SID: 200968838

Bioinspired computing

Coursework 1

Question 1

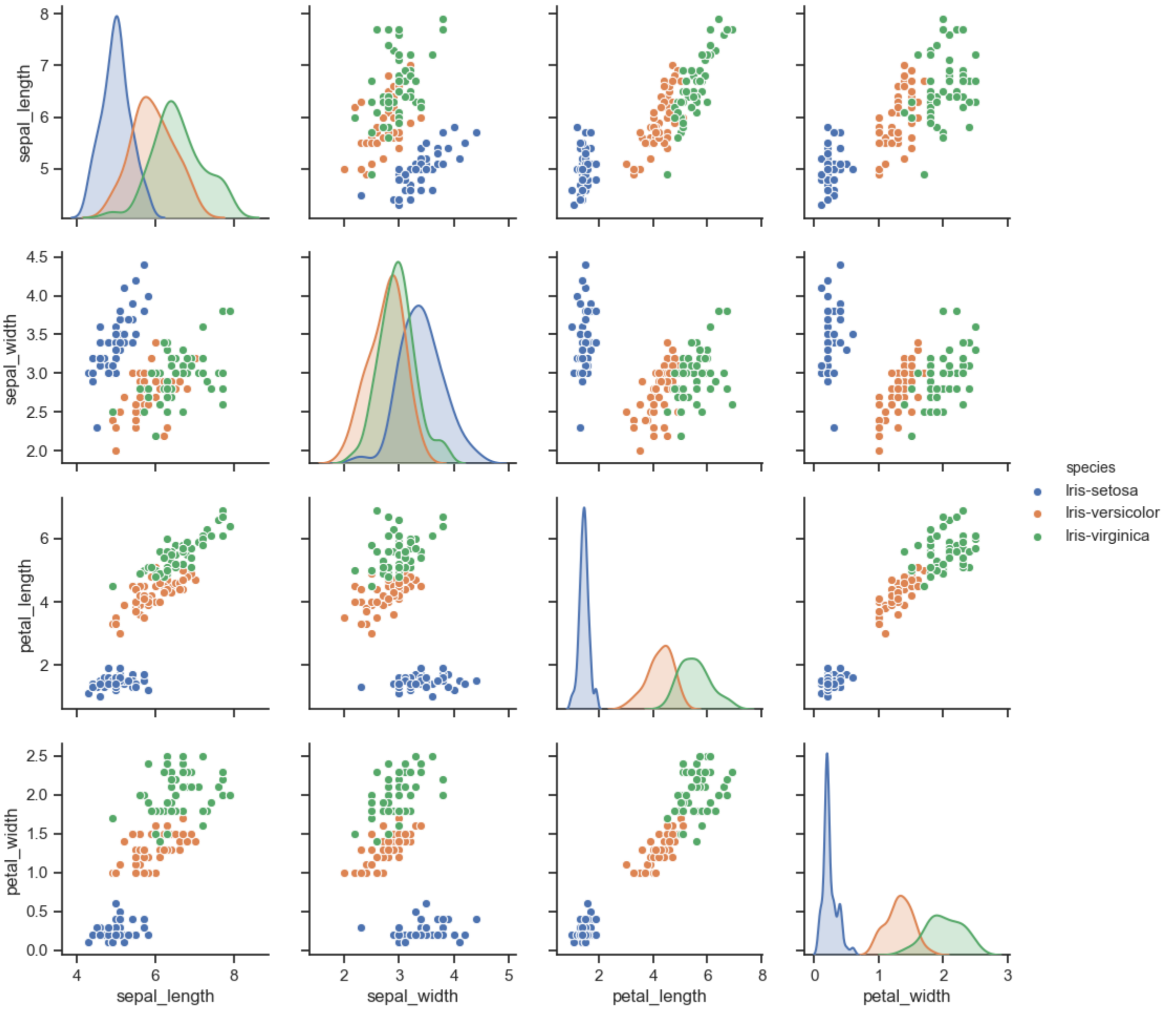


Figure 1 Scatter matrix of the iris dataset produce using the seaborn library

Question 2

In all the scatter plots abov the points classified as setosa (in blue) are linearly separable from all the others therefore we can safely say they can be classified by a single 2 inputs perceptron in all of the cases shown above, for certain parameters even a single input perceptron would be able to correctly classify setosa vs non setosa. This means that on the 4 inputs perceptron we would just set to 0 the weights of the inputs we don't use and get a clear separation between setosa and non setosa. Some of this possible simplified perceptron use weights [-5.1, 3.9, 0, 0, 15.0] for the one classifying against sepal width and length and [0, 0, -1.5, -4, 5] for the one classifying against petal width and length.

