機器學習概論作業

範圍： Gradient Descent

銘傳大學電腦與通訊工程系

班　　級	電通三乙
姓　　名	李柏賢
學　　號	07050862
作業成果	應繳作業共 3 題，前二題每題 30 分，第三題 40 分 我共完成 3 題，應得 100 分
授課教師	陳慶逸

■ 請確實填寫自己寫完成題數，填寫不實者(如上傳與作業明顯無關的答案，或是計算題數有誤者)，本次作業先扣 50 分。

一、試利用 gradient descent 求解 f(x) = x^3 – 3x + 2, -1<x<2 的最小值 (30%)

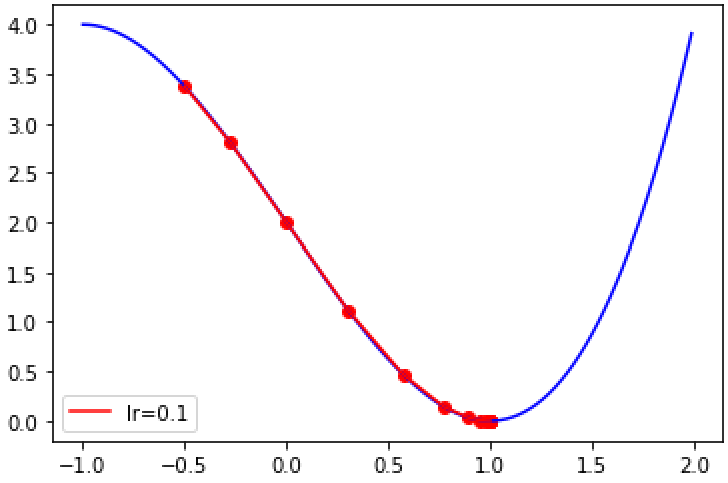
(1) x_start = - 0.5, learning rate = 0.1 時，epochs =15, 畫出圖形(15%)

(2) x_start = 2, learning rate = 0.05 時， epochs =15, 畫出圖形(15%)

程式碼 (x_start = - 0.5, learning rate = 0.1 時，epochs =15)

```
import numpy as np
import matplotlib.pyplot as plt
def func(x): return x**3-3*x + 2  #object fun
def dfunc(x): return 3*x*x - 3  #dy/dx=3x^2 - 3

#gradient descent fun
def GD(x_start, df, epochs, lr):
    xs = np.zeros(epochs+1)
    x = x_start
    xs[0] = x
    for i in range(epochs):
        dx = df(x)
        v = - lr * dx
        x += v
        xs[i+1] = x
    return xs
x_start = -0.5
epochs = 15
lr = 0.1        #learning rate
x = GD(x_start, dfunc, epochs, lr=lr)
color = 'r'
from numpy import arange
t = arange(-1.0, 2.0, 0.01)
plt.plot(t, func(t), c='b')
plt.plot(x, func(x), c=color, label='lr={}'.format(lr))
plt.scatter(x, func(x), c=color, )
plt.legend()
plt.show()
```

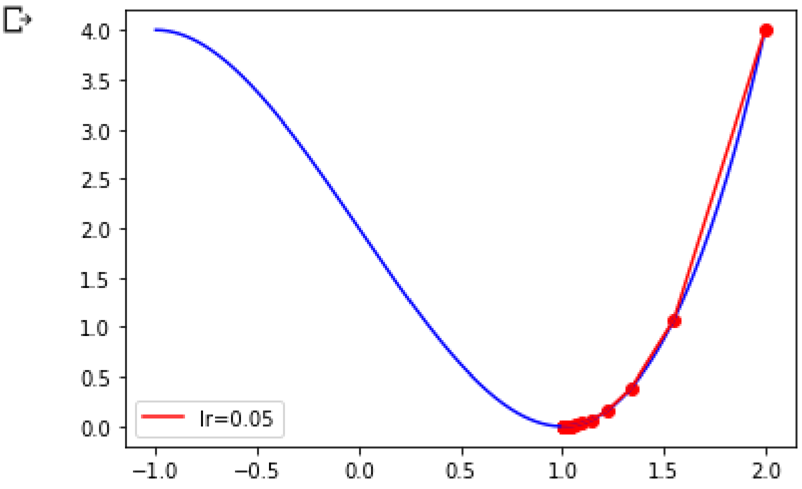
執行結果：



程式碼 (x_start = 2, learning rate = 0.05 時， epochs =15)

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

def func(x): return x**3-3*x + 2  #object fun

def dfunc(x): return 3*x*x - 3   #dy/dx=3x^2 - 3

#gradient descent fun
def GD(x_start, df, epochs, lr):
    xs = np.zeros(epochs+1)
    x = x_start
    xs[0] = x
    for i in range(epochs):
        dx = df(x)
        v = - lr * dx
        x += v
        xs[i+1] = x
    return xs
x_start = 2
epochs = 15
lr = 0.05        #learning rate

x = GD(x_start, dfunc, epochs, lr=lr)

color = 'r'

from numpy import arange
t = arange(-1.0, 2.0, 0.01)
plt.plot(t, func(t), c='b')
plt.plot(x, func(x), c=color, label='lr={}'.format(lr))
plt.scatter(x, func(x), c=color, )
plt.legend()
plt.show()
```

執行結果：



二、試利用 gradient descent 求解 f(x) = cosx, 0<x<2π 的最小值(30%)

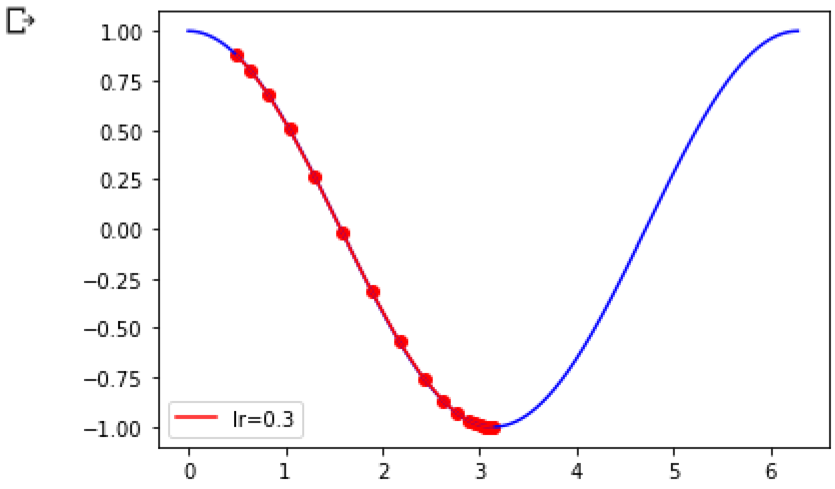
(1) x_start = 0.5, learning rate = 0.3 時，epochs =20, 畫出圖形(15%)

(2) x_start = 6, learning rate = 0.8 時，epochs =20, 畫出圖形(15%)

程式碼 (x_start = 0.5, learning rate = 0.3 時，epochs =20)

```
import numpy as np
import matplotlib.pyplot as plt
def func(x): return np.cos(x)  #object fun

def dfunc(x): return -np.sin(x)  #dy/dx= -sin(x)

