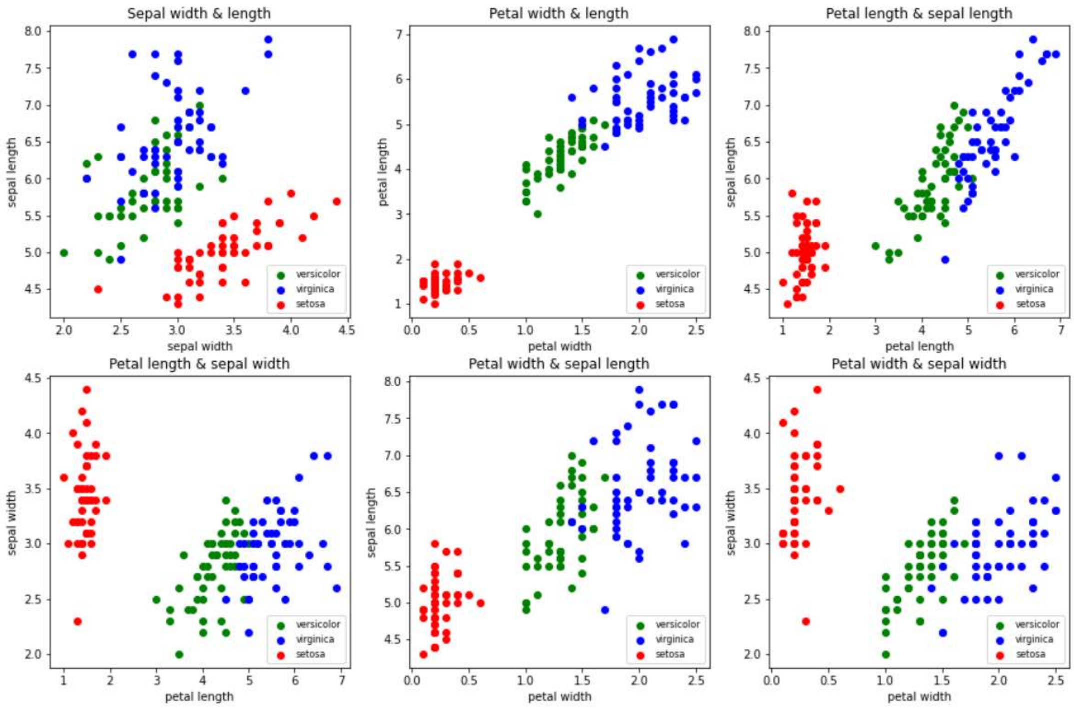
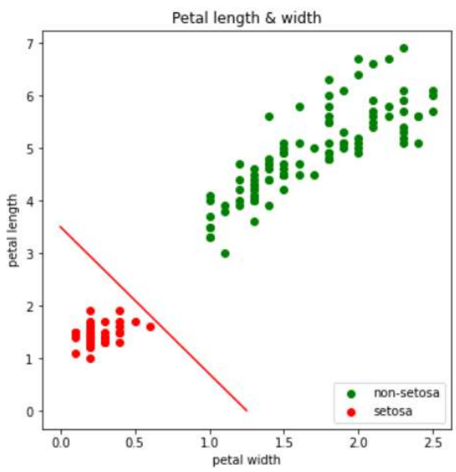
COMP5400M: BIC Coursework 2

Student ID: 201373470

1. Plots show as below.

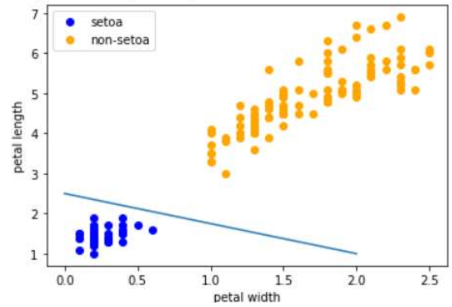


2. I do believe that a perceptron can learn setosa vs non-setosa. Based on the plots above, there are six plots of different two inputs' combinations. And we can see that all of these plots are linear separable between setosa and non-setosa flowers, especially for the plot that uses petal width and length 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 formula 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.



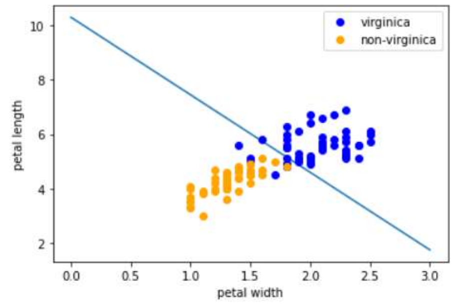
3. As for setosa vs non-setosa classification, the algorithm for training the perceptron without learning rate can converge. From the plot, we can observe that the setosa data is totally separable from non-setosa data. As long as we set rough but appropriate initial weights and bias, it still converges even without learning rate. And the output is correct. As the plot below, the decision line separates the setosa and non-setosa data completely.