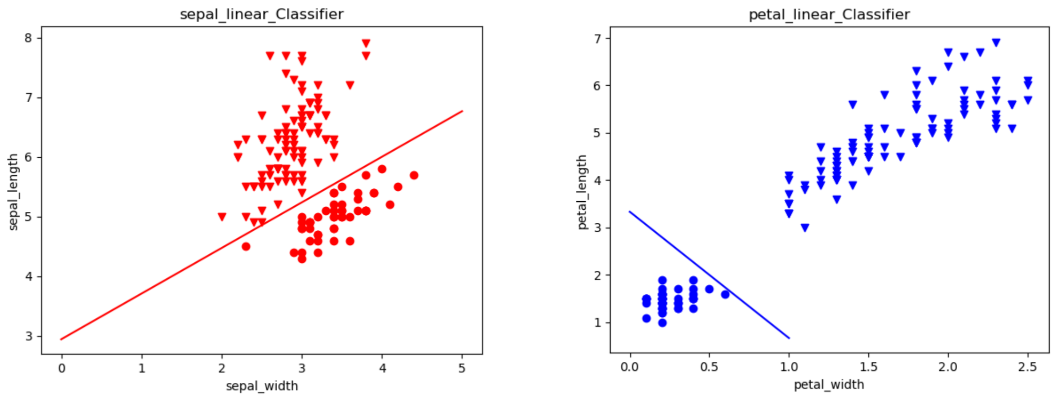


Figure 2 Sepal and petal classifiers circle are setosa triangles are non setosa

Question 3

As it can be seen from the scatter matrix on the first page, when comparing setosa to non setosa, have clear boundary condition in all the 2D instances and in the case of petal length and width even the one-dimensional representation has a clear boundary condition. The combination of these factors leads me to believe that the perceptron learning algorithm is separable in 4 dimensions by a hyperplane.

The setosa vs non setosa perceptron converges without the need to add a learning rate to the algorithm and in a number of iterations smaller than the entries in the data set. The weights obtained are [-0.50070756 2.35675675 -1.57468868 -1.05504908 0.02127709].

