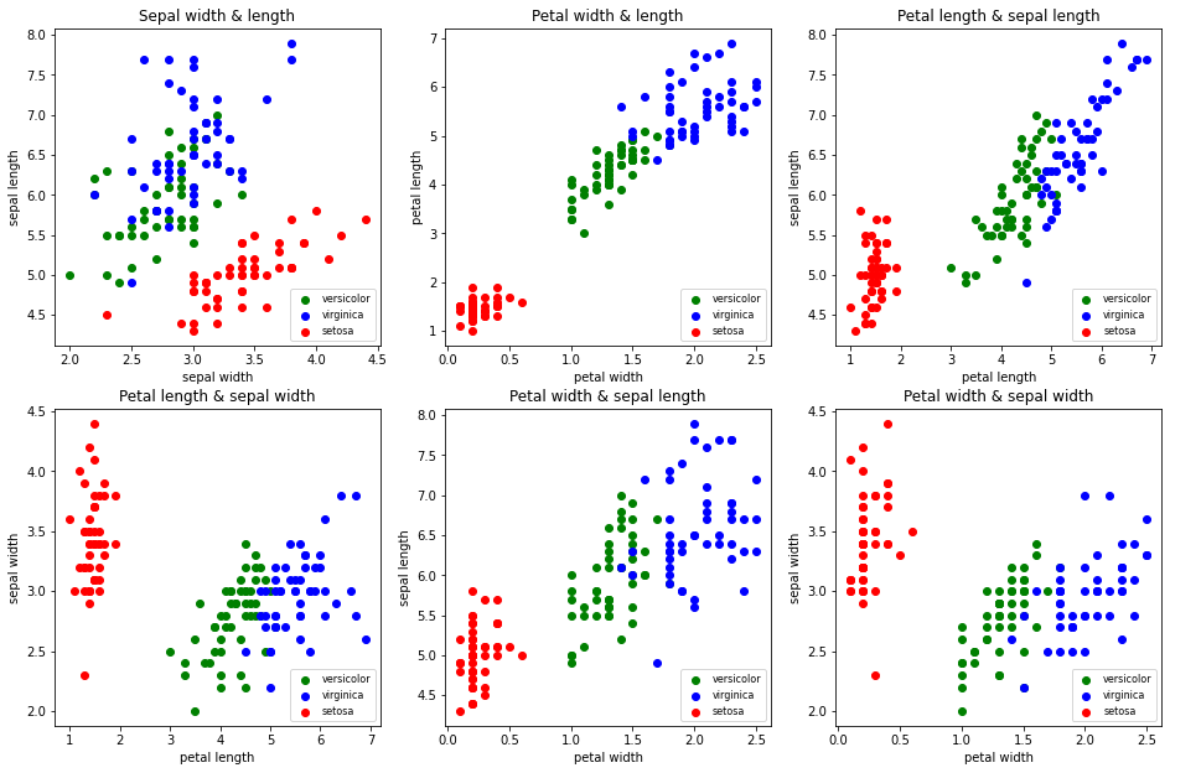
COMP5400M: BIC Coursework 2

Student ID: 201373470

1. Plots shows as below.



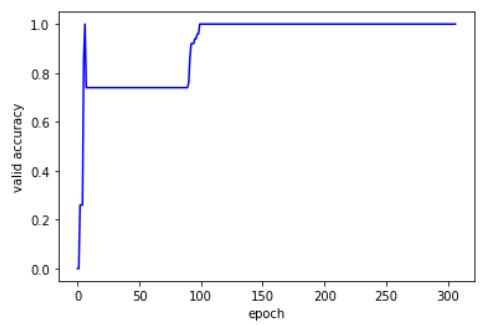
1. I do believe that setosa vs. non-setosa can be learnt by a perceptron. Based on the plots above, there are 6 plots of different 2 inputs’ combations. And we can see that all of these plots are linear separable between setosa and non-setosa flowers. Especially for the plot that use petal width and length as its inputs. There is a huge gap between setosa and the others, which means we can use one single decision line to separate it easily. In conclusion, the formular of the perceptron is *y + 2.8 \* x – 3.5 = 0.* The weight of petal length is 1, petal width is 2.8, both of sepal width and length are 0, and the bias is -3.5. The decision line on the plot shows as below.



