Assignment#4

- Part 1
  - Plot data from data.csv
  - Implement perceptron heuristic approach (right box in the next page)
  - Plot the initial separation line as red, subsequent ones after each iteration in dashed green, and the last one in black (see page 3)
  - Play with learning rate

- Part 2
  - Plot data from data.csv
  - Implement perceptron using Gradient Descent approach (left box in the next page)
  - Play with learning rate, number of epochs.
  - Plot the initial separation line as red, subsequent ones after each iteration in dashed green, and the last one in black
  - Compute log loss (error) and plot the error graph every 10 epoch (see page 4)

Learning by Gradient Descent (Left Side)

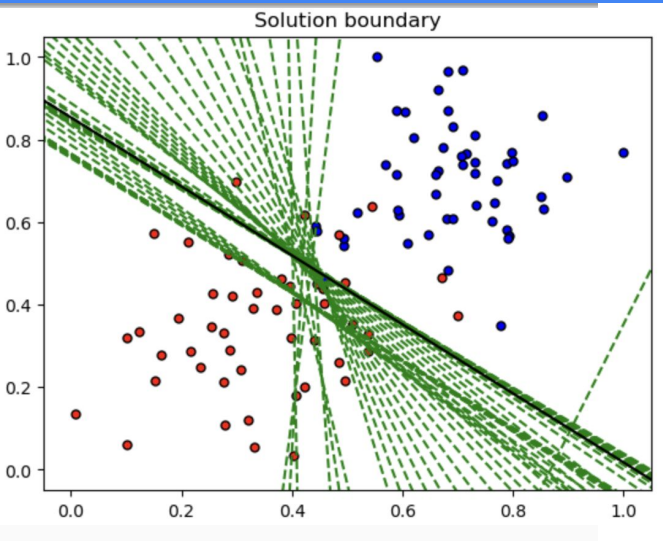
1. Start a perceptron with random weights and bias: $w_1, w_2, \ldots, w_n, b$
2. For all points (data) with their corresponding labels (answers):
   - 2.1. Compute prediction output ($\hat{y}$)
   - 2.2. Compute error function ($y - \hat{y}$)
   - 2.3. $b + r\mathbf{(y - \hat{y})} \rightarrow b$
   - 2.4. For all w_i: $w_i + r\mathbf{(y - \hat{y})}x_i \rightarrow w_i$
3. Repeat #2 until **error is small**

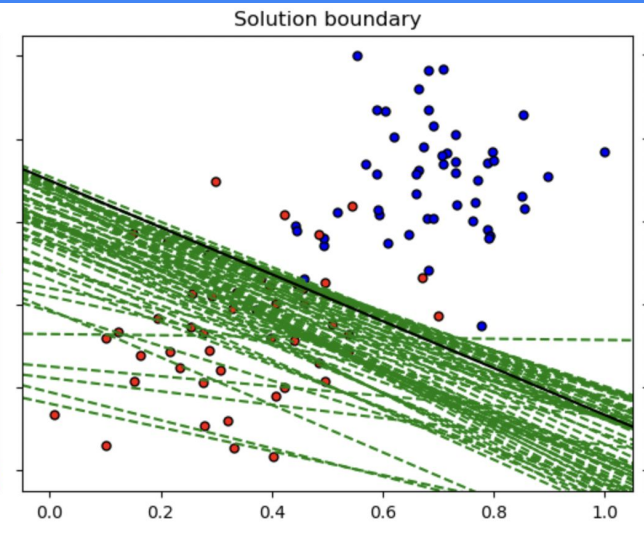
Note: $\hat{y}$ is no longer 0 or 1 from a step function

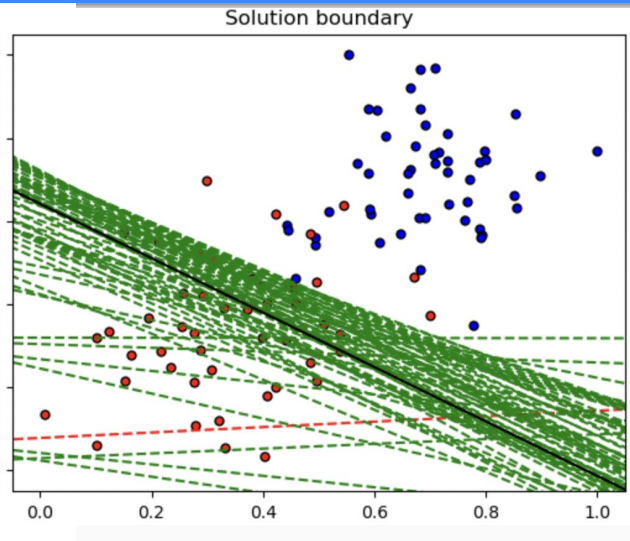
1. Start a perceptron with random weights and bias: $w_1, w_2, \ldots, w_n, b$
2. For all points (data) with their corresponding labels (answers):
   - 2.1. Classify according to the perceptron
   - 2.2. For a misclassified point $(x_1, x_2, \ldots, x_n)$:
     - 2.2.1. If classification==0:
       - 2.2.1.1. $b + r \rightarrow b$
       - 2.2.1.2. For all w_i: $w_i + rx_i \rightarrow w_i$
     - 2.2.2. If classification==1:
       - 2.2.2.1. $b - r \rightarrow b$
       - 2.2.2.2. For all w_i: $w_i - rx_i \rightarrow w_i$
3. Repeat #2 enough number of times

Earlier heuristic approach with binary classification

Learning rate (samples, your results may be different)







learning_rate = 0.01

iteration = 65

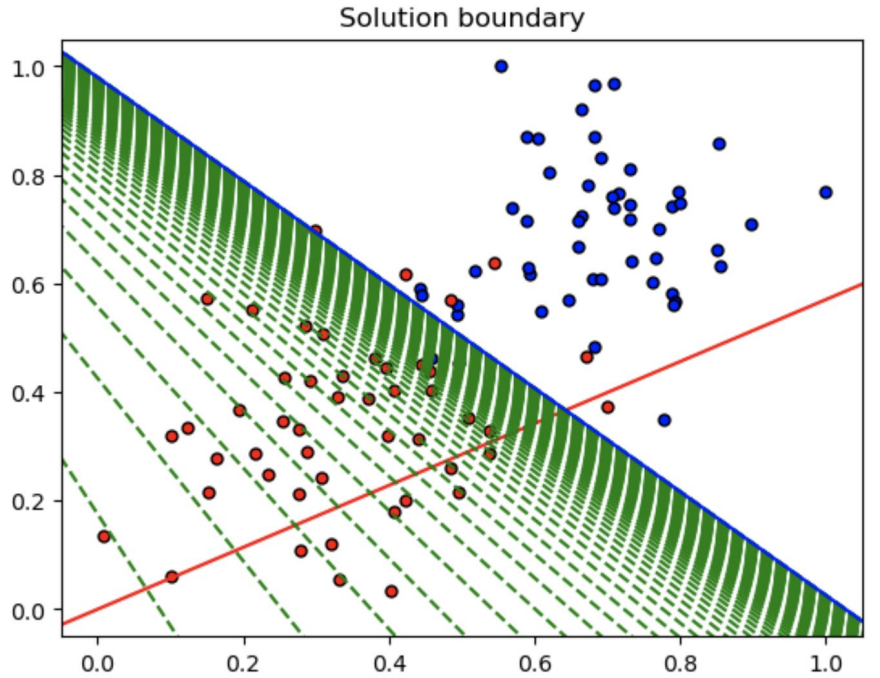
learning_rate = 0.1

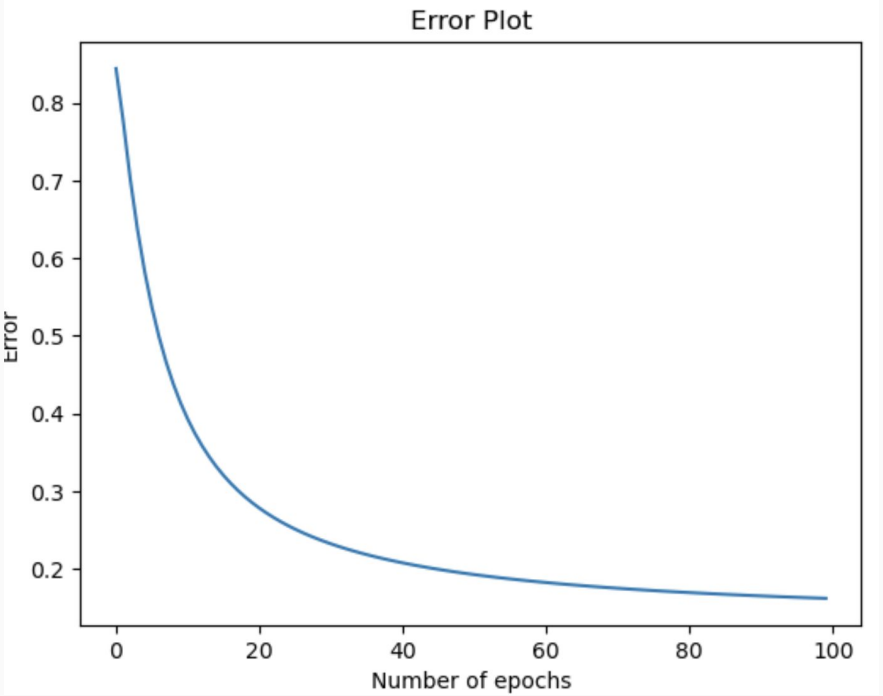
iteration = 65

learning_rate = 1

iteration = 65

For Part 2 (samples, your results may be different)





Submission

- **Due 11/12/2024**
- All data and initial code are on Canvas Assignment Page
- Jupyter Notebook code in Github
- PDF with copy of your code and shows the plotted graphs for each algorithm.