[EE538] Homework 4

Quang Minh Nguyen

Thursday 15th April, 2021

1. The generated data is in Figure 1. See the code in `main.py`.

2. The competitively learned clusters are presented in Figure 2. See `competitive.py` and `main.py`.

3. I don't really get what the problem means. One way of thinking about it is to treat the doughnut distributions as areas and the task is to find a linear classifier which minimize the misclassified area. This might involve many different cases of reasoning on the plane. Practically speaking, develop a program to approximate this minimizer is more feasible.

4. The decision boundaries with different learning rates are shown in Figure 3. We choose 0.3 as the appropriate learning rate. See `perceptron.py` and `main.py`.

5. The decision boundaries corresponding to different initializations are shown in Figure 4. See `perceptron.py` and `main.py`.

6. The computer program is distributed through out the files `activation_layer.py`, `activations.py`, `competitive.py`, `fclayer.py`, `layer.py`, `losses.py`, and `main.py`. My main coding reference was [1]. My significant contribution is introducing new activation functions and visualization functionality to the code.

7. Different learning rates are tried out as in Figure 5. We pick the learning rate 0.002 and experiment with different initializations in Figure 6.

8. Different learning rates are tried out as in Figure 7. We pick the learning rate 0.002 and experiment with different initializations in Figure 8. Note that we use sigmoid at the output neuron since only then can we synchronize the range of output with the range of desired output, which we can scale as $[0, 1]$. The same is for the upcoming problems and 11 and 12, where we choose to use ReLU.

   A notable feature of the ReLU nonlinearity as well as its derivative is that both are easy to compute and cost less time when placed in a network compared to sigmoid. Experimentally, we can see from Figures 6 and Figure 8 that while a 1 layer, 2 neuron network with sigmoid tends towards a linear decision boundary, the ReLU counterpart has a more sophisticated separation. At latter epochs, both exhibit a growing trend of cost function with high learning rates (≥ 0.01).

9. The five different initializations are tried out as in Figure 9. We pick the learning rate 0.05.

10. The five different initializations are tried out as in Figure 10.We pick the learning rate 0.08

11. The five different initializations are tried out as in Figure 11. We pick the learning rate 0.002.

12. The five different initializations are tried out as in Figure 12. We pick the learning rate 0.002.

Note that due to lack of computational resources, we could not experiment with sufficiently many settings of learning rates, initializations, or algorithm. For example, we prefix the maximum number of epochs at 100 for multilayer perceptrons, although it is likely that new learning patterns may emerge after this checkpoint.

Furthermore, the properties of this very specific problem are not simply scaled into different problems. Therefore, we are in no position to comment on these properties, in general.

References

[1] Aflak Omar. *Neural Network from scratch in Python*. https://web.archive.org/web/20201111191119if_/https://towardsdatascience.com/math-neural-network-from-scratch-in-python-d6da9f29ce65?gi=76adeb707de3. Accessed: 2020-04-16.

