma = 0.3	C = 1, gamma = 0.5	C = 5, gamma = 0.5	C = 10, gamma = 0.5	C = 50, gamma = 0.5	C = 50, gamma = 0.75
							

Density = 0.25

Having further reduced the density of the points means that a perfectly round 2-spiral can no longer be generated; a well-defined octagonal shape appears instead, at approximately C = 50, gamma = 0.5. Although rigid, this appears to be the best possible image for the given points.

C = 0.1, gamma = 0.1	C = 1, gamma = 0.1	C = 1, gamma = 0.3	C = 1, gamma = 0.5	C = 5, gamma = 0.5	C = 10, gamma = 0.5	C = 50, gamma = 0.5	C = 50, gamma = 0.75
							

Observably, as the density of the data points provided to the model is reduced, a higher C value is required to produce results of comparable accuracy for the same values of gamma. The dataset with density 0.5 (containing 98 data points) results in an acceptable 2-spiral with appearance comparable to the best results we produced for any number of points for an ANN.

In general, it was found that Support Vector Machines performed better than Artificial Neural Networks for this task. They produced much smoother images, which was as expected given the non-linear nature of the problem. This can be attributed to the kernel tricks inherent in the process, and the maximum marginal separation anticipated when solving such problems. There are also fewer key parameters to vary, and the output is more predictable. The converse of this, of course, is that ANNs offer far more versatility in their complexity and are able to be applied to a much wider variety of problems.

Additionally, it can be seen above that SVMs produced better output sooner (with regards to the parameters being increased across each table) at each level of density.

References

[1] S. K. Chalup and L. Wiklendt, "Variations of the two-spiral task," *Connection Science,* vol. 19, no. 2, pp. 183-199, 2007.

[2] scikit-learn, "RBF SVM Parameters," [Online]. Available: https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html.