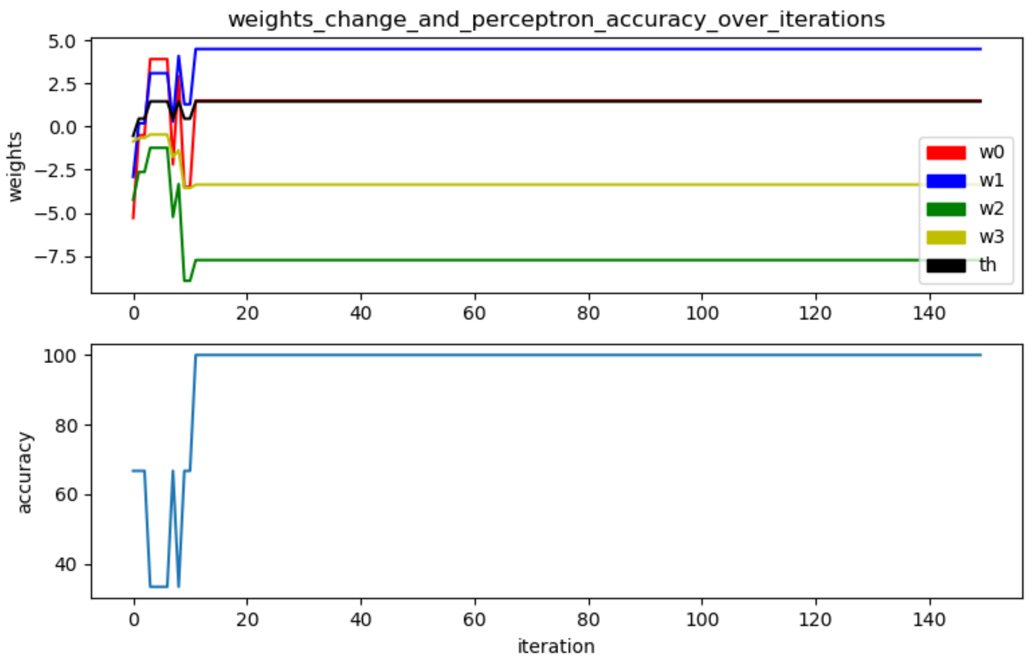


Figure 3 Weight and accuracy change over iterations for setosa vs non setosa perceptron

Moving on to the other to classifiers the situation is quite as good since both of those are not linearly classifiable for any combination of 2 parameters. This means that the classifier for virginica and versicolor won't have 100% classification accuracy. Virginica will probably get close to perfect accuracy since in most instances the overlap is minimal. Therefore, I would expect that, by adding learning rate to the algorithm the final perceptron would probably have accuracy above 90%.

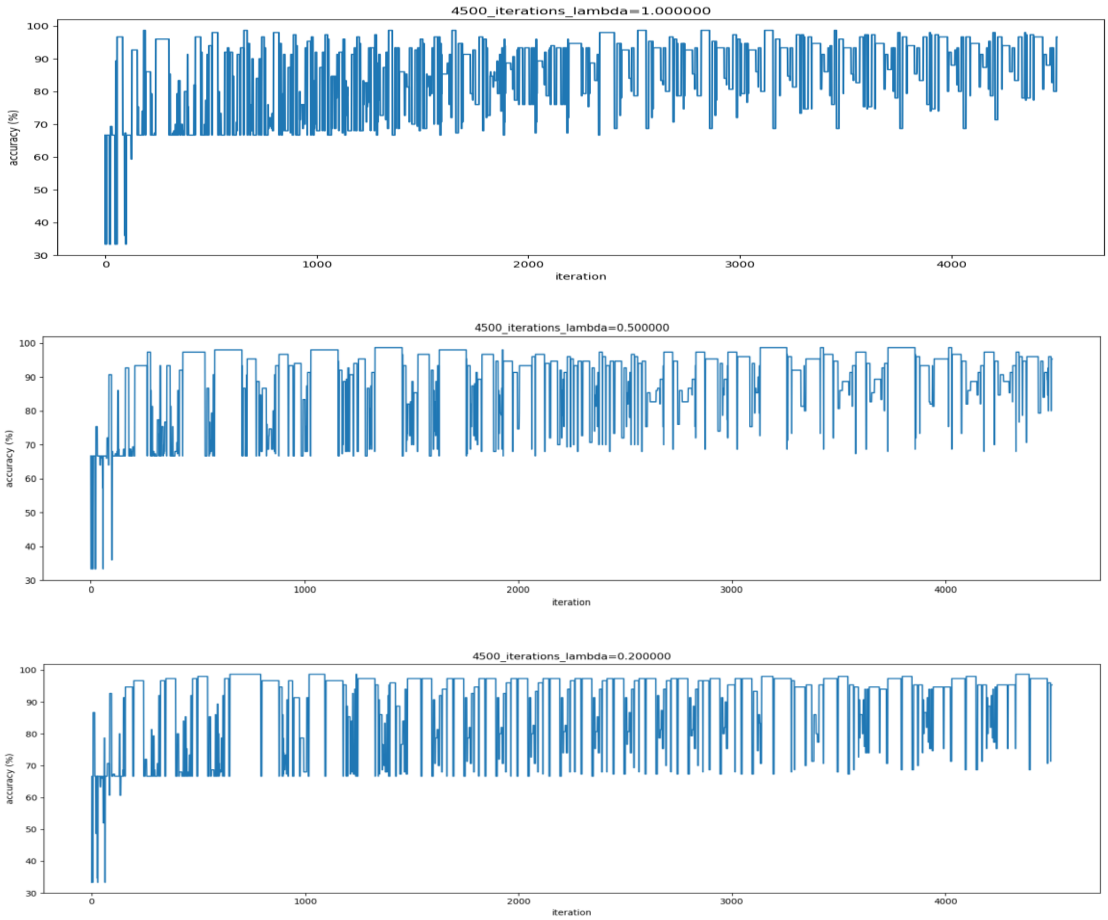


Figure 4 Accuracy of the virginica classifier during training

We can see that even after 4500 iteration of the code the classifier for virginica never reaches 100% classification accuracy. Reducing the learning rate makes it so the classifier's weights change by a smaller amount over each iteration. This Can be observed in the graphs where the accuracy jitters less and less the smaller the learning rate is.