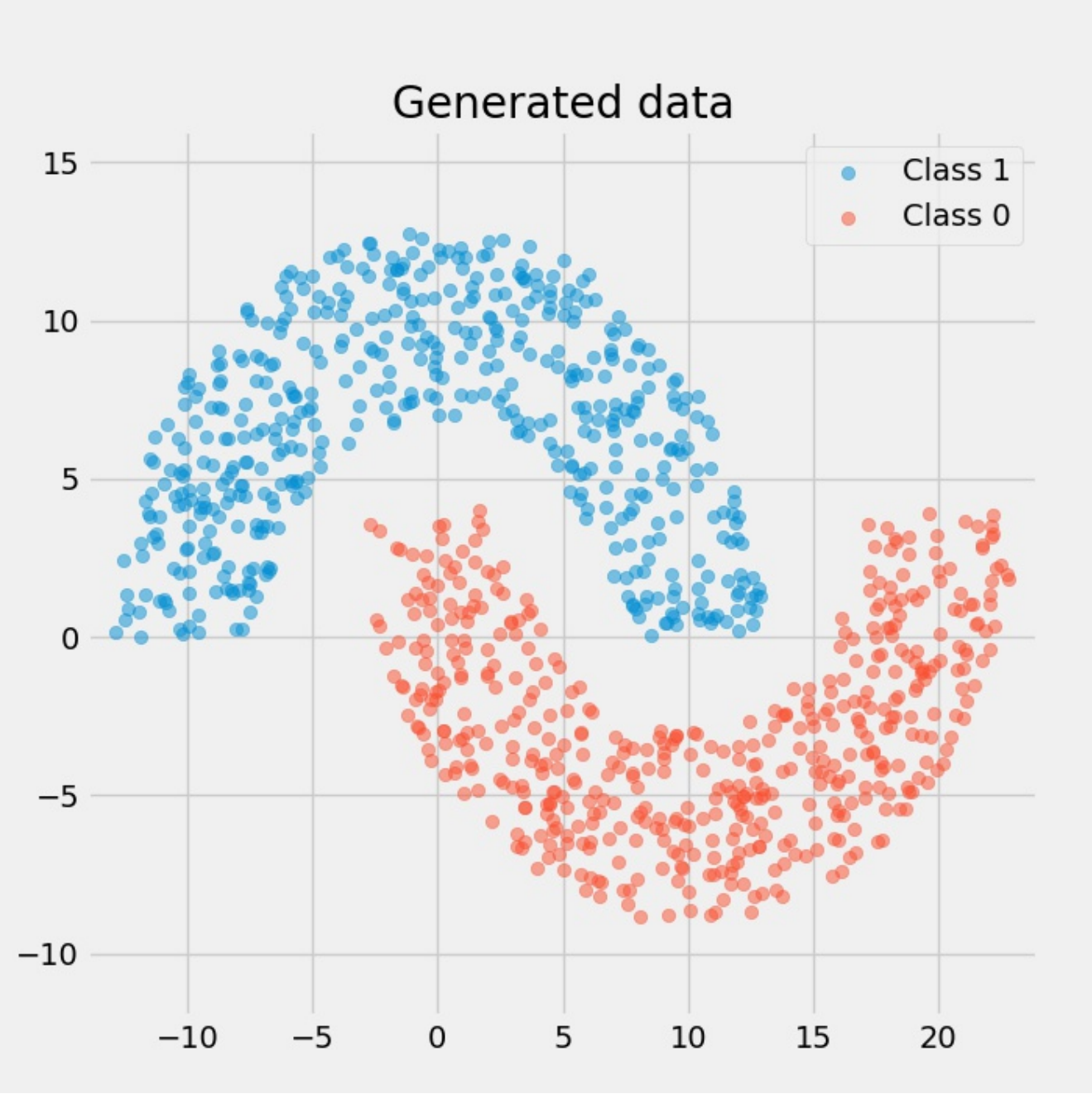

Figure 1: Problem 1—Generated data

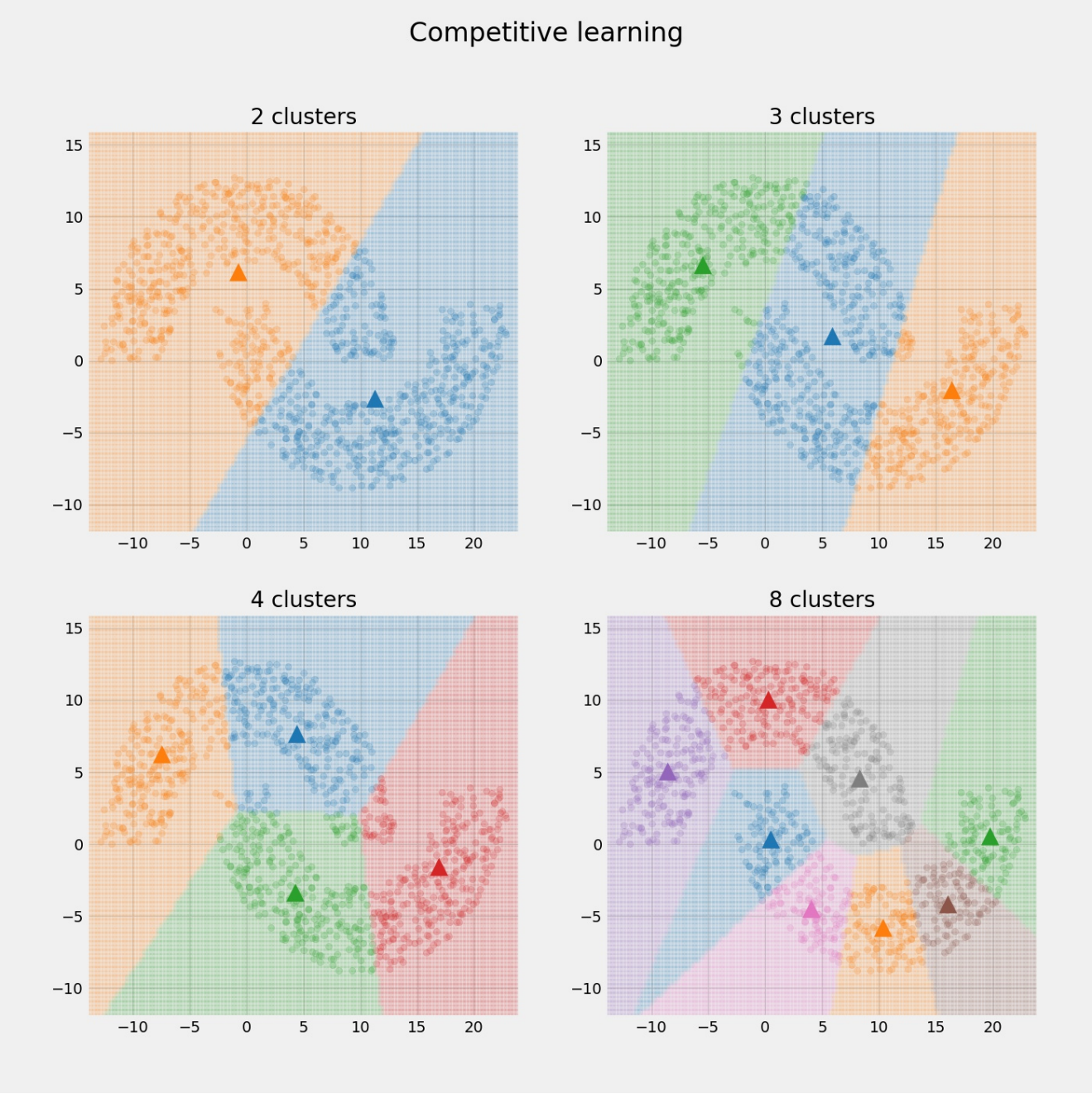

Figure 2: Problem 