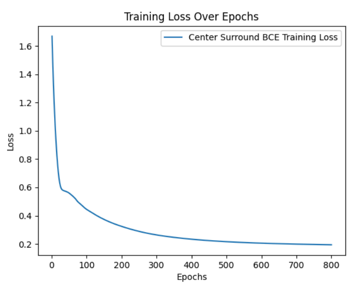
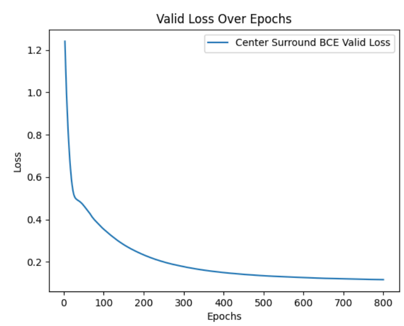
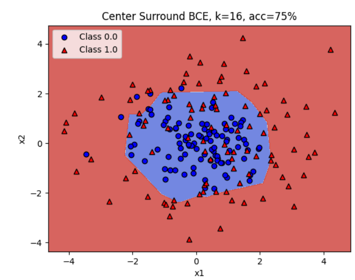
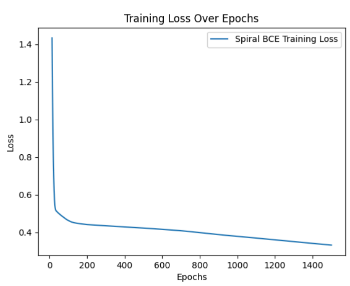
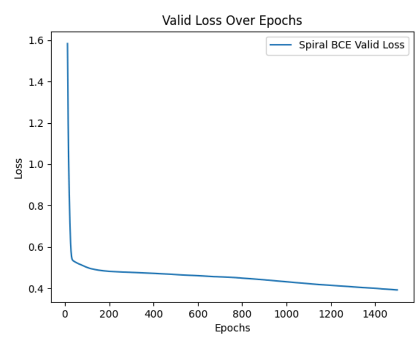
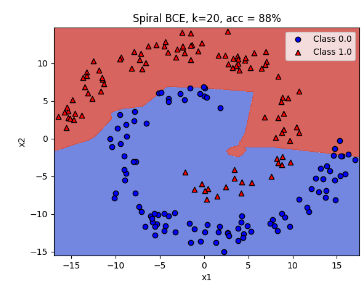
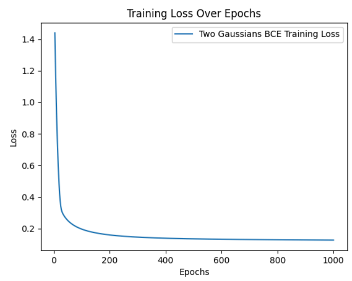
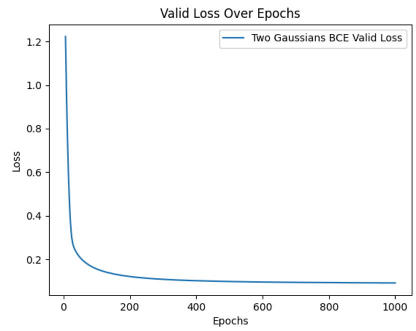
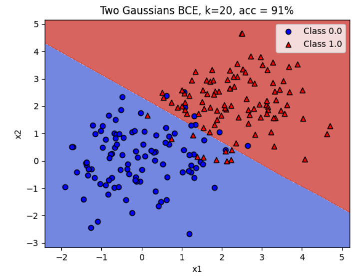
CS 449: Deep Learning
Homework 1

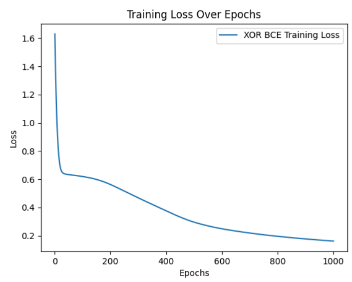
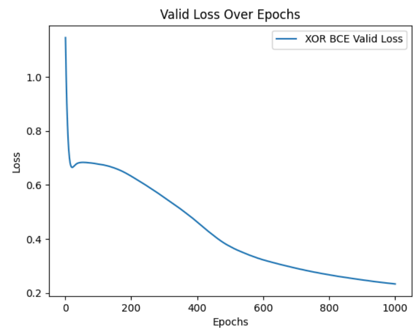
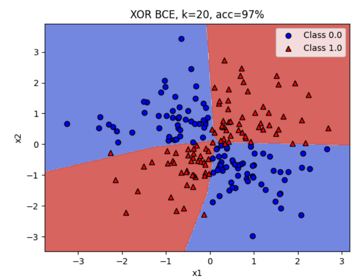
Group 14: Akash Vikram Shroff, Alexis Diaz-Waterman, Daniel Lee, Matthew Britt-Webb

1. **Binary Cross-Entropy MLPs**

   Our hyperparameters for each of the models were k (number of nodes in the hidden layer), learning rate, epochs, and batch size. For all the models, we used batch gradient descent owing to the small size of the training set. The other hyperparameters were decided after grid searching against the validation set.

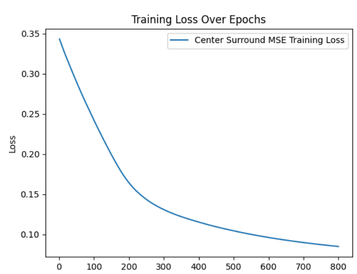
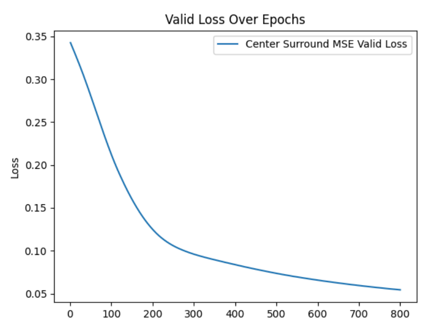
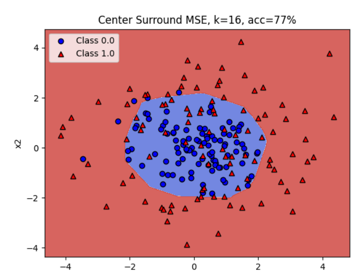
	Training Learning Curve	Validation Learning Curve	Decision Surface
Center Surround			 k = 16, accuracy = 75%
Spiral			 k = 20, accuracy = 88%

<table>
<tr><td>Two Gaussians</td><td>
</td><td>
</td><td>

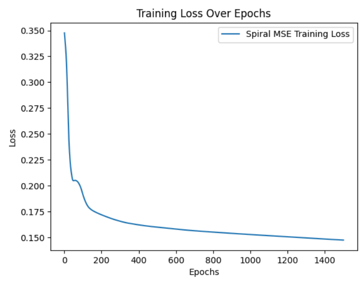
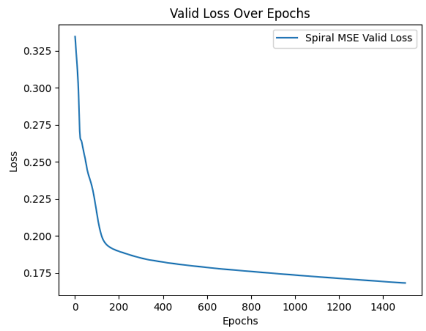
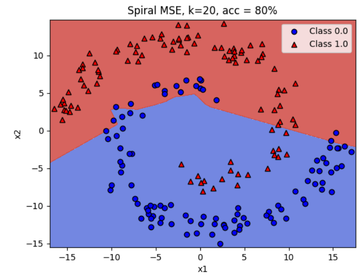
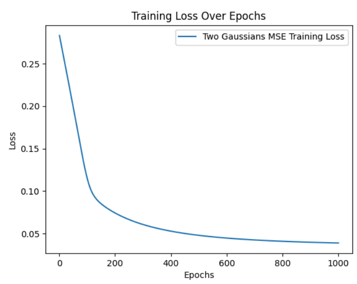
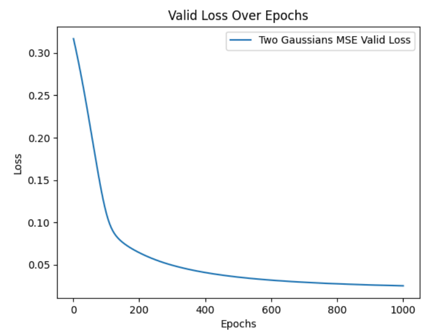
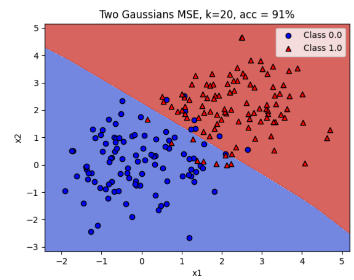
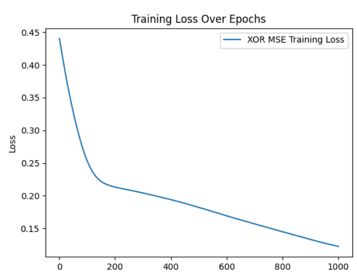
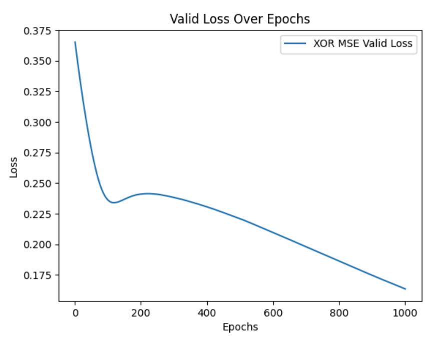
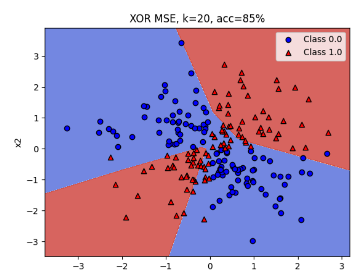
k = 20, accuracy = 91%</td></tr>
<tr><td>XOR</td><td>
</td><td>
</td><td>

k = 20, accuracy = 97%</td></tr>
</table>

2. Mean Squared Error MLPs

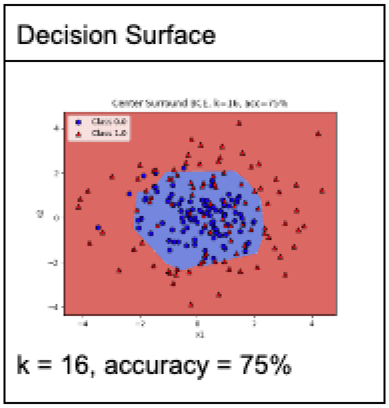
<table>
<tr><th></th><th>Training Learning Curve</th><th>Validation Learning Curve</th><th>Decision Surface</th></tr>
<tr><td>Center Surround</td><td>
</td><td>
</td><td>

k = 16, accuracy = 77%</td></tr>
</table>

Spiral			 k = 20, accuracy = 80%
Two Gaussians			 k = 20, accuracy 91%
XOR			 k = 20, accuracy 85%

3. **Select the worst-performing model(dataset, cost function and number of hidden nodes) from the above experiments and plot decision boundaries learned for each node in your hidden layer. Discuss why you selected this instance and speculate about factors contributing to poor performance.**

BCE, Center Surround with k=16 as our number of hidden nodes



The accuracy of this MLP is roughly 15% lower than the performance of the MLPs on other datasets. Grid searching for k revealed this to be close to the maximum accuracy that we could attain with an MLP.

The main factors contributing to the poor performance can be seen due to how close the points are to each other in the test set in the central area. As such, this makes it so that one hidden layer may not be enough to learn all the abstract relations between features and labels. As you can see in the image above, the blue and red (denoting classes 0 and 1 respectively) points overlap in the central region where the decision boundary has demarcated blue. Our simple MLP does correctly identify that the central region is more blue and the surrounding is red, however the central region has some disparity with overlapping red points which our model struggled with.

4. **(1.0points) Discuss how you might encourage your model to learn "feature maps" that could improve the performance of the instance selected for Step 3.**

   We could encourage our model to learn "feature maps" in a few ways. We could expand network architecture, meaning that we can try to introduce more hidden layers and increase the width of our model as well. This would allow us to learn more complex patterns in the data, however, would come with the increased potential of overfitting. To that end, we could employ some regularization techniques like dropout or weight regularization to counter-balance the potential for overfitting. Moreover, we could try to achieve something similar by using an ensemble of MLP models or feature engineering through AutoEncoders. Perhaps another avenue would be to use an architecture like a CNN and a moving filter to recognize underlying features in our data.