The best classifier we can obtain for virginica has an accuracy of 98% with weights [-2.50485896 -2.20186858 3.62850465 3.19246628 -0.98835995]

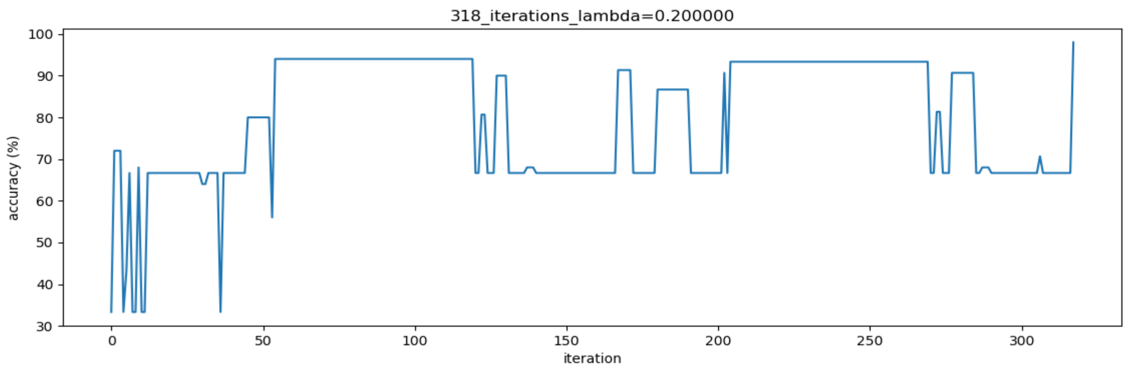
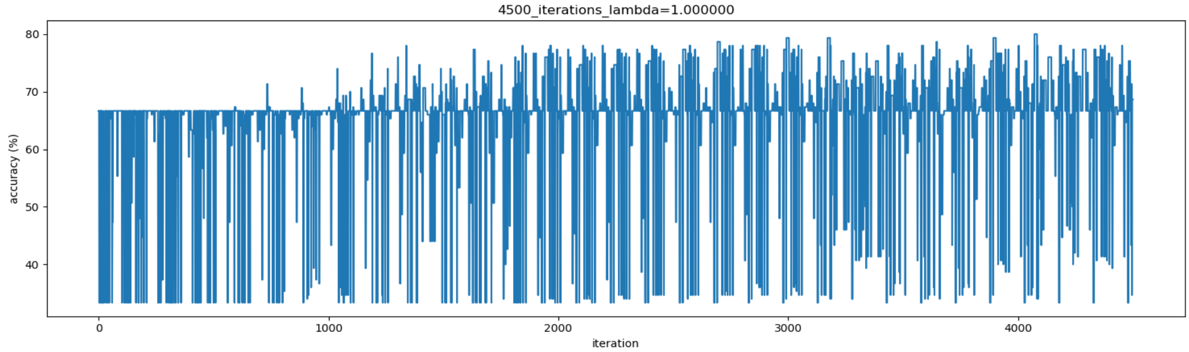
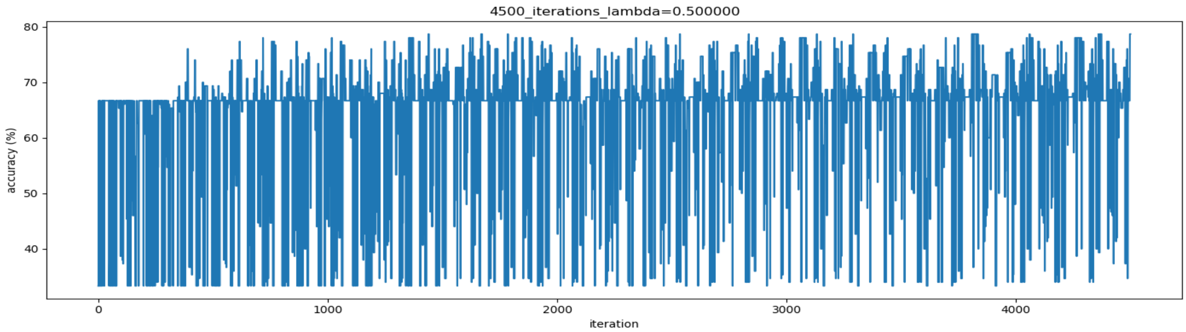