2—Competitively learned clusters

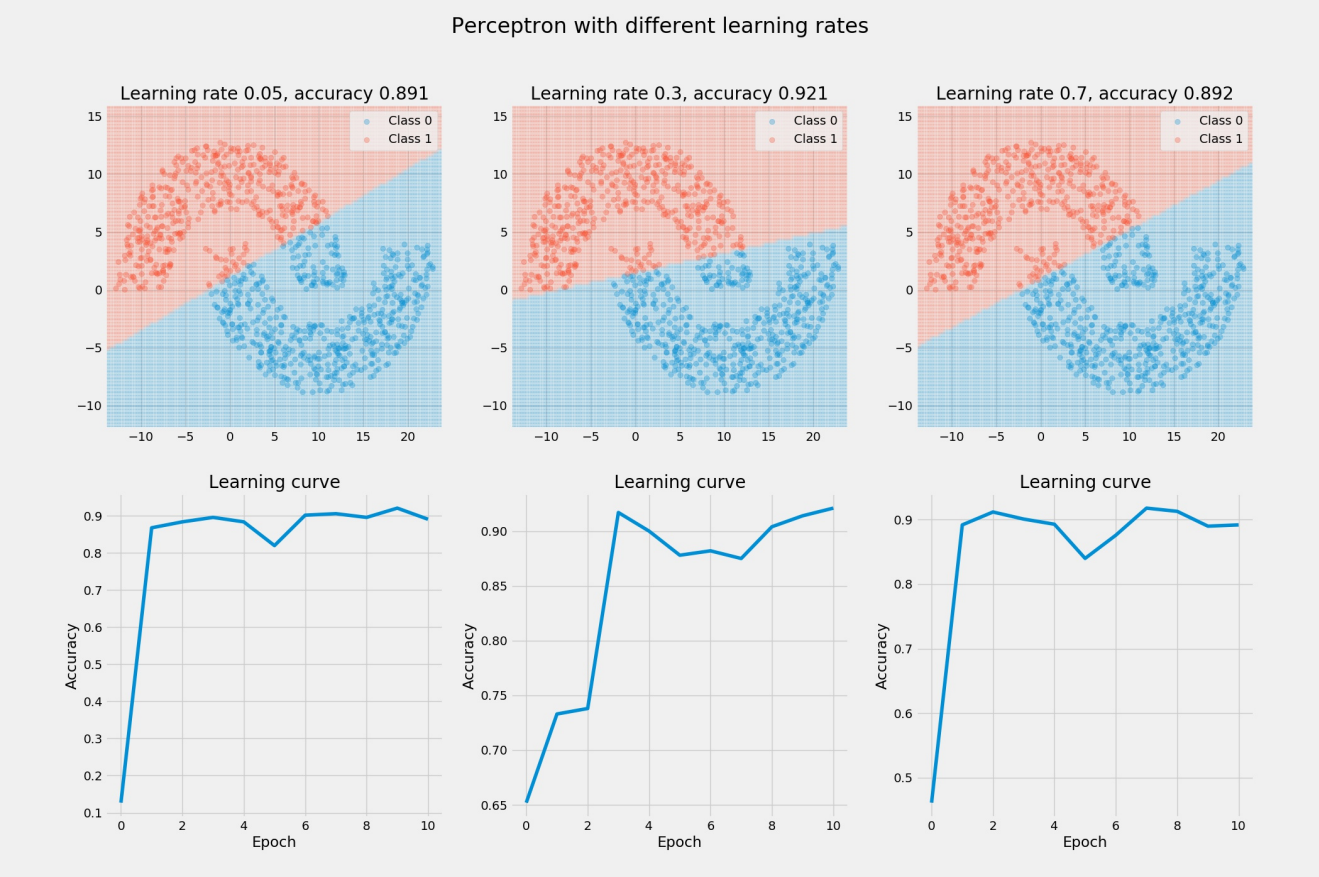

Figure 3: Problem 4—Perceptron with different learning rates

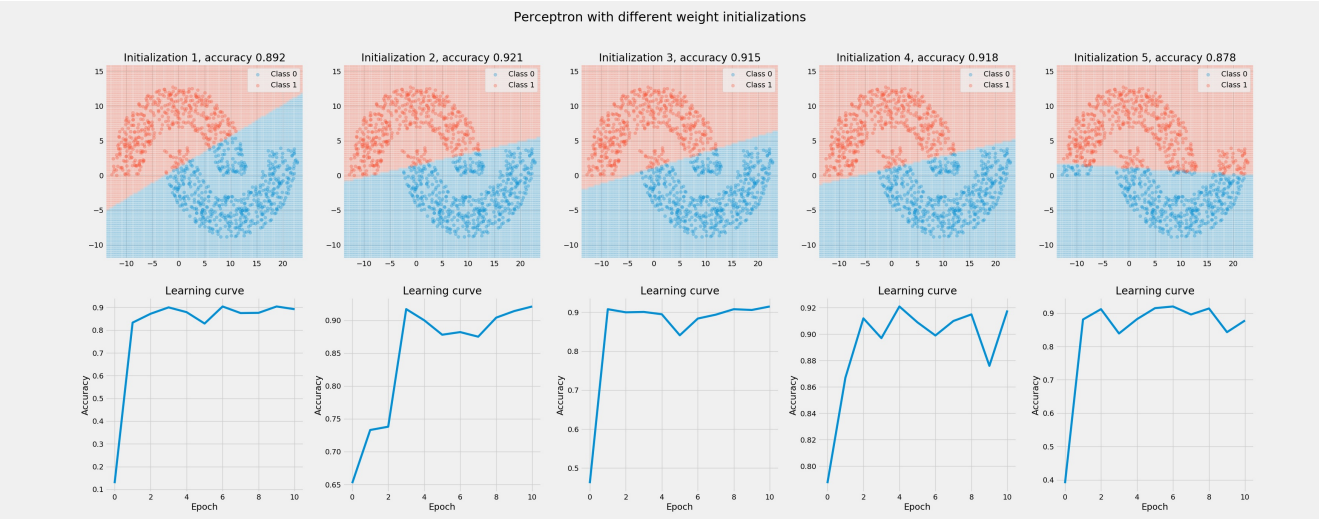

Figure 4: Problem 5—Perceptron with different initilizations

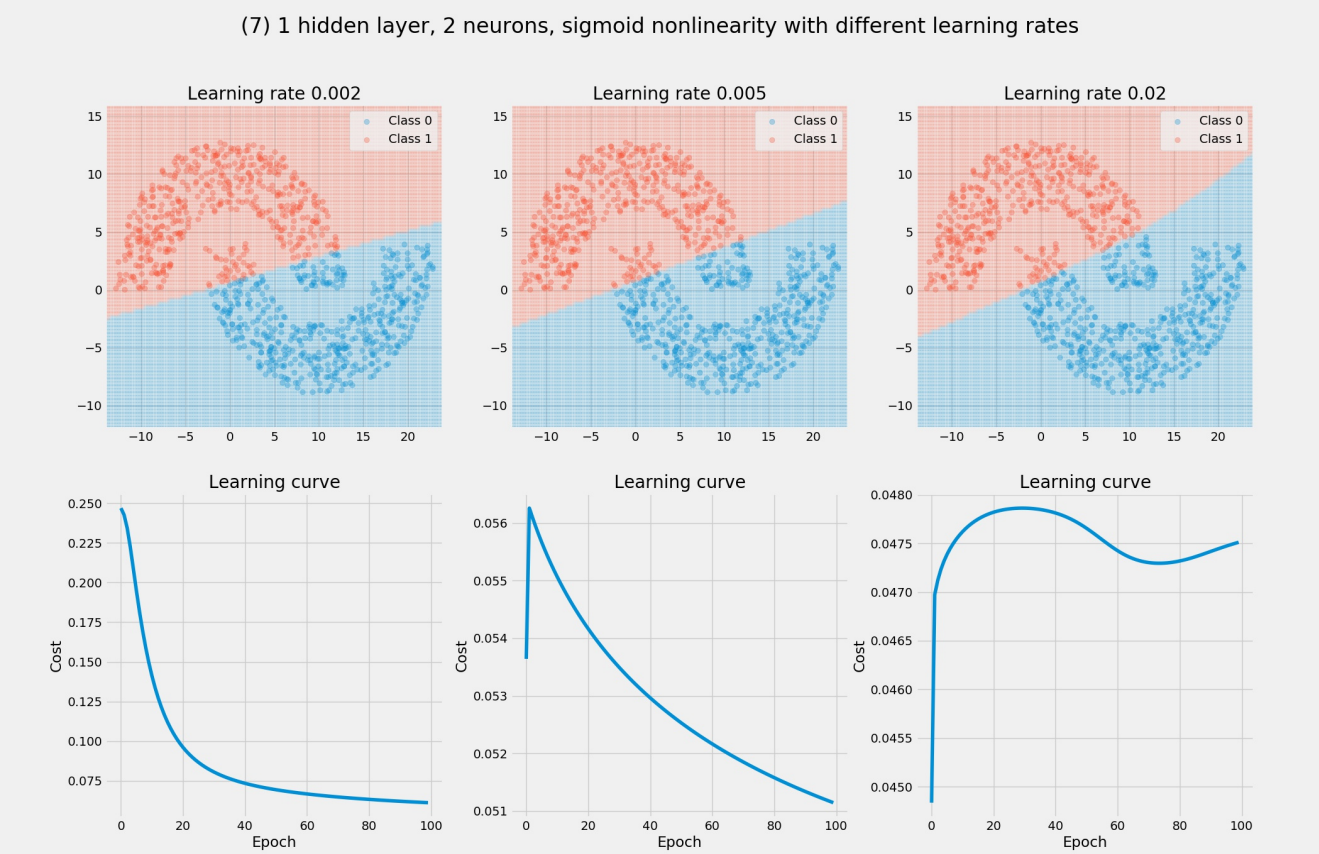

Figure 5: Problem 7—Different learning rates

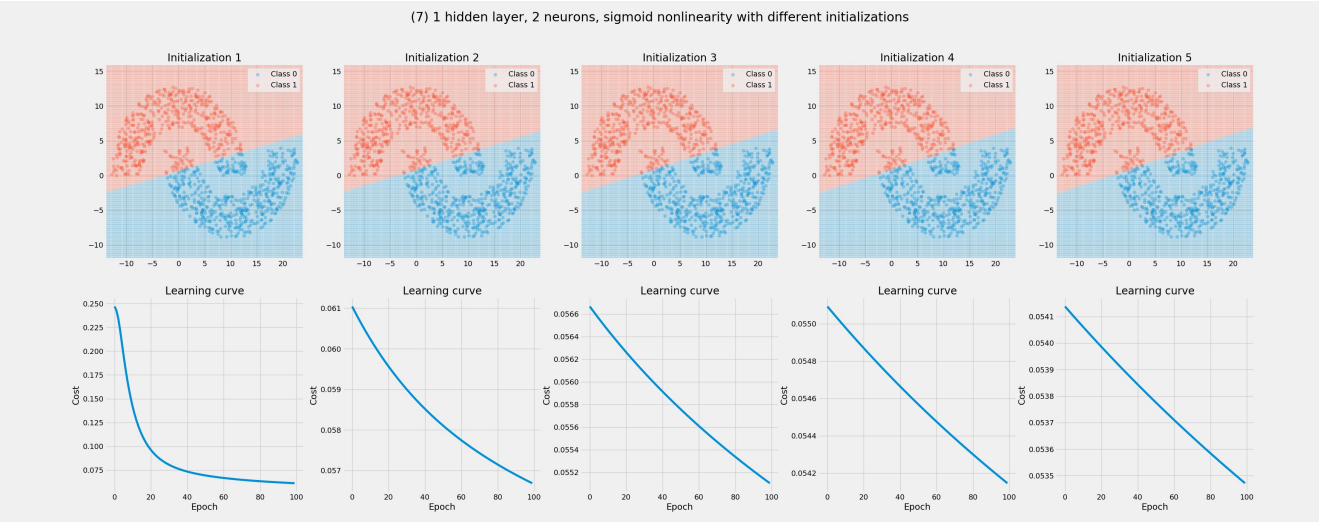

Figure 6: Problem 7—Different initializations

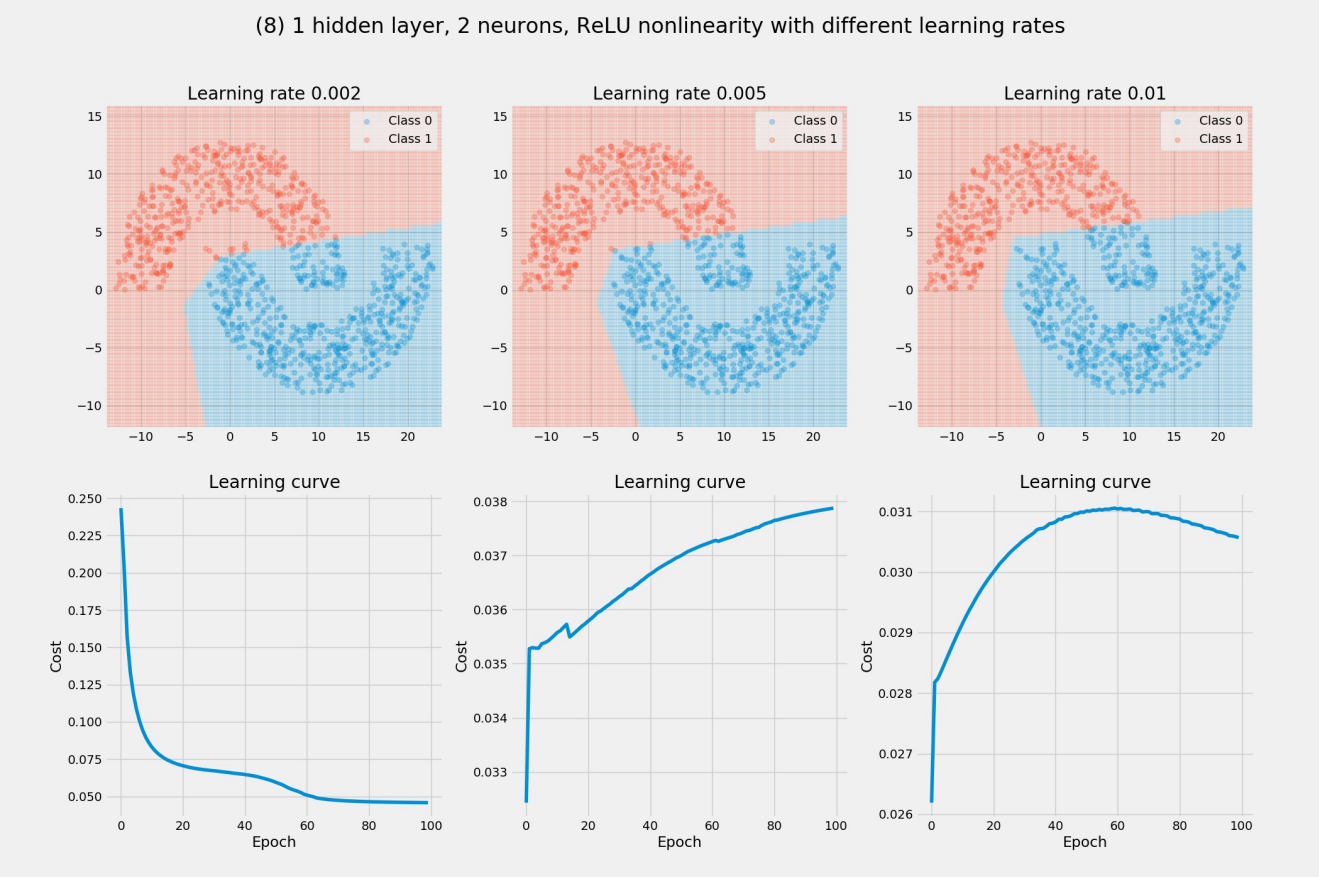

Figure 7: Problem 8—Different learning rates

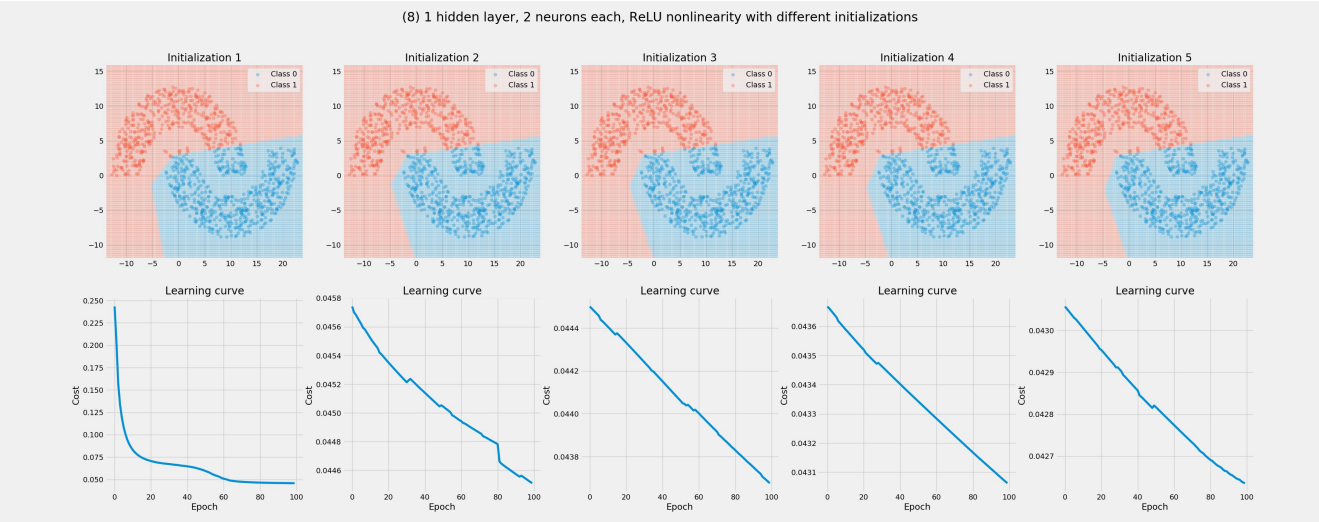

