Machine Learning
Homework #4
Question 3 Supplementary v2

Daniel Ho Kwan Leung
104360098
CSIE, Year 3

Question 3:

The desired points are,

$$x = \frac{1}{\sqrt{2}} \; or \; -\frac{1}{\sqrt{2}}$$
$$y = \frac{1}{\sqrt{2}} \; or \; -\frac{1}{\sqrt{2}}$$
$$\lambda = \frac{1}{\sqrt{2}} \; or \; -\frac{1}{\sqrt{2}}$$

$$when\ x = y = \lambda = \frac{1}{\sqrt{2}},\ f(x,y) \approx 1.4142135623730951$$

$$when\ x = y = \lambda = -\frac{1}{\sqrt{2}},\ f(x,y) \approx -1.4142135623730951$$

$$\therefore x = y = \lambda = \frac{1}{\sqrt{2}}\ will\ lead\ to\ maximum\ point\ of\ f(x,y)\ subject\ to\ x^2 + y^2 = 1$$

We can further add higher penalty so that we can achieve the point desired by gradient search method. Since we are going to increase the penalty, the term λ can be omitted.

$$let\ \mathcal{L}(x,y) = f(x,y) - 100[g(x,y)]^2$$
$$= x + y - 100(x^2 + y^2 - 1)^2$$

$$\frac{\partial \mathcal{L}}{\partial x} = 1 - 400x^3 - 400y^2x + 400x$$

$$\frac{\partial \mathcal{L}}{\partial y} = 1 - 400y^3 - 400x^2y + 400y$$

$$Initial\ State: x, y = random\ initialization\ with\ normal\ distribution$$

The code was uploaded to GitHub:

https://github.com/DHKLeung/NTUT_Machine_Learning/blob/master/HW4_Q3_higher_penalty_no_lambda.ipynb

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

```
learning_rate = 0.0001
epoch = 100000
x = np.random.randn()
y = np.random.randn()
L_history = list()
```

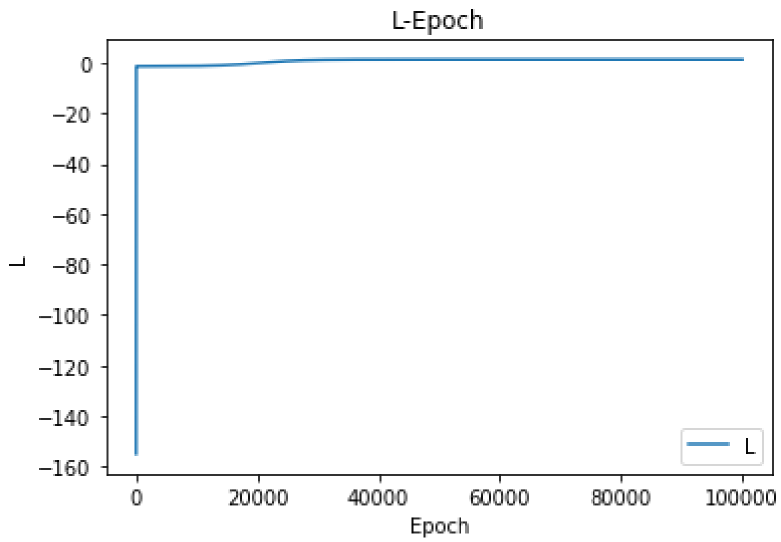
```
f = lambda x, y: x + y
g = lambda x, y: x**2 + y**2 - 1
l = lambda x, y: f(x, y) - 100 * (g(x, y))**2
L_history.append(l(x, y))
print('Initial State: L: {}, x: {}, y: {}'.format(L_history[0], x, y))
for i in range(epoch):
    temp_x = x + learning_rate * (1 - 400 * x**3 - 400 * y**2 * x + 400 * x)
    temp_y = y + learning_rate * (1 - 400 * x**2 * y - 400 * y**3 + 400 * y)
    x = temp_x
    y = temp_y
    L_history.append(l(x, y))
    print('Epoch {}: L: {}, x: {}, y: {}'.format(i + 1, L_history[i + 1], x, y))
```

```
plt.plot(L_history, label='L')
plt.title('L-Epoch')
plt.legend(loc='best')
plt.ylabel('L')
plt.xlabel('Epoch')
plt.show()
```

