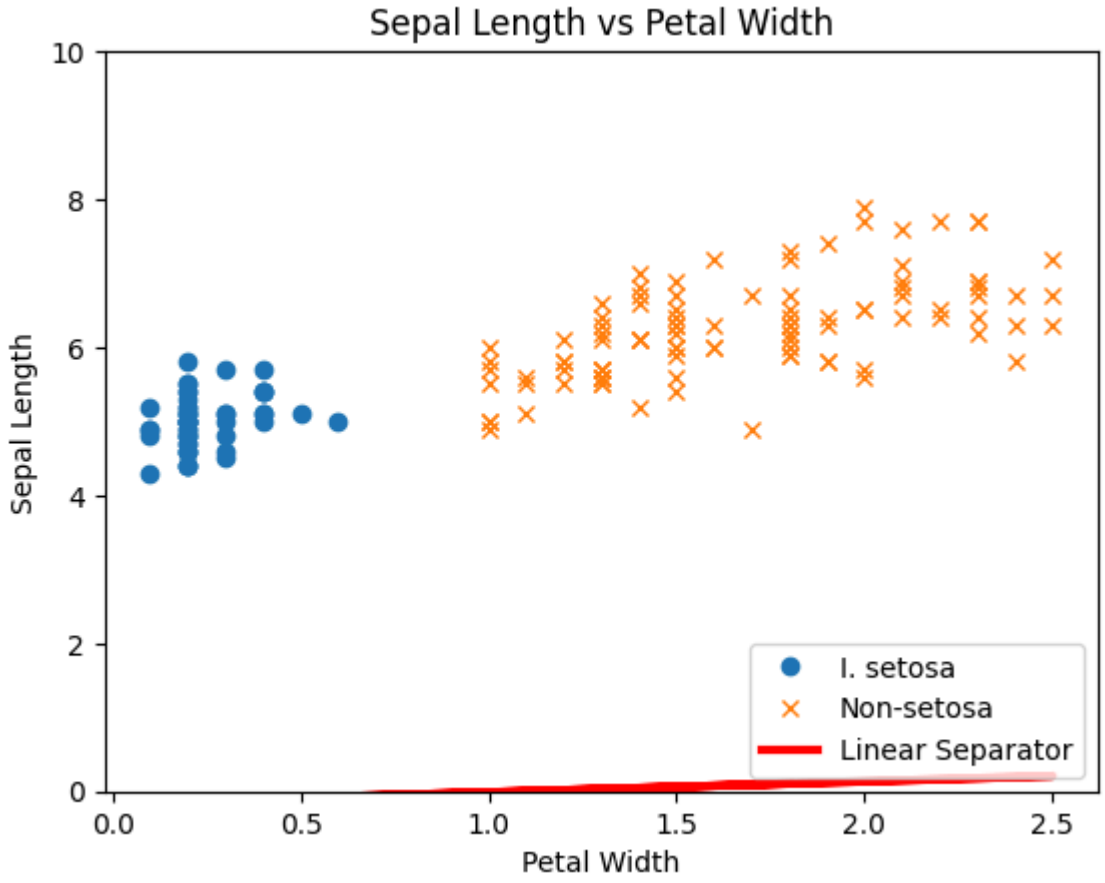
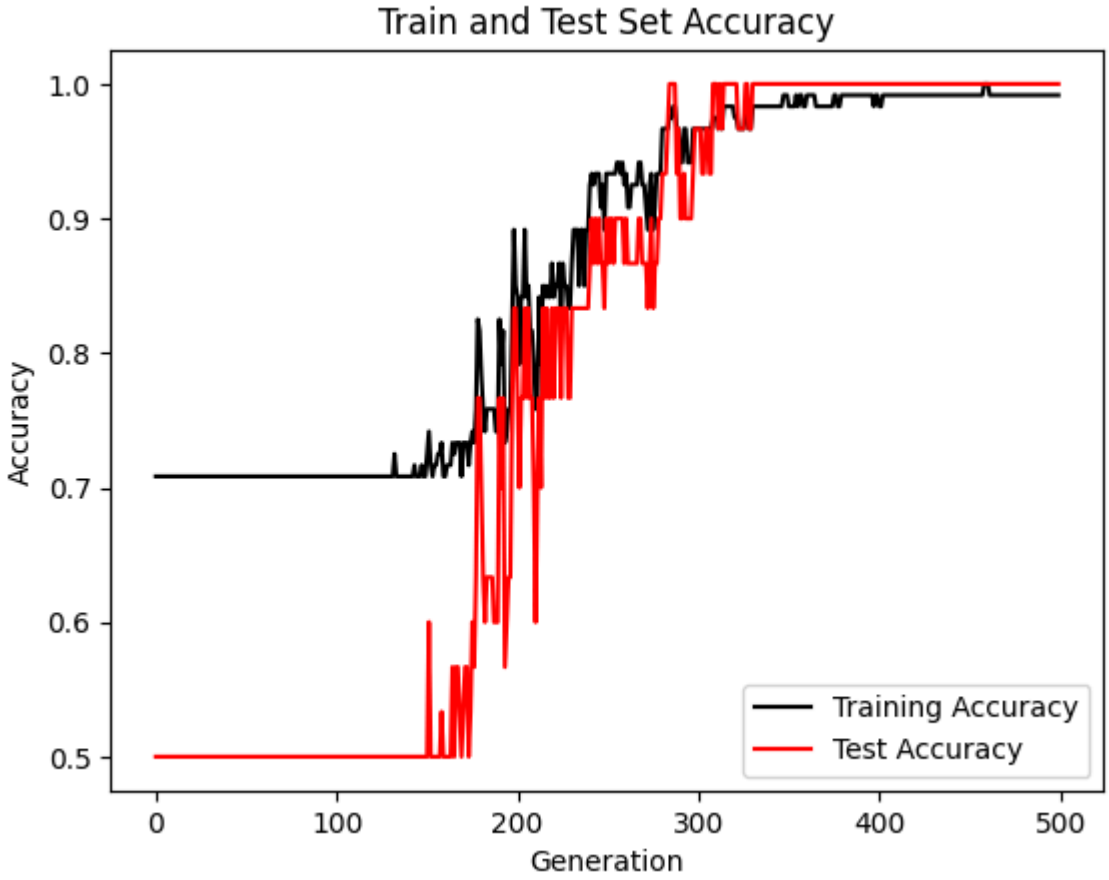
In [1]:

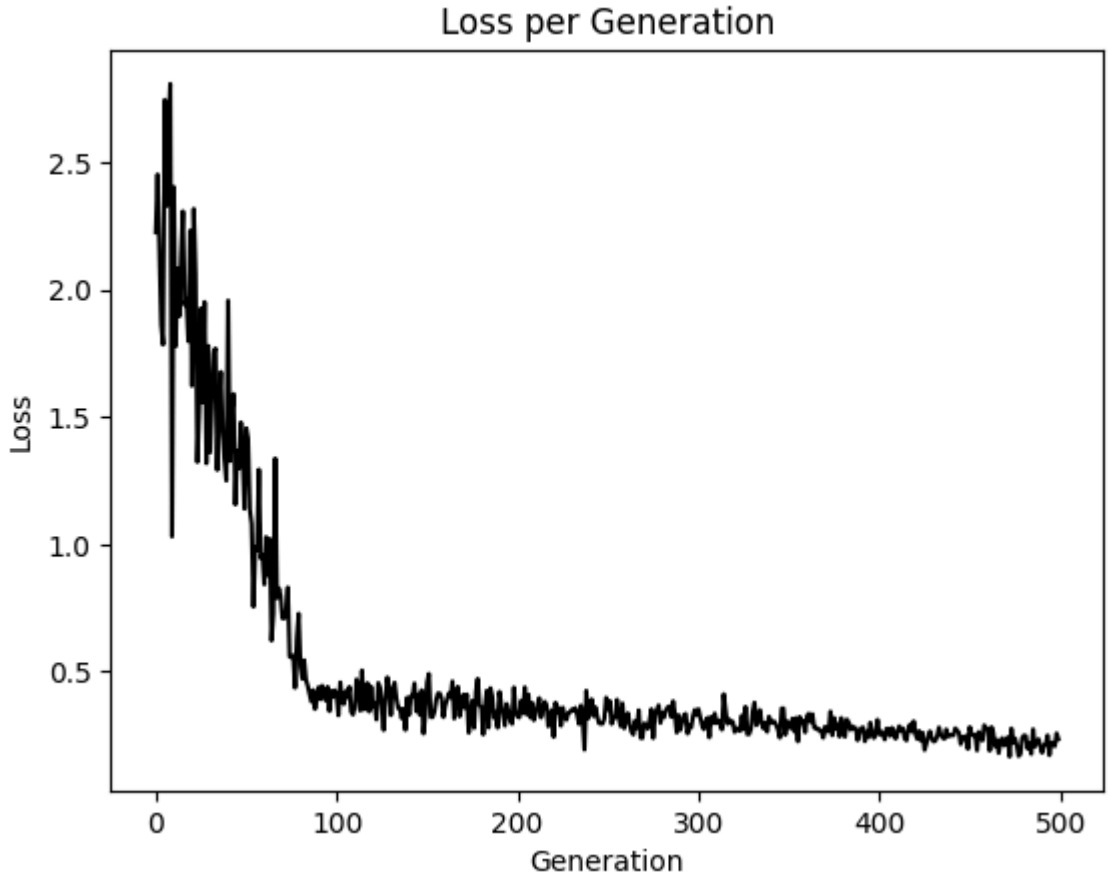
```
# 使用 tensorflow.compat.v1 和 tf.disable_v2_behavior() 来兼容 TensorFlow 1.x。

import matplotlib.pyplot as plt
import numpy as np
import tensorflow.compat.v1 as tf  # Use TensorFlow v1 compatibility mode
from sklearn import datasets

tf.disable_v2_behavior()  # Disable eager execution

# Load and preprocess data
iris = datasets.load_iris()
x_vals = np.array([[x[0], x[3]] for x in iris.data])
y_vals = np.array([1 if y == 0 else -1 for y in iris.target])

# Train/test split
train_indices = np.random.choice(len(x_vals), round(len(x_vals) * 0.8), replace=
test_indices = np.array(list(set(range(len(x_vals))) - set(train_indices)))
x_vals_train = x_vals[train_indices]
x_vals_test = x_vals[test_indices]
y_vals_train = y_vals[train_indices]
y_vals_test = y_vals[test_indices]

# Model parameters
batch_size = 100
x_data = tf.placeholder(shape=[None, 2], dtype=tf.float32)
y_target = tf.placeholder(shape=[None, 1], dtype=tf.float32)
A = tf.Variable(tf.random_normal(shape=[2, 1]))
b = tf.Variable(tf.random_normal(shape=[1, 1]))

# SVM Model
model_output = tf.subtract(tf.matmul(x_data, A), b)
l2_norm = tf.reduce_sum(tf.square(A))
alpha = tf.constant(0.01)
classification_term = tf.reduce_mean(tf.maximum(0., tf.subtract(1., tf.multiply(