Figure 8: Problem 8—Different initializations

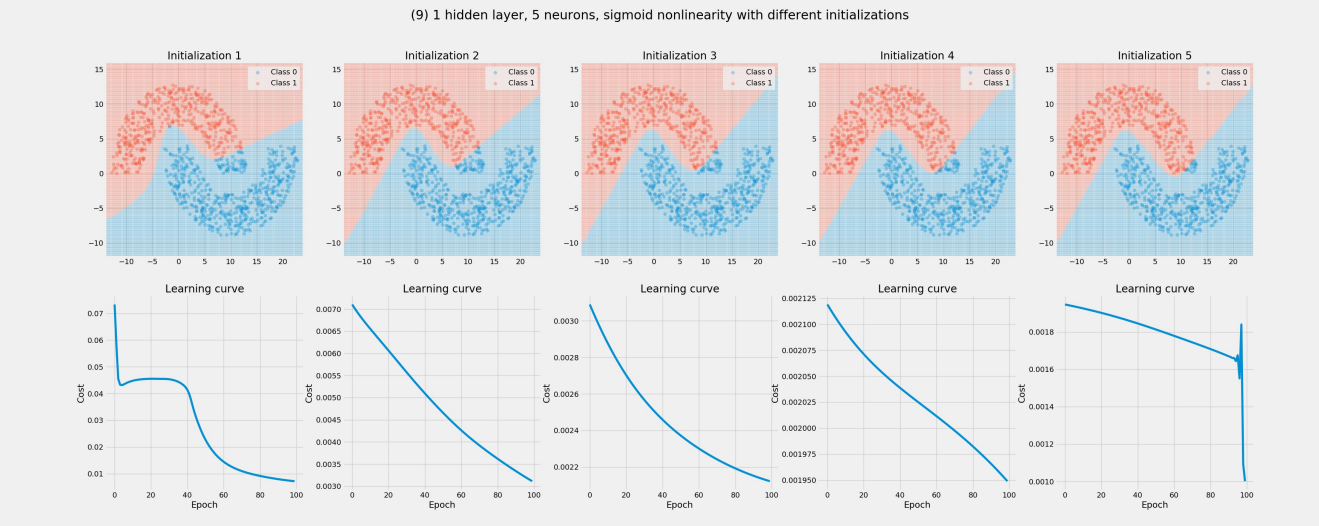

Figure 9: Problem 9

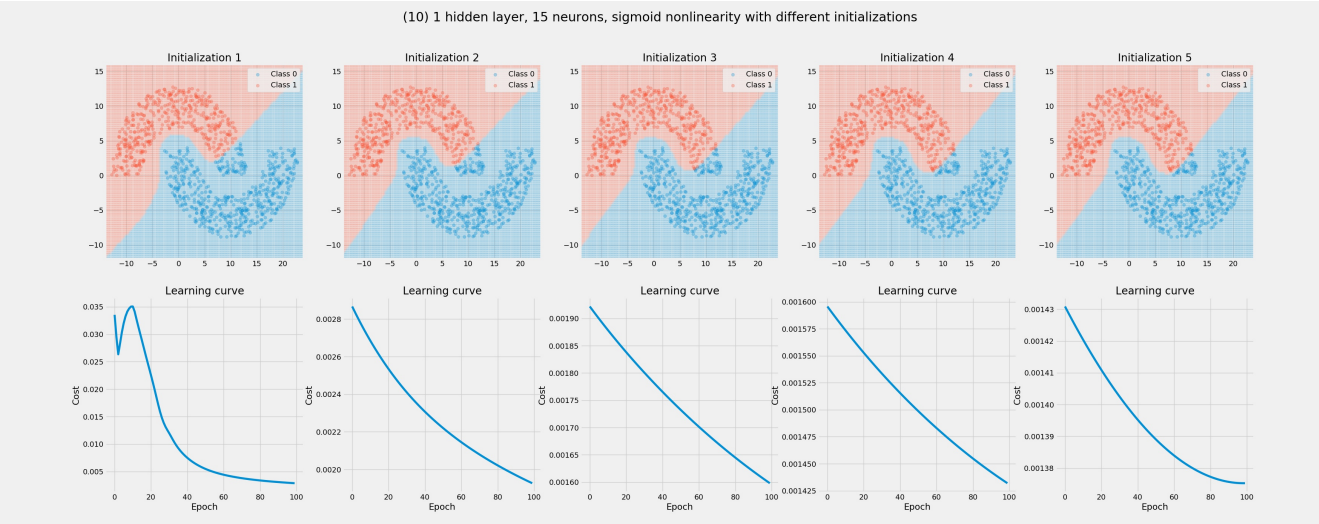

