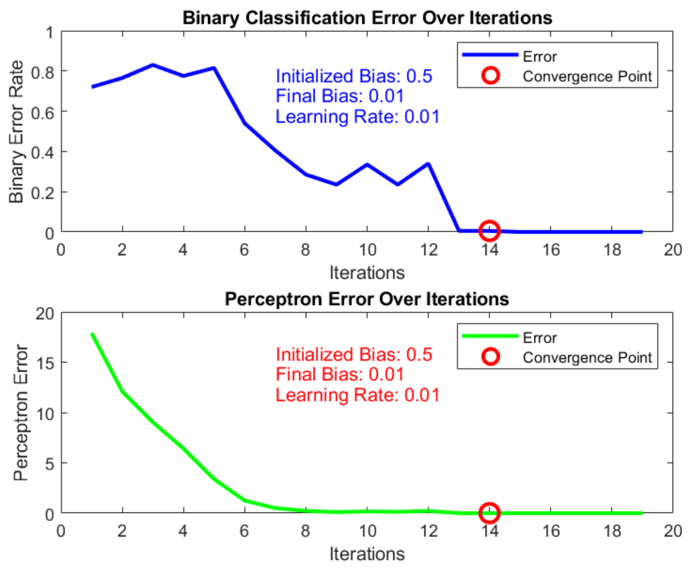
Homework 2

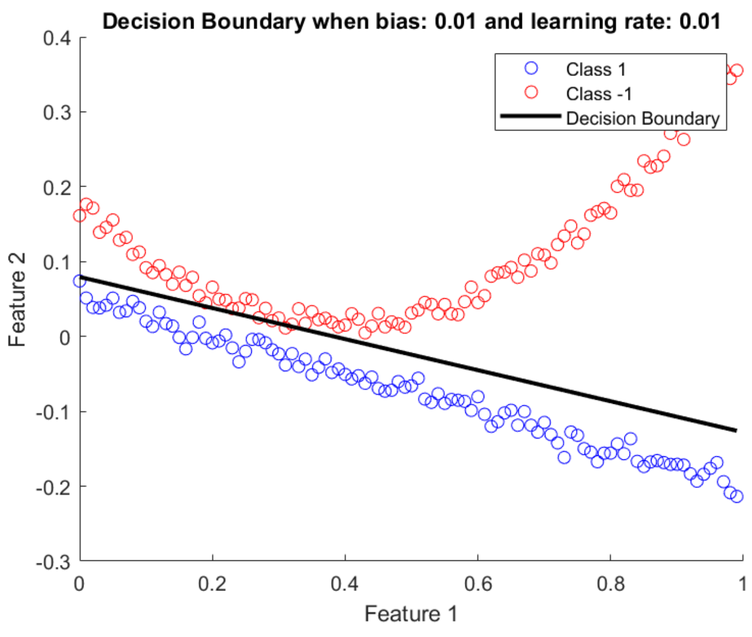
Komal Niraula (N16417290)

kn2505@nyu.edu

Problem 1

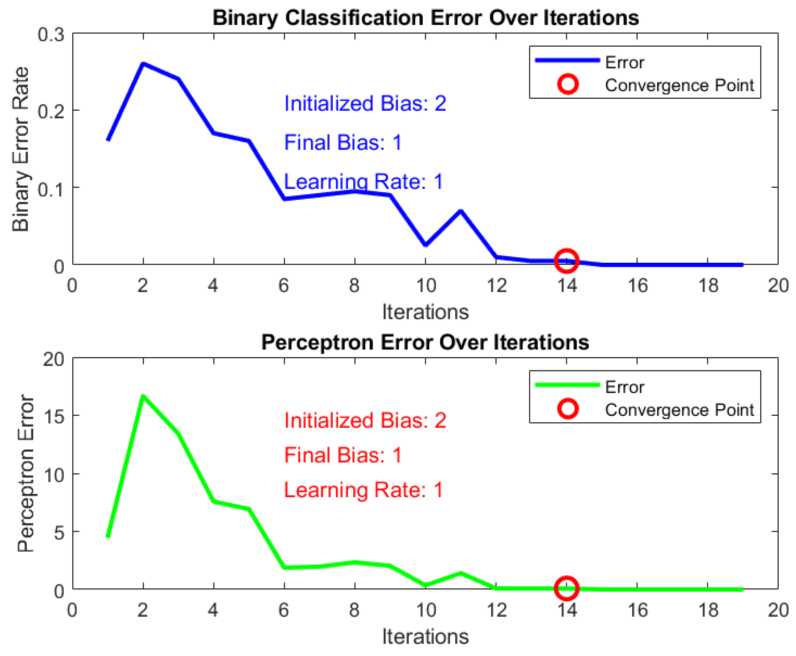
When bias was initialized at 0.5 and the learning rate was kept at 0.01, the bias after converging (learned bias) was 0.01. The graph converged at the 14th iteration.

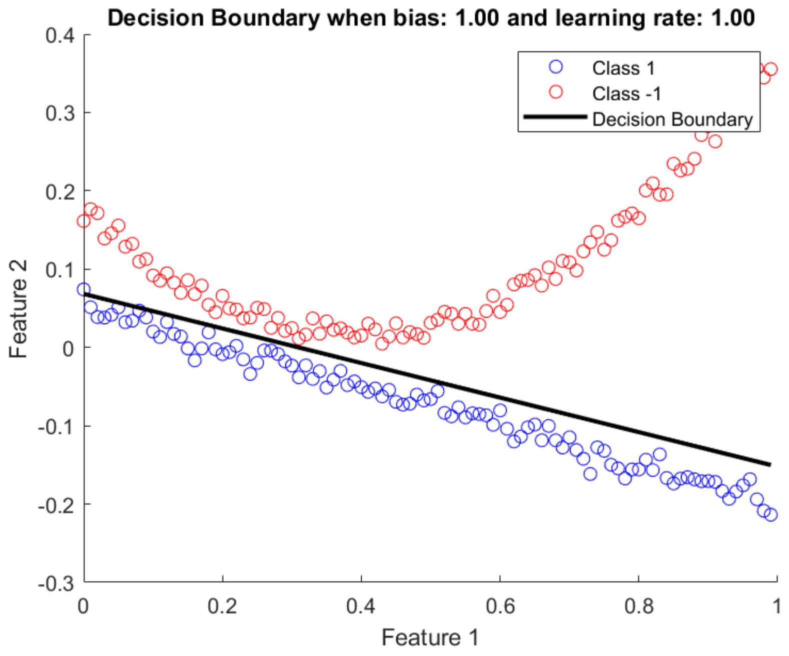




The decision boundary doesn't seem to be perfectly separating the classes. It is overlapping some of the red data points (Class -1).

Increasing the initial bias to 2 and making the learning rate 1, the bias after converging (learned bias) was 1. The graph converged at the 14th iteration.