loss = tf.add(classification_term, tf.multiply(alpha, l2_norm))

# Prediction and accuracy
prediction = tf.sign(model_output)
accuracy = tf.reduce_mean(tf.cast(tf.equal(prediction, y_target), tf.float32))

# Optimization
optimizer = tf.train.GradientDescentOptimizer(0.01)
train_step = optimizer.minimize(loss)
init = tf.global_variables_initializer()

# Run session
with tf.Session() as sess:
    sess.run(init)
    loss_vec = []
    train_accuracy = []
    test_accuracy = []

    for i in range(500):
        rand_index = np.random.choice(len(x_vals_train), size=batch_size)
        rand_x = x_vals_train[rand_index]
        rand_y = np.transpose([y_vals_train[rand_index]])

        sess.run(train_step, feed_dict={x_data: rand_x, y_target: rand_y})
```

```
        temp_loss = sess.run(loss, feed_dict={x_data: rand_x, y_target: rand_y})
        loss_vec.append(temp_loss)

        train_acc_temp = sess.run(accuracy, feed_dict={x_data: x_vals_train, y_t
        train_accuracy.append(train_acc_temp)

        test_acc_temp = sess.run(accuracy, feed_dict={x_data: x_vals_test, y_tar
        test_accuracy.append(test_acc_temp)

        if (i + 1) % 100 == 0:
            print('Stop #' + str(i+1) + 'A =' + str(sess.run(A)) + 'b =' + str(s
            print('Loss =', temp_loss)

    # Retrieve model parameters