Figure 5 Virginica classifier accuracy during training with exit condition set to 98% classification accuracy

Versicolor on the other hand is not suitable to be classified by a single perceptron since all the datapoints in the dataset are situated in-between the data points of setosa and virginica, making it impossible to dive from the latter with a single line.





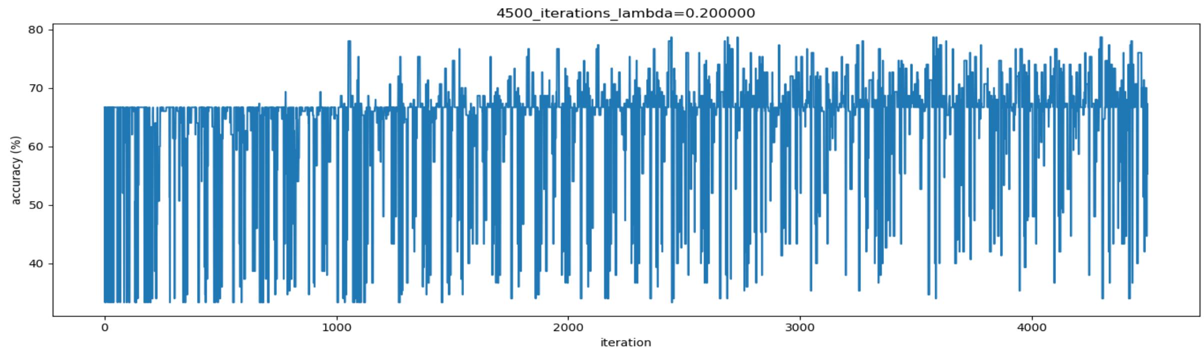


Figure 6 Accuracy of the versicolor classifier during training

While training the classifier accuracy never exceeds 78% even while decreasing the learning rate. When compared to the virginica training we can also see that going through training the classification accuracy changes much more frantically. The weights obtained from training are [2.96015761 -5.7989648   1.04779707 -5.10024532 1.98070432]

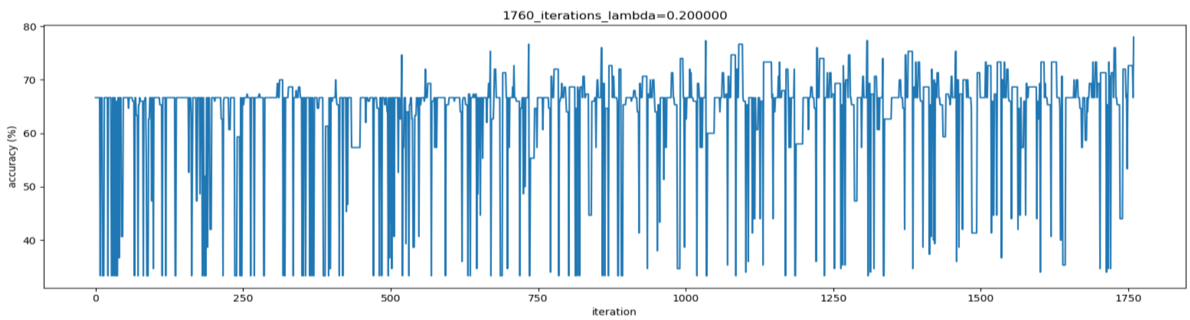


Figure 7 Versicolor classifier accuracy during training with exit condition set to 98% classification accuracy

Question 4

The perceptrons trained in the previous section can be used to build a multilayer neural network to be used to classify an iris for sepal length and width and petal length and width. Two different architecture have been developed, both have 4 inputs and 3 outputs but one has 3 hidden neurons while the other only has 2.