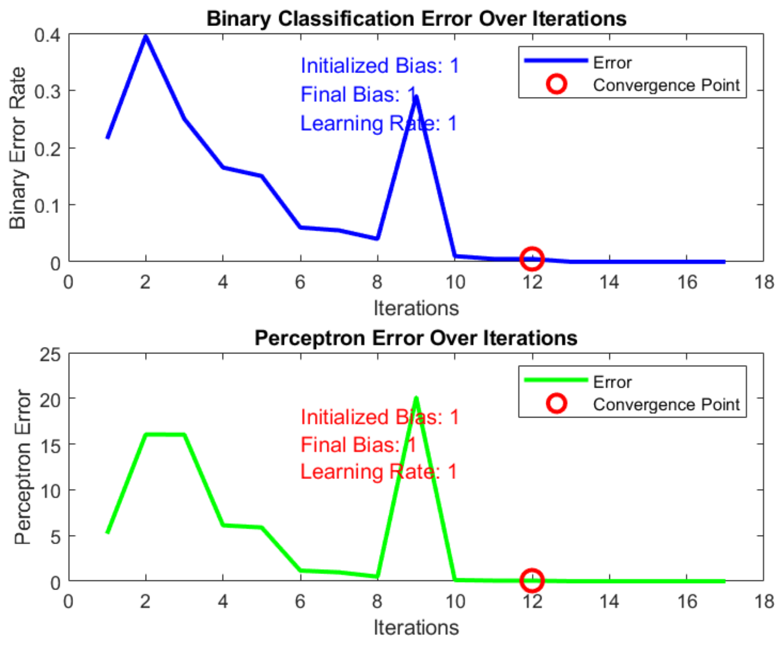


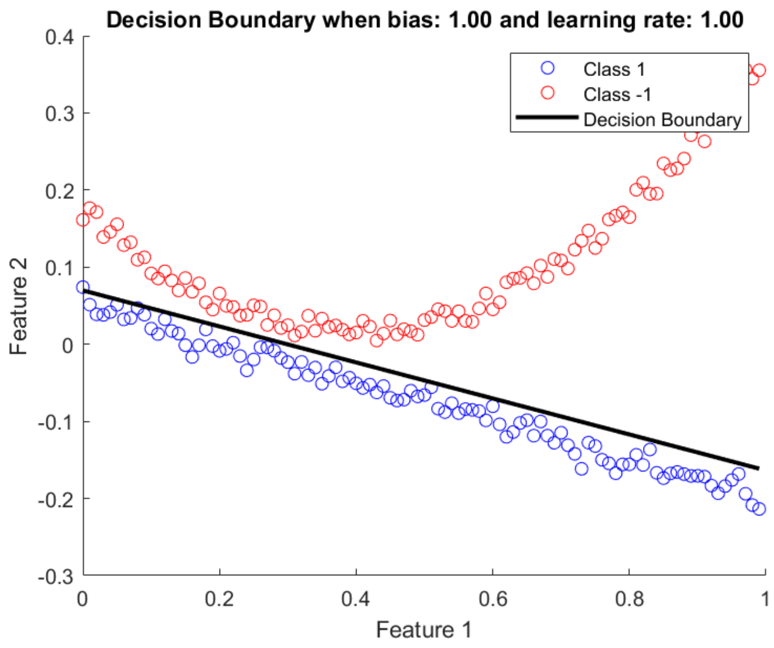


The problem of decision boundary overlapping red data points have been solved but now it's touching some of the blue data points (Class 1).

Let's take this final bias as an initialized bias for the next experiment.

When bias was initialized at 1 and the learning rate was kept at 1, the bias after converging (learned bias) was also 1. The graph converged at the 12th iteration.

