Machine Learning HW #5 answer

1. No. ICA does not tell us anything about the order of independent components or how many of them are relevant.
2.

$$S_w = S_1 + S_2 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

$$S_w{}^{-1} = \frac{\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}}{4-1} = \begin{bmatrix} 2/3 & -1/3 \\ -1/3 & 2/3 \end{bmatrix}$$

$$w_{opt} = c_0 S_w{}^{-1}(\mu_x - \mu_y) = c_0 \begin{bmatrix} 2/3 & -1/3 \\ -1/3 & 2/3 \end{bmatrix} \begin{bmatrix} 0-2 \\ 0-0 \end{bmatrix}$$

$$= c_0 \begin{bmatrix} 2/3 & -1/3 \\ -1/3 & 2/3 \end{bmatrix} \begin{bmatrix} -2 \\ 0 \end{bmatrix} = c_0 \begin{bmatrix} -4/3 \\ 2/3 \end{bmatrix}$$

$$=> c_0\left(\frac{-4}{3}, \frac{2}{3}\right) = \sqrt{\frac{16c_0{}^2}{9} + \frac{4c_0{}^2}{9}} = \sqrt{\frac{20c_0{}^2}{9}} = \frac{\sqrt{20}c_0}{3}$$

$$w_{opt} = \left(\frac{-4}{3}c_0\frac{3}{\sqrt{20}c_0}, \frac{2}{3}c_0\frac{3}{\sqrt{20}c_0}\right) = \left(\frac{-4}{\sqrt{20}}, \frac{2}{\sqrt{20}}\right) = \left(\frac{-2}{\sqrt{5}}, \frac{1}{\sqrt{5}}\right)$$

$$\alpha_{opt} = w_{opt}\frac{\mu_x + \mu_y}{2} = \left(\frac{-2}{\sqrt{5}}, \frac{1}{\sqrt{5}}\right)\frac{\begin{bmatrix} 0+2 \\ 0+0 \end{bmatrix}}{2} = \frac{-2}{\sqrt{5}}$$

3.

$$\mathcal{L}(x, y, \lambda) = x^2 + y^2 + \lambda(x + y - 2)$$

$$\frac{\partial}{\partial x}\mathcal{L} = 2x + \lambda = 0$$

$$\frac{\partial}{\partial y}\mathcal{L} = 2y + \lambda = 0$$

$$\frac{\partial}{\partial \lambda}\mathcal{L} = x + y - 2 = 0$$

$$2x = -\lambda$$

$$2y = -\lambda$$

$$=> x = y$$

$$x + x - 2 = 2x - 2 = 0$$

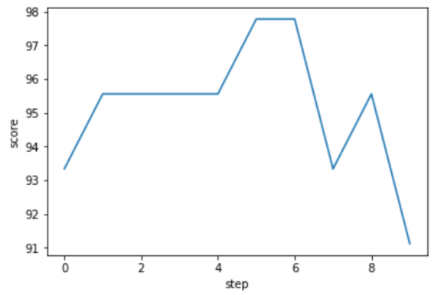
$$x = 1, y = 1, \lambda = -2$$

4.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
from sklearn.decomposition import PCA
from sklearn import datasets

iris_dataset = datasets.load_iris()

df_X = iris_dataset.data[:, :4]
df_y = iris_dataset.target
```

```
KNN_scores = []
def kNN_classifier_PCA(X_train, X_test, y_train, y_test):
    pca = PCA(n_components=3)
    pca.fit(X_train)
    X_train_tf = pca.transform(X_train)
    X_test_tf = pca.transform(X_test)

    classifier = KNeighborsClassifier(n_neighbors=5)
    classifier.fit(X_train_tf, y_train)
    return round(accuracy_score(classifier.predict(X_test_tf), y_test) * 100, 3)
```

```
for i in range(10):
        X_train, X_test, y_train, y_test = train_test_split(df_X, df_y, test_size=0.3)

        KNN_scores.append(kNN_classifier_PCA(X_train, X_test, y_train, y_test))

plt.plot(KNN_scores)
plt.xlabel('step')
plt.ylabel('score')
plt.show()

print(f"KNN Score = ", KNN_scores)
print(f"KNN Score average: {np.mean(KNN_scores)}")
```



KNN Score = [93.333, 95.556, 95.556, 95.556, 95.556, 97.778, 97.778, 93.333, 95.556, 91.111]

KNN Score average: 95.1113

5.

(a)

```
from decimal import Decimal

x = 1
y = 1
learning_rate = 0.0005

for i in range(1000):
    try:
        print("i = " + str(i) + " x = " + str(Decimal.from_float(x)) + " y = " + str(Decimal.from_float(y)))
        dx = 1 - 400 * x * (x**2 + y**2 - 1)
        dy = 1 - 400 * y * (x**2 + y**2 - 1)
        x = x - learning_rate * dx
        y = y - learning_rate * dy

    except OverflowError as e:
        print("Overflow!")
        break
```

```
i = 0 x = 1 y = 1
i = 1 x = 1.199500000000000010658141036401502788066864013671875 y = 1.199500000000000010658141036401502788066864013671875
i = 2 x = 1.6494363599500001527076165075413882732391357421875 y = 1.6494363599500001527076165075413882732391357421875
i = 3 x = 3.11405830487143564511143267736770212650299072265625 y = 3.11405830487143564511143267736770212650299072265625
i = 4 x = 14.570003332726432887511691660620272159576416015625 y = 14.570003332726432887511691660620272159576416015625
i = 5 x = 1248.85274885137232558918185532093048095703125 y = 1248.85274885137232558918185532093048095703125
i = 6 x = 779101876.85182130336761474609375 y = 779101876.85182130336761474609375
i = 7 x = 189165852963375348841447424 y = 189165852963375348841447424
i = 8 x = 2707623160904736228066902206884753661395287666999609532609530697787525134548992 y = 27076231609047362280669022068847536613952876669996
i = 9 x = 794007587356151867191168096870025448634376628474934807524240099158727477235653664256234792416477795012441070836462110842418633703817836
Overflow!
```

(b)

$$\mathcal{L}(\mathrm{x},\mathrm{y},\lambda) = x + y - 100(x^2 + y^2 - 1)^2$$

$$\frac{\partial}{\partial \mathrm{x}}\mathcal{L} = 1 - 200(x^2 + y^2 - 1)(2x) = 0$$

$$\frac{\partial}{\partial \mathrm{y}}\mathcal{L} = 1 - 200(x^2 + y^2 - 1)(2y) = 0$$

$$\frac{\partial}{\partial \lambda}\mathcal{L} = -200(x^2 + y^2 - 1) = 0$$

$$set\ \ k = (x^2 + y^2 - 1)$$

$$1 - 400kx = 0$$

$$1 - 400ky = 0$$

$$=> x = \frac{1}{400k} = y$$

$$2x^2 - 1 = 0$$

$$x^2 = \frac{1}{2}$$

$$x = \pm\frac{\sqrt{2}}{2}, y = \pm\frac{\sqrt{2}}{2}$$

No, they are different.

(c) We can use the estimated gradient by the centered difference formula.