#gradient descent fun
def GD(x_start, df, epochs, lr):
    xs = np.zeros(epochs+1)
    x = x_start
    xs[0] = x
    for i in range(epochs):
        dx = df(x)
        v = - lr * dx
        x += v
        xs[i+1] = x
    return xs
x_start = 0.5
epochs = 20
lr = 0.3       #learning rate
x = GD(x_start, dfunc, epochs, lr=lr)
color = 'r'
from numpy import arange
t = arange(0, 2*np.pi, 0.01)
plt.plot(t, func(t), c='b')
plt.plot(x, func(x), c=color, label='lr={}'.format(lr))
plt.scatter(x, func(x), c=color, )
plt.legend()
  plt.show()
```

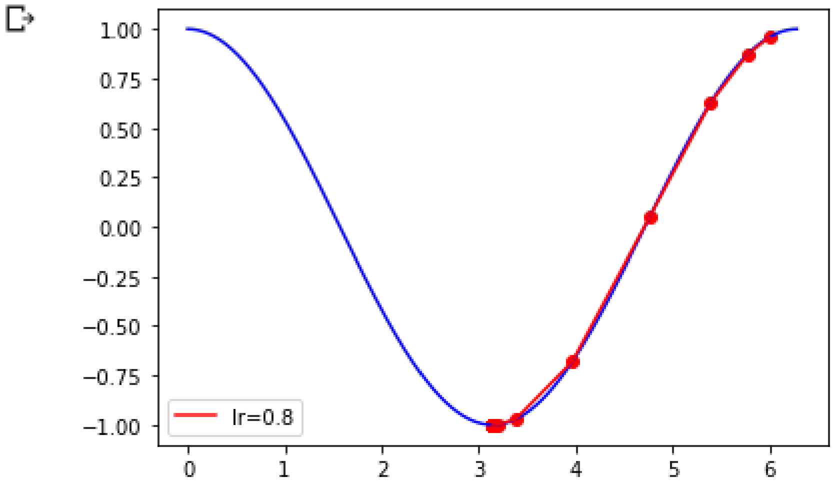
執行結果：



程式碼 (x_start = 6, learning rate = 0.8 時，epochs =20)

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

def func(x): return np.cos(x)  #object fun

def dfunc(x): return -np.sin(x)  #dy/dx= -sin(x)

#gradient descent fun
def GD(x_start, df, epochs, lr):
    xs = np.zeros(epochs+1)
    x = x_start
    xs[0] = x
    for i in range(epochs):
        dx = df(x)
        v = - lr * dx
        x += v
        xs[i+1] = x
    return xs
x_start = 6
epochs = 20
lr = 0.8      #learning rate

x = GD(x_start, dfunc, epochs, lr=lr)

color = 'r'

from numpy import arange
t = arange(0, 2*np.pi, 0.01)
plt.plot(t, func(t), c='b')
plt.plot(x, func(x), c=color, label='lr={}'.format(lr))
plt.scatter(x, func(x), c=color, )
plt.legend()
  plt.show()
```

執行結果：



三、試利用 gradient descent 求解 $f(x) = 0.1x^2 + sin(0.1x^2)$, $-10<x<10$ 的最小值(40%)

$(f'(x) = 0.2x + 0.2xcos(0.1x^2))$

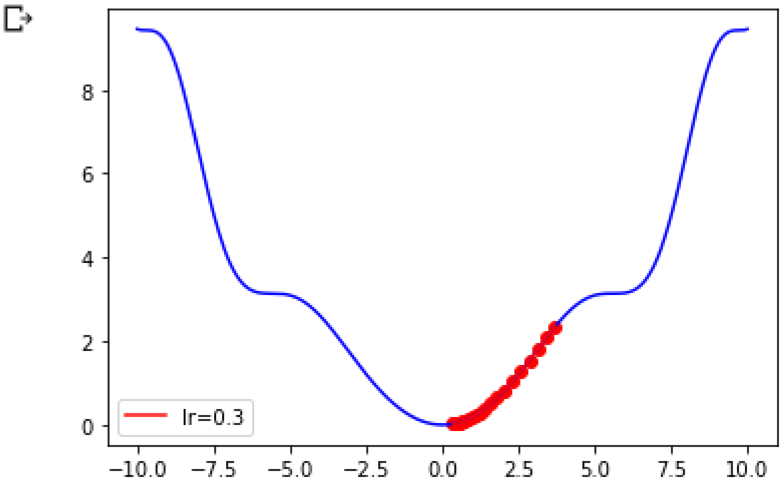
(1) x_start = 3.7, learning rate = 0.3 時，epochs =20，畫出圖形(20%)