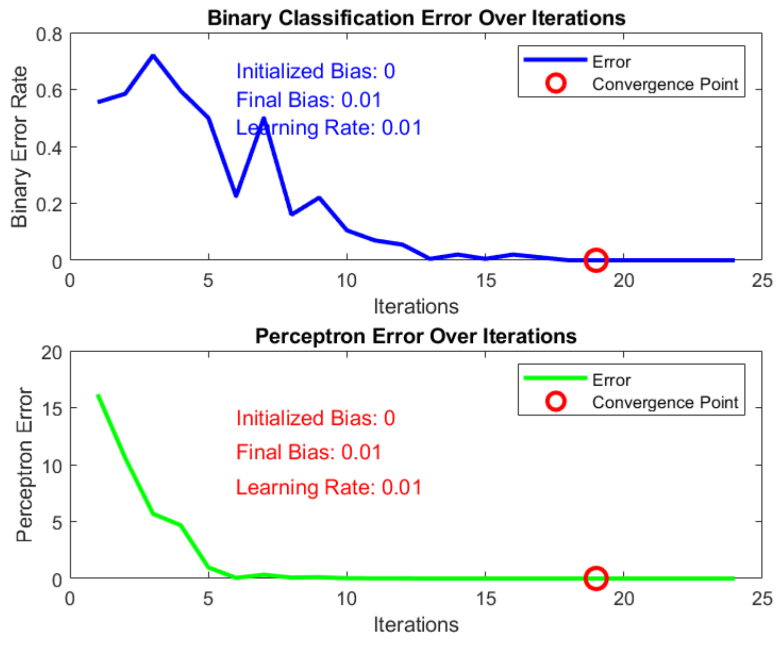


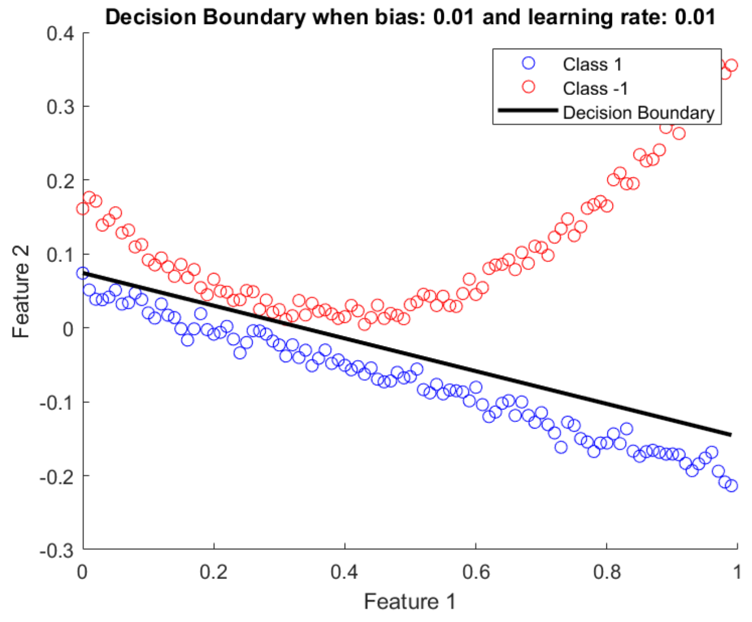


The decision boundary seems to be a little better than the previous, however, it’s still slightly touching the blue data points.

Finally, let's initialize the bias at 0 and keep the learning rate small (0.01)

When bias was initialized at 0 and the learning rate was kept at 0.01, the bias after converging (learned bias) was 0.01. The graph converged at the 19th iteration.





In this experiment, the lowest convergence was on 12th iteration when bias was initialized at 1 and learning rate was also 1. Surprisingly, in this scenario, the learned bias was also 1.

Overall, based on this experiment we can also conclude that perceptron error converges faster than the binary classification error.

Problem 2

a.

The error function (E) for a single training example is:

$$E = -\sum_i [t_i \log(x_i) + (1 - t_i)\log(1 - x_i)]$$

Here, $x_i = \frac{1}{1+e^{-s_i}}$ and $s_i = \sum_j y_j w_{ji}$

Computing the derivative of E with respect to the weights w_{ji}:

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial x_i} \cdot \frac{\partial x_i}{\partial s_i} \cdot \frac{\partial s_i}{\partial w_{ji}}$$

$$= \frac{\partial}{\partial x_i}\left(-\sum_i [t_i \log(x_i) + (1 - t_i)\log(1 - x_i)]\right)\frac{\partial x_i}{\partial s_i}\frac{\partial s_i}{\partial w_{ji}}$$

$$= \frac{\partial}{\partial x_i}(t_i \log(x_i) + (1 - t_i)\log(1 - x_i))\frac{\partial x_i}{\partial s_i}\frac{\partial s_i}{\partial w_{ji}}$$

$$= \left(\frac{1 - t_i}{1 - x_i} - \frac{t_i}{x_i}\right)\frac{\partial x_i}{\partial s_i}\frac{\partial s_i}{\partial w_{ji}}$$