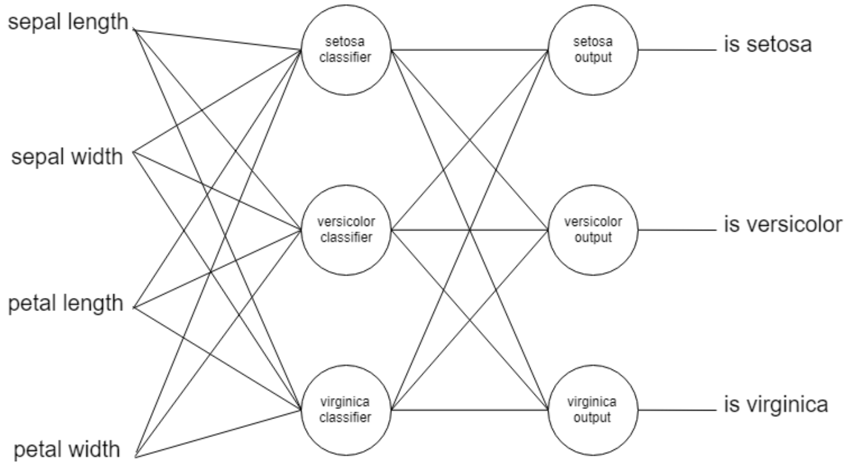


Figure 8 3 perceptrons hidden layer iris classifier

The weights for the first layer's perceptrons are the same as specified in the previous section and the weights for the output perceptrons are, from top to bottom, [1.0, 0.0, 0.0, -0.5], [-0.5, 1, -0.5, -0.6] and [-1.1, -0.6, 0.5, 0.5]. These values create the following logic.

Setosa classifier	Versicolor classifier	Virginica classifier	Setosa Output	Versicolor Output	Virginica Output
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	0	1
1	0	0	1	0	0
1	0	1	1	0	0
1	1	0	1	0	0
1	1	1	1	0	0

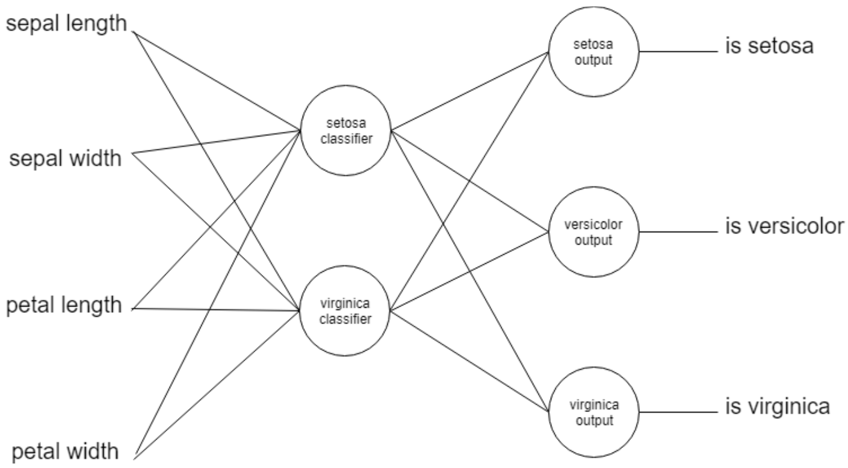


Figure 9 2 perceptrons hidden layer iris classifier

The weights for the first layer's perceptrons are the same as specified in the previous section and the weights for the output perceptrons are, from top to bottom, [1.0, 0.0, -0.5], [-0.5, -0.5, -0.4] and [-0.6, 0.6 0.5]. These values create the following logic.

Setosa classifier	Virginica classifier	Setosa Output	Versicolor Output	Virginica Output
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	1	0	1

When comparing the 2 network the one with 2 neurons hidden layer has higher classification accuracy. This can be attributed to the fact that it only uses data coming from reliable classifiers and returns a versicolor classification only when neither setosa nor virginica have been recognised.