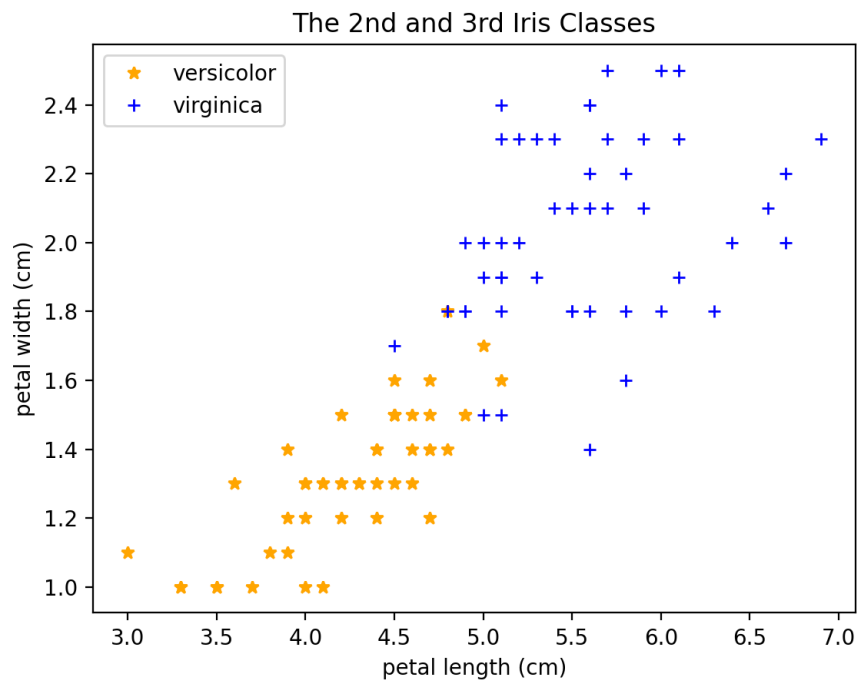


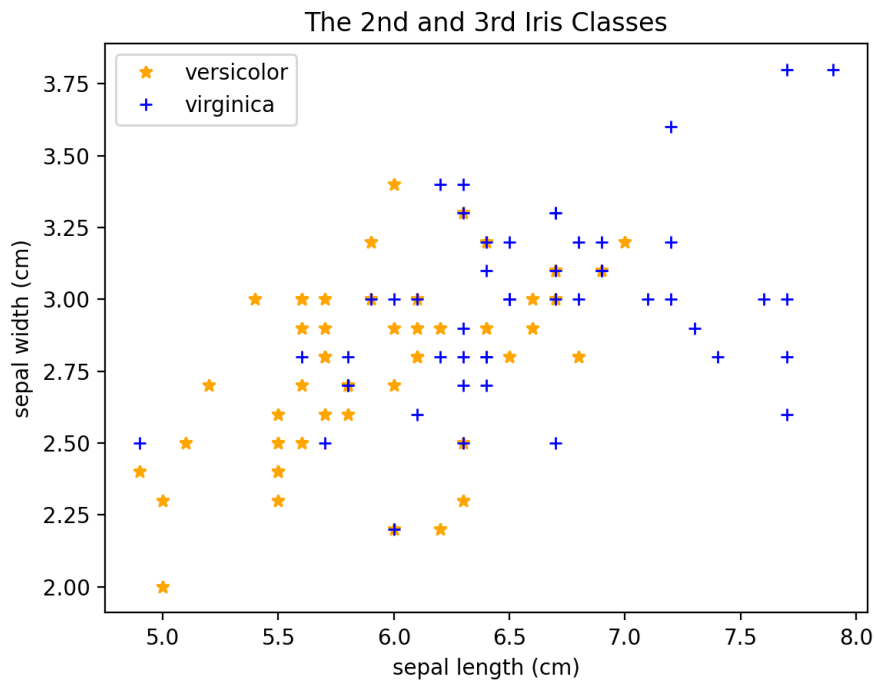
Exercise 1

(a)

The plot of 2nd and 3rd classes of the dataset with petal width vs. petal length:



The plot of 2nd and 3rd classes of the dataset with sepal width vs. sepal length:



We can see that petal width vs. petal length can better separate the 2nd and 3rd classes.

(b)

We define the output unit: $y = w_1 \cdot \text{petal_length} + w_2 \cdot \text{petal_width} + w_0$

w_1 and w_2 are weights, w_0 is bias.