$$= \left(\frac{1 - t_i}{1 - x_i} - \frac{t_i}{x_i}\right)\frac{\partial}{\partial s_i}\sigma(s_i)\frac{\partial s_i}{\partial w_{ji}}$$

$$= \left(\frac{1 - t_i}{1 - x_i} - \frac{t_i}{x_i}\right)\sigma'(s_i)\frac{\partial s_i}{\partial w_{ji}}$$

$$= \left(\frac{1 - t_i}{1 - x_i} - \frac{t_i}{x_i}\right)x_i(1 - x_i)\frac{\partial s_i}{\partial w_{ji}}$$ [Taking derivative of the sigmoid activation]

$$= \frac{(1 - t_i)x_i - t_i + t_i x_i}{(1 - x_i)x_i}x_i(1 - x_i)\frac{\partial s_i}{\partial w_{ji}}$$

$$= \frac{x_i - t_i}{(1 - x_i)x_i}x_i(1 - x_i)\frac{\partial s_i}{\partial w_{ji}}$$

$$= (x_i - t_i)\frac{\partial s_i}{\partial w_{ji}}$$

$$= (x_i - t_i)y_j$$

Backpropagation on weight w_{kj}:

$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial s_j}\frac{\partial s_j}{\partial w_{kj}}$$

$$\frac{\partial E}{\partial w_{kj}} = \sum_i \frac{\partial E}{\partial s_i}\frac{\partial s_i}{\partial s_j}\frac{\partial s_j}{\partial w_{kj}}$$

$$= \sum_i (x_i - t_i)\frac{\partial s_i}{\partial s_j}\frac{\partial s_j}{\partial w_{kj}}$$

$$= \sum_i (x_i - t_i)\frac{\partial}{\partial s_j}\left(\sum_j y_j w_{ji}\right)\frac{\partial s_j}{\partial w_{kj}}$$

$$= \sum_i (x_i - t_i)\frac{\partial}{\partial s_j}\left(\sigma(s_j) w_{ji}\right)\frac{\partial s_j}{\partial w_{kj}}$$

$$= \sum_i (x_i - t_i) w_{ji} \sigma'(s_j)\frac{\partial s_j}{\partial w_{kj}}$$

$$= \sum_i (x_i - t_i) w_{ji} y_j (1 - y_j)\frac{\partial s_j}{\partial w_{kj}}$$

$$= \sum_i (x_i - t_i) w_{ji} y_j (1 - y_j)\frac{\partial}{\partial w_{kj}}\sum_j w_{kj} z_k$$

$$= \sum_i (x_i - t_i) w_{ji} y_j (1 - y_j) z_k$$

b.

The error function is given as:

$$E = -\sum_i t_i \log(x_i)$$

Here, SoftMax activation function:

$$x_i = \frac{e^{s_i}}{\sum_{c=1}^{m} e^{s_c}}$$

Computing the derivative of E with respect to the weights w_{ji}:

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial x_i} \cdot \frac{\partial x_i}{\partial s_i} \cdot \frac{\partial s_i}{\partial w_{ji}}$$

$$= \frac{\partial}{\partial s_i}\left(-\sum_j t_j \log(x_j)\right) \cdot \frac{\partial x_i}{\partial s_i} \cdot \frac{\partial s_i}{\partial w_{ji}}$$

$$= -\frac{t_i}{x_i} \cdot \frac{\partial x_i}{\partial s_i} \cdot \frac{\partial s_i}{\partial w_{ji}}$$

$$= -\frac{t_i}{x_i}\frac{\partial}{\partial s_i}\sigma_2(s_i)\frac{\partial s_i}{\partial w_{ji}}$$

$$= -\frac{t_i}{x_i}\frac{\partial}{\partial s_i}\left(\frac{e^{s_i}}{\sum_k e^{s_k}}\right)\frac{\partial s_i}{\partial w_{ji}}$$