Figure 10: Problem 10

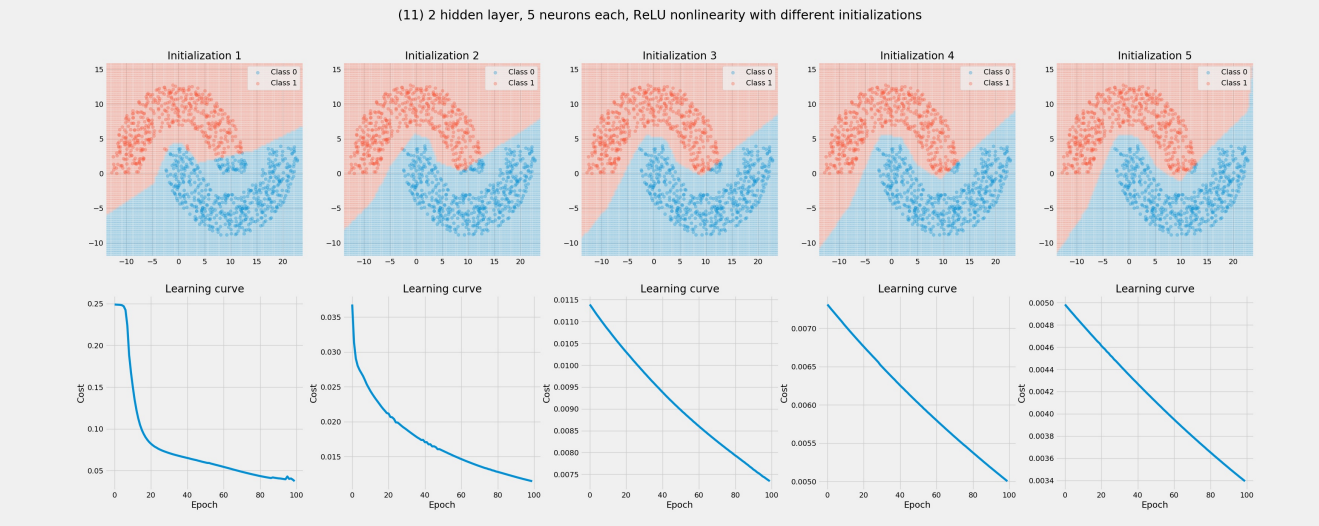

Figure 11: Problem 11

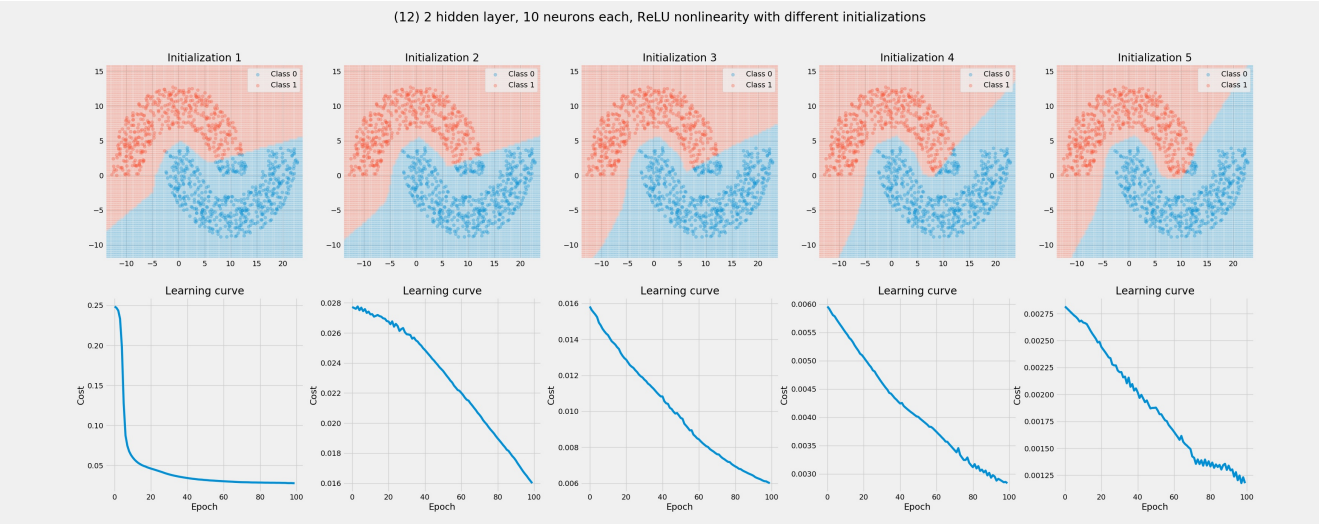

Figure 12: Problem 12