The sigmoid function is $\text{sigmoid} = \frac{1}{1 + e^{-y}}$

sigmoid function code:

```
#sigmoid function
def sigmoid(length:float, width:float) -> float:
    w = [-3.18, 0.30, 0.98]
    z = w[0] + w[1] * length + w[2] * width
    sigmoid = 1 / (1 + math.exp(-z))
    return sigmoid
```

The sigmoid function classifies the data. If the value of sigmoid is less than 0.5, the data is flower is classified as versicolor with an output of 0. Otherwise, the flower is classified as virginica with an output of 1.

code that computes output:

```
def output(length:float, width:float) -> float:
    sig = sigmoid(length, width)
    if (sig < 0.5):
        return 0
    else:
        return 1
```

(c)

Because the output unit is $y = w_1 \cdot \text{petal_length} + w_2 \cdot \text{petal_width} + w_0$ and the decision boundary is when $y = 0$,

we use x_1 to represent petal length and x_2 to represent petal width:

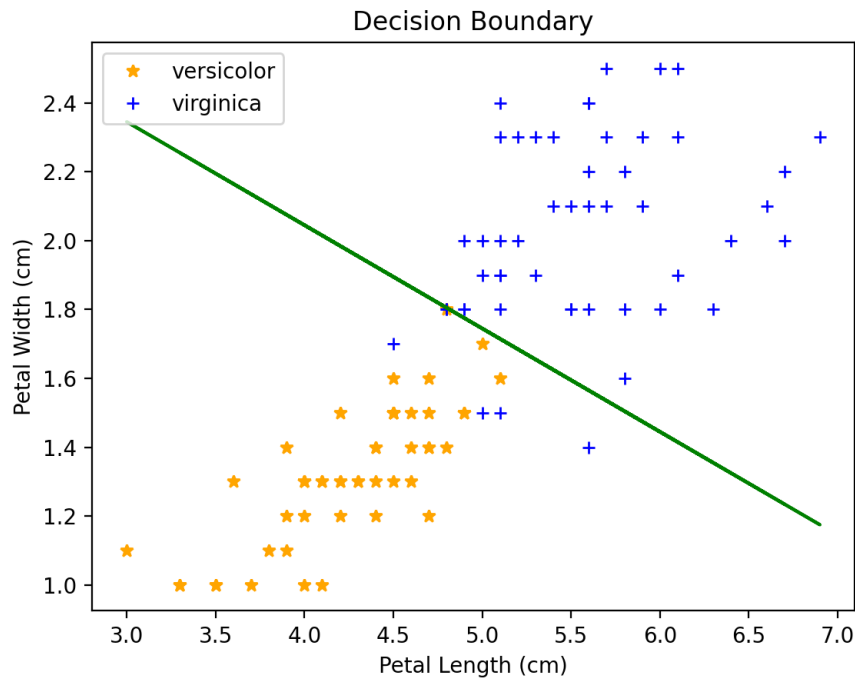
$$0 = w_1 \cdot x_1 + w_2 \cdot x_2 + w_0$$

$$x_2 = -\frac{w_1 \cdot x_1 + w_0}{w_2}$$

decision boundary code:

```
w = [-3.18, 0.30, 0.98]
x = dataset['petal_length']
y = [-(w[1] * x_value + w[0]) / w[2]] for x_value in x]
```

We draw the decision boundary on the graph using python: `plt.plot(x_1, x_2)`



(d)

From the graph generated in (a), we can see that the range of petal length is from about 3.0 to 7.0 cm and the range of petal width is from about 1.0 to 2.5 cm, so we set the range of petal length from 3.0 to 7.0 cm and the range of petal width from 1.0 to 2.5 cm for the surface that we are going to draw.

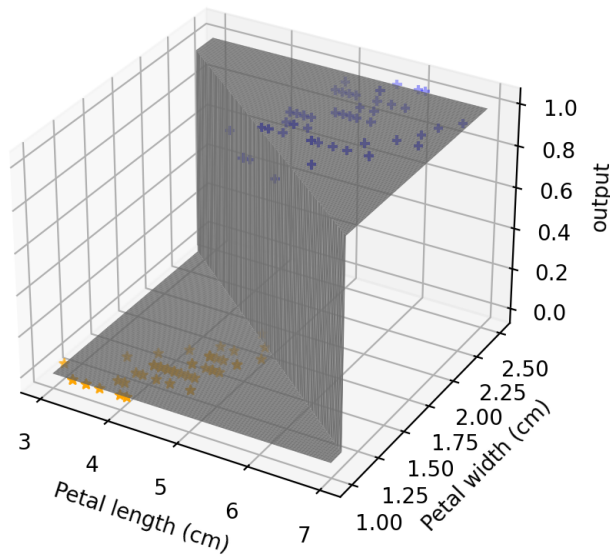
Then we use the output function we defined in (b) to calculate the output of each pair of petal length and petal width.

Code to prepare petal length, petal width, and the output:

```
x = np.arange(3.0, 7.0, 0.01)
y = np.arange(1.0, 2.5, 0.01)
X, Y = np.meshgrid(x, y)
z = np.array([[output(a,b) for a,b in zip(np.ravel(X), np.ravel(Y))]])
Z = z.reshape(X.shape)
```

We use the petal length, petal width, and the output to draw a 3D neural network

Output of Neural Network



(e)

We selected 8 examples.

petal length (cm)	petal width (cm)	species
4.8	1.8	versicolor
5.0	1.7	versicolor
3.5	1.0	versicolor
3.8	1.1	versicolor
5.1	1.8	virginica
5.0	1.5	virginica
4.9	1.8	virginica
5.9	2.3	virginica