Outputs,

```
Initial State: L: -154.95919725652044, x: -0.9341371082804776, y: -1.1677938385033801
Epoch 1: L: -106.08477200875028, x: -0.8878401203577656, y: -1.1099415477793082
Epoch 2: L: -75.58798462492024, x: -0.8515080639961957, y: -1.0645457070135376
Epoch 3: L: -55.511640670009406, x: -0.8221732871801302, y: -1.0278967211953827
Epoch 4: L: -41.75531060930276, x: -0.7979822751957849, y: -0.9976776786345146
Epoch 5: L: -32.03210812039769, x: -0.7777049275094334, y: -0.9723509411160884
Epoch 6: L: -24.987251230331246, x: -0.7604863570698417, y: -0.9508478903075963
Epoch 7: L: -19.77894922052011, x: -0.7457104545499026, y: -0.9323983817073224
Epoch 8: L: -15.863534481703908, x: -0.7329199294487325, y: -0.9164307953067949
Epoch 9: L: -12.878464719957783, x: -0.7217669977774696, y: -0.9025103948033103
Epoch 10: L: -10.575388565290243, x: -0.7119816753829995, y: -0.8902996940301425
Epoch 11: L: -8.780235755164238, x: -0.7033506696174162, y: -0.8795320714136159
Epoch 12: L: -7.3685696533109475, x: -0.6957029058538512, y: -0.8699936780427531
Epoch 13: L: -6.249897912185869, x: -0.6888993482723721, y: -0.8615107136089268
Epoch 14: L: -5.357419008800791, x: -0.6828256809773864, y: -0.8539402777795541
Epoch 15: L: -4.641164557165322, x: -0.6773869426040579, y: -0.8471636634451505
Epoch 16: L: -4.063317953377743, x: -0.6725035247807736, y: -0.8410813547939325
Epoch 17: L: -3.594959881008407, x: -0.6681081415730172, y: -0.8356092391177856
Epoch 18: L: -3.213768502936975, x: -0.6641435023822897, y: -0.830675697939818

Epoch 99982: L: 1.415461359943938, x: 0.7083722431872217, y: 0.7083347206319167
Epoch 99983: L: 1.4154613599440786, x: 0.7083722405387163, y: 0.7083347232805626
Epoch 99984: L: 1.4154613599442192, x: 0.7083722378905848, y: 0.7083347259288345
Epoch 99985: L: 1.415461359944359, x: 0.7083722352428271, y: 0.7083347285767326
Epoch 99986: L: 1.4154613599444992, x: 0.7083722325954432, y: 0.7083347312242569
Epoch 99987: L: 1.4154613599446393, x: 0.7083722299484331, y: 0.7083347338714074
Epoch 99988: L: 1.4154613599447794, x: 0.7083722273017966, y: 0.7083347365181842
Epoch 99989: L: 1.4154613599449195, x: 0.7083722246555337, y: 0.7083347391645873
Epoch 99990: L: 1.4154613599450596, x: 0.7083722220096444, y: 0.7083347418106168
Epoch 99991: L: 1.4154613599451995, x: 0.7083722193641288, y: 0.7083347444562728
Epoch 99992: L: 1.4154613599453394, x: 0.7083722167189864, y: 0.7083347471015552
Epoch 99993: L: 1.4154613599454793, x: 0.7083722140742176, y: 0.7083347497464642
Epoch 99994: L: 1.4154613599456192, x: 0.7083722114298221, y: 0.7083347523909997
Epoch 99995: L: 1.4154613599457588, x: 0.7083722087857999, y: 0.708334755035162
Epoch 99996: L: 1.4154613599458987, x: 0.7083722061421509, y: 0.7083347576789509
Epoch 99997: L: 1.4154613599460386, x: 0.7083722034988752, y: 0.7083347603223666
Epoch 99998: L: 1.415461359946178, x: 0.7083722008559726, y: 0.7083347629654092
Epoch 99999: L: 1.415461359946318, x: 0.708372198213443, y: 0.7083347656080786
Epoch 100000: L: 1.4154613599464576, x: 0.7083721955712866, y: 0.708334768250375
```



We can see that the outcomes computed are close to the desired outputs, especially $x \approx y \approx 0.707 \approx \frac{1}{\sqrt{2}}$ and $\mathrm{L} \approx 1.41 \approx \text{desired } f(x, y)$.