$$= \frac{t_i}{x_i}\left(\frac{e^{s_i}\sum_k e^{s_k} - e^{s_i}e^{s_i}}{(\sum_k e^{s_k})^2}\right)\frac{\partial s_i}{\partial w_{ji}}$$

$$= \frac{t_i}{x_i}\left(\frac{x_i(1-x_i)}{\sum_k e^{s_k}}\right)\frac{\partial s_i}{\partial w_{ji}}$$

$$= -t_i(x_i - x_i^2)\frac{\partial s_i}{\partial w_{ji}}$$

$$= -t_i(1-x_i)\frac{\partial}{\partial w_{ji}}\sum_j y_j w_{ji}$$

$$= -t_i(1-x_i)y_j$$

Backpropagation on weight w_{kj}:

$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial s_j}\frac{\partial s_j}{\partial w_{kj}}$$

$$= \sum_i \frac{\partial E}{\partial s_i} \frac{\partial s_i}{\partial s_j} \frac{\partial s_j}{\partial w_{kj}}$$

$$= \sum_i \frac{\partial s_j}{\partial w_{kj}} \big(-t_i(1 - x_i)\big)$$

$$= \sum_i y_j \big(-t_i(1 - x_i)\big)$$

$$= \sum_i y_j \big(1 - y_j\big)\big(-t_i(1 - x_i)\big)$$

$= \sum_i y_j\big(1 - y_j\big)\big(-t_i(1 - x_i)\big) \sum_j w_{kj} z_k$ [Further backpropagating the gradient]

$$= \sum_i w_{ji} y_j \big(1 - y_j\big)\big(-t_i(1 - x_i)\big) z_k$$

$$= -\sum_i t_i(1 - x_i) w_{ji} y_j \big(1 - y_j\big) z_k$$

Problem 3

The entropy function: $H = -\sum_{k=1}^{N} p_k \log p_k$

The Lagrangian function $\mathcal{L}$ is:

$$\mathcal{L} = -\sum_{k=1}^{N} p_k \log p_k + \lambda\left(\sum_{k=1}^{N} p_k - 1\right)$$

$$\mathcal{L} = -\sum_{k=1}^{N} p_k \log p_k + \lambda \sum_{k=1}^{N} p_k - \lambda$$

Taking derivative of $\mathcal{L}$ with respect to p$_k$: $\frac{\partial \mathcal{L}}{\partial p_k} = -(\log p_k + 1) + \lambda$

Making this 0 for maximization:

$$-(\log p_k + 1) + \lambda = 0$$

$$\log p_k = \lambda - 1$$

$$p_k = e^{\lambda - 1}$$

$\sum_{k=1}^{N} e^{\lambda-1} = 1$ [Applying normalization constant]

$$Ne^{\lambda-1} = 1$$

$$e^{\lambda-1} = \frac{1}{N}$$

$$p_k = \frac{1}{N}$$

Therefore, the probability distribution, that maximizes entropy is:

$$p_k = \frac{1}{N} \quad \text{for all } k = 1,2, \ldots, N$$

Problem 4

The VC dimension measures the complexity of a hypothesis, specifically how many points can be perfectly classified by the hypothesis class.

For instance, a single point can be easily separated by an axis-aligned square. The square can either include or exclude the point. Similarly, in case of 2 points, an axis-aligned square can either include one or both points or do not include any. Also, for three points, a square can include or exclude any combination of the points. However, in case of four points, not all subsets of the points can be realized by axis-aligned squares.

Consider where only two diagonal points need to be included. Let these two diagonal points be A and K in the below figure. There is no way for an axis-aligned square to only consider A and K while excluding L and N.

```
A . . . L
. . . . .
N . . . K
```

(Supposing this is a square with four corners: A, L, N, K)

So, In the case of axis-aligned squares, the VC dimension is 3. This is the maximum number of points that axis-aligned squares can shatter.