    [[a1], [a2]] = sess.run(A)
    [[b]] = sess.run(b)
    slope = -a1 / a2
    y_intercept = b / a2

# Plotting results
x_vals_plot = [d[1] for d in x_vals]
best_fit = [slope * i + y_intercept for i in x_vals_plot]

setosa_x = [d[1] for i, d in enumerate(x_vals) if y_vals[i] == 1]
setosa_y = [d[0] for i, d in enumerate(x_vals) if y_vals[i] == 1]
not_setosa_x = [d[1] for i, d in enumerate(x_vals) if y_vals[i] == -1]
not_setosa_y = [d[0] for i, d in enumerate(x_vals) if y_vals[i] == -1]

plt.plot(setosa_x, setosa_y, 'o', label="I. setosa")
plt.plot(not_setosa_x, not_setosa_y, 'x', label='Non-setosa')
plt.plot(x_vals_plot, best_fit, 'r-', label='Linear Separator', linewidth=3)
plt.ylim([0, 10])
plt.legend(loc='lower right')
plt.title('Sepal Length vs Petal Width')
plt.xlabel('Petal Width')
plt.ylabel('Sepal Length')
plt.savefig('linear_separator.png')
plt.show()

plt.plot(train_accuracy, 'k-', label='Training Accuracy')
plt.plot(test_accuracy, 'r-', label='Test Accuracy')
plt.title("Train and Test Set Accuracy")
plt.xlabel('Generation')
plt.ylabel('Accuracy')
plt.legend(loc='lower right')
plt.savefig('accuracy.png')
plt.show()

plt.plot(loss_vec, 'k-')
plt.title('Loss per Generation')
plt.xlabel('Generation')
plt.ylabel("Loss")
plt.savefig('loss.png')
plt.show()
```

```
WARNING:tensorflow:From C:\Users\fl\AppData\Local\Temp\ipykernel_18544\188333007
