perceptron-xor

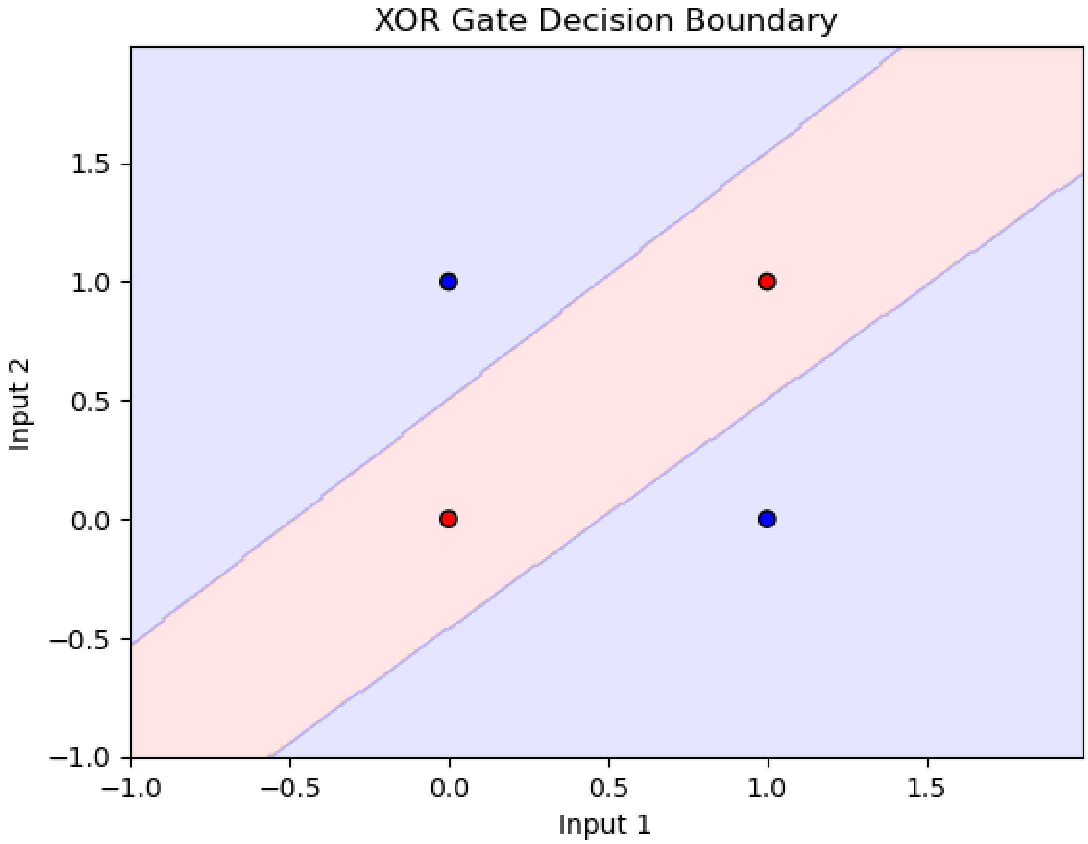
December 13, 2023

[5]:

```
import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap

from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score
import numpy as np

def plot_decision_boundary(X, y, model, title):
    h = 0.01
    x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
    y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
    xx, yy = np.meshgrid(np.arange(x_min, x_max, h), np.arange(y_min, y_max, h))

    Z = model.predict(np.c_[xx.ravel(), yy.ravel()])
    Z = Z.reshape(xx.shape)

    plt.contourf(xx, yy, Z, cmap=ListedColormap(['#FFAAAA', '#AAAAFF']),␣
 ↪alpha=0.3)
    plt.scatter(X[:, 0], X[:, 1], c=y, cmap=ListedColormap(['#FF0000',␣
 ↪'#0000FF']), edgecolors='k', marker='o')
    plt.title(title)
    plt.xlabel('Input 1')
    plt.ylabel('Input 2')
    plt.show()


inputs = np.array([[0, 0],
                   [0, 1],
                   [1, 0],
                   [1, 1]])

outputs = np.array([0, 1, 1, 0])

mlp = MLPClassifier(hidden_layer_sizes=(3,), activation='relu', max_iter=10000,␣
 ↪random_state=42)

mlp.fit(inputs, outputs)
```

```python
test_data = np.array([[0, 0],
                      [0, 1],
                      [1, 0],
                      [1, 1]])

predictions = mlp.predict(test_data)

accuracy=accuracy_score(outputs, predictions)
print("Predictions after training:")
print(predictions)
print("Accuracy:",accuracy)
plot_decision_boundary(inputs, outputs, mlp, 'XOR Gate Decision Boundary')
```

```
Predictions after training:
[0 1 1 0]
Accuracy: 1.0
```



[ ]: