```
import numpy as np
import matplotlib.pyplot as plt

# Sample data
X1 = np.array([[1, 1], [-1, -1], [0, 0.5], [0.1, 0.5], [0.2, 0.2], [0.9, 0.5]])
X2 = np.array([1, -1, -1, -1, 1, 1])

# Initial weight vector and bias
w = np.array([1, 1])
b = 0

# Perceptron learning algorithm
converged = False
epoch = 0
while not converged:
    epoch += 1
    misclassified = False
    for i in range(len(X)):
        x = X1[i]
        y = X2[i]
        y_hat = np.dot(w, x) + b
        if y * y_hat <= 0:
            w = w + lr * np.dot(y, x)
            b = b + lr * y
            misclassified = True
            print("Epoch {}: Sample {}: x={}, y={}, y_hat={}, Incorrect, Update w={}, b={}".format(epoch, i+1, x, y, y_hat, w, b))
        else:
            print("Epoch {}: Sample {}: x={}, y={}, y_hat={}, Correct".format(epoch, i+1, x, y, y_hat))
    if not misclassified:
        converged = True

# Final weight vector and decision boundary
print("\nFinal weight vector: w={}, b={}".format(w, b))
print("Decision boundary: {}x1 + {}x2 + {} = 0".format(w[0], w[1], b))

# Plot decision boundary
x = np.linspace(-1, 1, 100)
y = -w[0]/w[1]*x - b/w[1]
plt.scatter(X1[:, 0], X1[:, 1], c=X2, cmap='bwr')
plt.plot(x, y, '-r')
plt.xlim(-1, 1)
plt.ylim(-1, 1)
plt.xlabel('x1')
plt.ylabel('x2')
plt.title('Perceptron Learning Algorithm (Epoch {})'.format(epoch))
plt.show()
```

```
Epoch 1: Sample 1: x=[1. 1.], y=1, y_hat=2.0, Correct
Epoch 1: Sample 2: x=[-1. -1.], y=-1, y_hat=-2.0, Correct
Epoch 1: Sample 3: x=[0.  0.5], y=-1, y_hat=0.5, Incorrect, Update w=[1.  0.5], b=-1
Epoch 1: Sample 4: x=[0.1 0.5], y=-1, y_hat=-0.65, Correct
Epoch 1: Sample 5: x=[0.2 0.2], y=1, y_hat=-0.7, Incorrect, Update w=[1.2 0.7], b=0
Epoch 1: Sample 6: x=[0.9 0.5], y=1, y_hat=1.4300000000000002, Correct
Epoch 2: Sample 1: x=[1. 1.], y=1, y_hat=1.9, Correct
Epoch 2: Sample 2: x=[-1. -1.], y=-1, y_hat=-1.9, Correct
Epoch 2: Sample 3: x=[0.  0.5], y=-1, y_hat=0.35, Incorrect, Update w=[1.2 0.2], b=-1
Epoch 2: Sample 4: x=[0.1 0.5], y=-1, y_hat=-0.78, Correct
Epoch 2: Sample 5: x=[0.2 0.2], y=1, y_hat=-0.72, Incorrect, Update w=[1.4 0.4], b=0
Epoch 2: Sample 6: x=[0.9 0.5], y=1, y_hat=1.46, Correct
Epoch 3: Sample 1: x=[1. 1.], y=1, y_hat=1.7999999999999998, Correct
Epoch 3: Sample 2: x=[-1. -1.], y=-1, y_hat=-1.7999999999999998, Correct
Epoch 3: Sample 3: x=[0.  0.5], y=-1, y_hat=0.19999999999999998, Incorrect, Update w=[ 1.4 -0.1], b=-1
Epoch 3: Sample 4: x=[0.1 0.5], y=-1, y_hat=-0.91, Correct
Epoch 3: Sample 5: x=[0.2 0.2], y=1, y_hat=-0.74, Incorrect, Update w=[1.6 0.1], b=0
Epoch 3: Sample 6: x=[0.9 0.5], y=1, y_hat=1.49, Correct
Epoch 4: Sample 1: x=[1. 1.], y=1, y_hat=1.6999999999999997, Correct
Epoch 4: Sample 2: x=[-1. -1.], y=-1, y_hat=-1.6999999999999997, Correct
Epoch 4: Sample 3: x=[0.  0.5], y=-1, y_hat=0.04999999999999999, Incorrect, Update w=[ 1.6 -0.4], b=-1
Epoch 4: Sample 4: x=[0.1 0.5], y=-1, y_hat=-1.04, Correct
Epoch 4: Sample 5: x=[0.2 0.2], y=1, y_hat=-0.76, Incorrect, Update w=[ 1.8 -0.2], b=0
Epoch 4: Sample 6: x=[0.9 0.5], y=1, y_hat=1.5199999999999998, Correct
Epoch 5: Sample 1: x=[1. 1.], y=1, y_hat=1.5999999999999999, Correct
Epoch 5: Sample 2: x=[-1. -1.], y=-1, y_hat=-1.5999999999999999, Correct
Epoch 5: Sample 3: x=[0.  0.5], y=-1, y_hat=-0.1, Correct
Epoch 5: Sample 4: x=[0.1 0.5], y=-1, y_hat=0.07999999999999999, Incorrect, Update w=[ 1.7 -0.7], b=-1
Epoch 5: Sample 5: x=[0.2 0.2], y=1, y_hat=-0.8, Incorrect, Update w=[ 1.9 -0.5], b=0
Epoch 5: Sample 6: x=[0.9 0.5], y=1, y_hat=1.4599999999999997, Correct
Epoch 6: Sample 1: x=[1. 1.], y=1, y_hat=1.3999999999999997, Correct
Epoch 6: Sample 2: x=[-1. -1.], y=-1, y_hat=-1.3999999999999997, Correct
Epoch 6: Sample 3: x=[0.  0.5], y=-1, y_hat=-0.24999999999999997, Correct
Epoch 6: Sample 4: x=[0.1 0.5], y=-1, y_hat=-0.06, Correct
Epoch 6: Sample 5: x=[0.2 0.2], y=1, y_hat=0.27999999999999997, Correct
Epoch 6: Sample 6: x=[0.9 0.5], y=1, y_hat=1.4599999999999997, Correct

Final weight vector: w=[ 1.9 -0.5], b=0
Decision boundary: 1.8999999999999997x1 + -0.49999999999999994x2 + 0 = 0
```