7.py:8: The name tf.disable_v2_behavior is deprecated. Please use tf.compat.v1.di
sable_v2_behavior instead.

WARNING:tensorflow:From C:\Users\fl\AppData\Local\Packages\PythonSoftwareFoundati
on.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-packages\te
nsorflow\python\compat\v2_compat.py:98: disable_resource_variables (from tensorfl
ow.python.ops.resource_variables_toggle) is deprecated and will be removed in a f
uture version.
Instructions for updating:
non-resource variables are not supported in the long term
WARNING:tensorflow:From C:\Users\fl\AppData\Local\Temp\ipykernel_18544\188333007
7.py:42: The name tf.train.GradientDescentOptimizer is deprecated. Please use tf.
compat.v1.train.GradientDescentOptimizer instead.

Stop #100A =[[ 0.07941679]
 [-0.7193926 ]]b =[[0.44058007]]
Loss = 0.42560723
Stop #200A =[[ 0.12253466]
 [-0.92332613]]b =[[0.39137986]]
Loss = 0.3391244
Stop #300A =[[ 0.15164283]
 [-1.1227256 ]]b =[[0.34507978]]
Loss = 0.32662183
Stop #400A =[[ 0.19010557]
 [-1.3265513 ]]b =[[0.29387966]]
Loss = 0.30920973
Stop #500A =[[ 0.2231495]
 [-1.5164369]]b =[[0.24427962]]
Loss = 0.23328632
```




Train and Test Set Accuracy
Accuracy
Generation
Training Accuracy
Test Accuracy
1.0
0.9
0.8
0.7
0.6
0.5
0
100
200
300
400
500


Loss per Generation
Loss
Generation
2.5
2.0
1.5
1.0
0.5
0
100
200
300
400
500