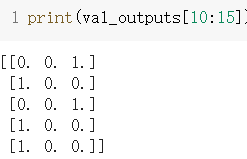
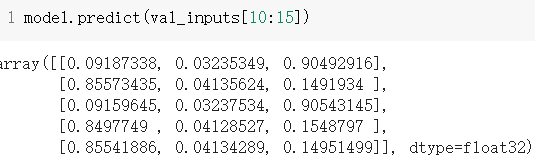
1. No, it’s not gonna converge without learning rate. When using the gradient

descent algorithm for optimization, the weight update rule will multiply the gradient term by a coefficient. This coefficient is the learning rate. Without it, there will be no update to the weight and also no optimization for the model. By the end of training, the loss remain unchanged and the predicted numpy doesn’t match the output labels. So the output result is also incorrect.

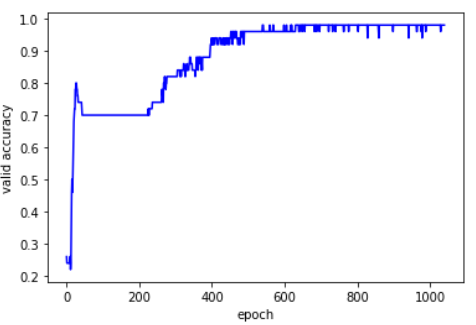
In order to make the gradient descent optimizer have better performance, I need to set the value of the learning rate in an appropriate range. And I have set the learning rate as 0.08. In this way, it’s not likely to exceed the optimal value and also the algorithm can converge for a quite short time. Besides, the stopping criterion is that when the valid accuracy is not improved anymore within 200 epoches, then the training procedure is stopped. The plot for epoch and valid accuracy as below.



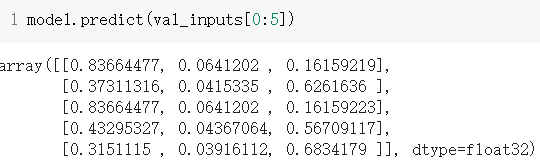
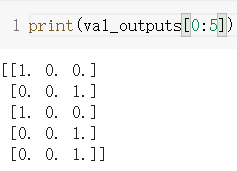
As for the output result, below are the part of the predict result for valid data set and the output labels. I set the setosa label as [1. 0. 0.], and for both versicolor and virginica, which is non-setosa labels are [0. 0. 1.]. From the predict array, the highest probability prefect match the output labels, which proved that the results are correct. Although there are 3 rows in one array, the second and third are represent the same non-setosa label.



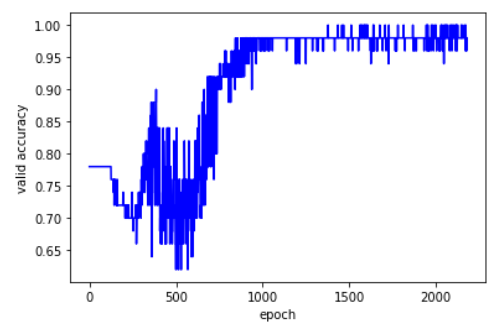
As for virginica vs. non-virginica classification. Without learning rate, the algorithm is still not gonna converge, and I set learning rate as 0.05, tolerated non-improved epoches is 400. The plot for epoch and valid accuracy as below.



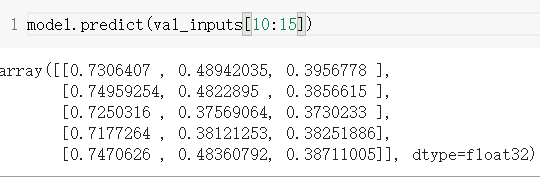
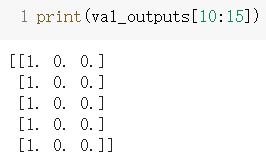
Part of the predict result for valid data set and the output labels show as below. And the result looks good.

As for versicolor vs. non-versicolor classification. Without learning rate, the algorithm is still not gonna converge, and I set learning rate as 0.08, tolerated non-improved epoches is 800. The plot for epoch and valid accuracy as below.



Part of the predict result for valid data set and the output labels show as below.

And there are differences between versicolor and the other two. According to the plots in Q1, we can see that the versicolor data (green points in Q1 plots) is in the middle of setosa and virginica. This phenomenon also tells us that it’s unlikely to separate versicolor from other flowers with one single descision line. And that also explain why it takes more epoches to train a perceptron for the versicolor vs. non-versicolor classification problem.