(2) x_start = 9, learning rate = 0.3 時，epochs =20，畫出圖形(20%)

程式碼 (x_start = 3.7, learning rate = 0.3 時，epochs =20)

```
import numpy as np
import matplotlib.pyplot as plt
def func(x): return 0.1*x**2 +np.sin(0.1*x**2)  #object fun
def dfunc(x): return 0.2*x + 0.2*x*np.cos(0.1*(x**2))  #dy/dx= 0.2x + cos(0.1x^2)*0.2x

#gradient descent fun
def GD(x_start, df, epochs, lr):
    xs = np.zeros(epochs+1)
    x = x_start
    xs[0] = x
    for i in range(epochs):
        dx = df(x)
        v = - lr * dx
        x += v
        xs[i+1] = x
    return xs
x_start = 3.7
epochs = 20
lr = 0.3      #learning rate
x = GD(x_start, dfunc, epochs, lr=lr)
color = 'r'
from numpy import arange
t = arange(-10.0, 10.0, 0.01)
plt.plot(t, func(t), c='b')
plt.plot(x, func(x), c=color, label='lr={}'.format(lr))
plt.scatter(x, func(x), c=color, )
plt.legend()
  plt.show()
```

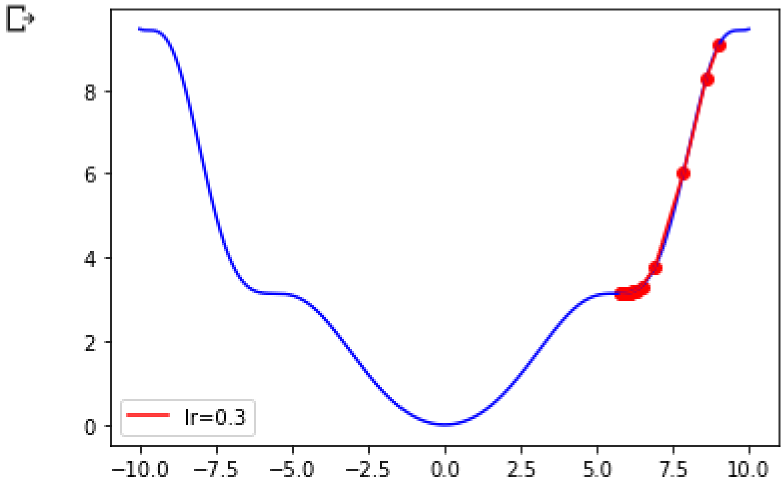
執行結果：



程式碼（x_start = 9, learning rate = 0.3 時，epochs =20)

```
import numpy as np
import matplotlib.pyplot as plt
def func(x): return 0.1*x**2 +np.sin(0.1*x**2)  #object fun
def dfunc(x): return 0.2*x + 0.2*x*np.cos(0.1*(x**2))  #dy/dx= 0.2x + cos(0.1x^2)*0.2x

#gradient descent fun
def GD(x_start, df, epochs, lr):
    xs = np.zeros(epochs+1)
    x = x_start
    xs[0] = x
    for i in range(epochs):
        dx = df(x)
        v = - lr * dx
        x += v
        xs[i+1] = x
    return xs
x_start = 9
epochs = 20
lr = 0.3      #learning rate
x = GD(x_start, dfunc, epochs, lr=lr)
color = 'r'
from numpy import arange
t = arange(-10.0, 10.0, 0.01)
plt.plot(t, func(t), c='b')
plt.plot(x, func(x), c=color, label='lr={}'.format(lr))
plt.scatter(x, func(x), c=color, )
plt.legend()
  plt.show()
```

執行結果：


lr=0.3