<matplotlib.legend.Legend at 0x7f29c6e132e8>



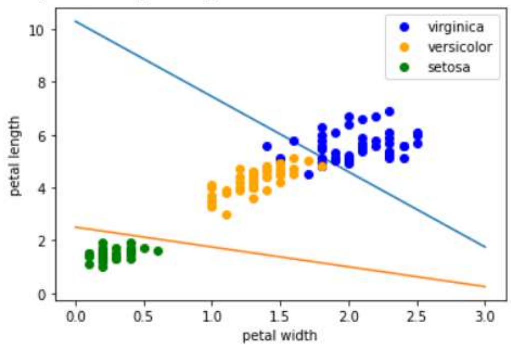
As for Virginia vs non-Virginia classification, the algorithm cannot converge without learning rate. A little part of Virginia data is overlap with the non-Virginia. So it needs a learning rate to improve the perceptron classify ability and separate the data as correct as possible. And I set the learning rate as 0.1. If the perceptron classifies more than 95 data point (out of 100) correctly within one epoch, then I believe the perceptron is good enough so far, and the program will stop training.

<matplotlib.legend.Legend at 0x7f29c6ab9898>

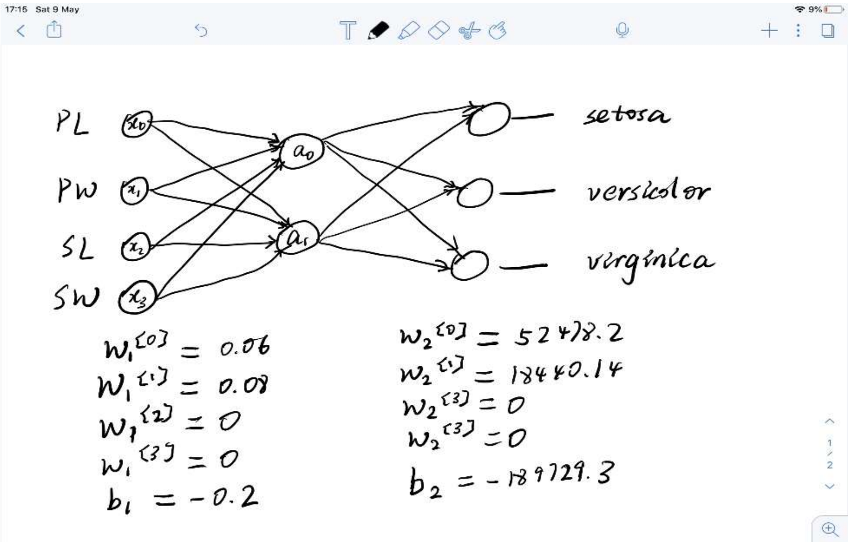


And there are differences between Versicolor and the other two. According to the plots in Q1, we can see that the Versicolor data (green points in Q1 plots) is in the middle of setosa and Virginia. This phenomenon also tells us that it's unlikely to separate Versicolor from other flowers with one single decision line or one perceptron. To classify Versicolor need to build multi-layer perceptron network. Or use the previous two perceptrons to sort the data. If the data doesn't belong to either Virginia nor setosa, then it's Versicolor. As the two decision lines in the plot below.

<matplotlib.legend.Legend at 0x7f29c6bdc9b0>



4. I use multi-layer perceptron to build the network. With four inputs and three outputs. Two previous perceptions as two nodes in the hidden layer and also with RELU activation function. The hand-draw network and all the weights and biases show as below.



After 1000 epoch, evaluate the model with valid dataset and labels and get the accuracy as 0.64.

```
accuracy: 0.64
```

5. When using valid dataset and label to evaluate the Keras based model, the accuracy can achieve more than 0.90. And the result also shows that the Keras has better performance than mine.

```
Epoch 1496/1500
100/100 [==============================] - 0s 112us/sample - loss: 0.0477 - acc: 0.9800 - val_loss: 0.0625 - val_acc: 0.9800
Epoch 1497/1500
100/100 [==============================] - 0s 93us/sample - loss: 0.0476 - acc: 0.9900 - val_loss: 0.0653 - val_acc: 0.9000
Epoch 1498/1500
100/100 [==============================] - 0s 99us/sample - loss: 0.0477 - acc: 0.9800 - val_loss: 0.0622 - val_acc: 0.9800
Epoch 1499/1500
100/100 [==============================] - 0s 105us/sample - loss: 0.0474 - acc: 0.9900 - val_loss: 0.0639 - val_acc: 0.9200
Epoch 1500/1500
100/100 [==============================] - 0s 113us/sample - loss: 0.0473 - acc: 0.9800 - val_loss: 0.0625 - val_acc: 0.9600
<tensorflow.python.keras.callbacks.History at 0x7f099a560e10>
```

The model's ability to predict inputs is not perfect. A small part of the Versicolor and Virginia data overlapped, which makes it hard to tell the differences and predicate labels. Also, the perceptron can deal with the linear sperate problem. Clearly, in the case of the iris, some of the data are not complete linear separable, so the classification cannot be perfect.