Práctica IA Week3

Ccahui Huaman Kristian

April 2019

1 Actividad 1

Plot the data.

```
import matplotlib.pyplot as plt
import csv
import numpy as np
# Cargar Data
def cargarData(file):
    with open(file, newline='') as File:
        reader = csv.reader(File)
        x = []
        y = []

        for row in reader:
            x.append(row[0])
            y.append(row[1])

    x = np.array(x, dtype='float64')  # x = ['1','2'  ...
    y = np.array(y, dtype='float64')  # type: array # x = ['1','2'
...
    return (x,y)

x, y = cargarData('data01.txt')

plt.title('GRAFICA')
plt.plot(x, y, 'rx')
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```

2 Actividad 2

Perform a manual method to find the Thetas that minimizes the cost.

```
import csv
import numpy as np
from mpl_toolkits.mplot3d import axes3d
import matplotlib.pyplot as plt

def costo(teta0, teta1, x, y):
    yy = x * teta1 + teta0  # Funcion y calculando y prima
    costo = (sum((y - yy) ** 2)) / x.size

    return costo

def cargarData(file):
    with open(file, newline='') as File:
        reader = csv.reader(File)
        x = []
        y = []

        for row in reader:
            x.append(row[0])
            y.append(row[1])

    x = np.array(x, dtype='float64')  # x = ['1','2'  ...
    y = np.array(y, dtype='float64')  # type: array # x = ['1','2'
...
    return (x,y)

x, y = cargarData('data01.txt')

# Fin Graficar

x3 = []
y3 = []
z3 = []

#Rango teta0 j
#rango teta1 i

b = 0.1  #Intervalo
for i in np.arange(-1, 4, b):
    for j in np.arange(-10, 10, b):
        c = costo(j, i, x, y)
        z3.append(c)
```

```
        x3.append(j)
        y3.append(i)

fig = plt.figure()

# Agrrgamos un plano 3D
ax = plt.axes(projection='3d')
ax.plot_trisurf(x3, y3, z3, linewidth=0, antialiased=False)
plt.show()
```

3 Actividad 3

Implement the gradient descent algorithm.

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

def cargarData(file):
    with open(file, newline='') as File:
        reader = csv.reader(File)
        x = []
        y = []

        for row in reader:
            x.append(row[0])
            y.append(row[1])

    x = np.array(x, dtype='float64')  # x = ['1','2'  ...
    y = np.array(y, dtype='float64')  # type: array # x = ['1','2'
...
    return (x,y)

def costo(teta0, teta1, x, y):
    yy = x * teta1 + teta0  # Funcion y calculando y prima
    costo = (sum((y - yy) ** 2)) / x.size

    return costo

def calcularY(teta0, teta1, x):
    return teta1*x + teta0

def gradiente(teta0, teta1, alfa, x, y):
```

```
    t0 = teta0
    t1 = teta1
    m = x.size
    for i in range(0, 1500):
        yy = calcularY(t0, t1, x)
        t0 = t0 - (alfa/m)*sum((yy - y))
        t1 = t1 - (alfa/m)*sum((yy - y)*x)

    return (t0, t1)


x, y = cargarData('data01.txt')
teta0, teta1 = gradiente(0, 0, 0.01, x, y)
print(teta0, teta1)

yy = calcularY(teta0, teta1, x)

plt.title('GRAFICA')
plt.plot(x, y, 'rx')
plt.plot(x, yy)
plt.xlabel('x')
plt.ylabel('y')


plt.show()
```

4 Actividad 4

(Optional but considered to evaluation) Plot the Thetas obtained after each itera- tion performed of gradient descent, you can do this into the surface above or use curves (level curves) to plot it.

```
import csv
import numpy as np
from mpl_toolkits.mplot3d import axes3d
import matplotlib.pyplot as plt


def cargarData(file):
    with open(file, newline='') as File:
        reader = csv.reader(File)
        x = []
        y = []

        for row in reader:
```

```
            x.append(row[0])
            y.append(row[1])

    x = np.array(x, dtype='float64')  # x = ['1','2'  ...
    y = np.array(y, dtype='float64')  # type: array # x = ['1','2'
...
    return (x,y)

def costo(teta0, teta1, x, y):
    yy = x * teta1 + teta0  # Funcion y calculando y prima
    costo = (sum((y - yy) ** 2)) / x.size

    return costo

def calcularY(teta0, teta1, x):
    return teta1*x + teta0

def gradiente(teta0, teta1, alfa, x, y):

    t0 = teta0
    t1 = teta1
    m = x.size

    arrayT0 = []
    arrayT1 = []
    arrayC = []
    for i in range(0, 1500):
        yy = calcularY(t0, t1, x)
        t0 = t0 - (alfa/m)*sum((yy - y))
        t1 = t1 - (alfa/m)*sum((yy - y)*x)
        c = costo(t0, t1, x, y)

        arrayT0.append(t0)
        arrayT1.append(t1)
        arrayC.append(c)

    return (t0, t1, arrayT0, arrayT1, arrayC)


x, y = cargarData('data01.txt')
teta0, teta1, xdata, ydata, zdata = gradiente(0, 0, 0.01, x, y)
print(teta0, teta1)

x3 = []
y3 = []
z3 = []
```

```
b = 0.1  #Intervalo
for i in np.arange(0, 2, b):
    for j in np.arange(-4, 0, b):
        c = costo(j, i, x, y)
        z3.append(c)
        x3.append(j)
        y3.append(i)

fig = plt.figure()

# Agrrgamos un plano 3D
ax = plt.axes(projection='3d')
ax.plot_trisurf(x3, y3, z3, linewidth=0, antialiased=False)
ax.scatter(xdata, ydata, zdata, c='r')
plt.show()
```

5 Figuras

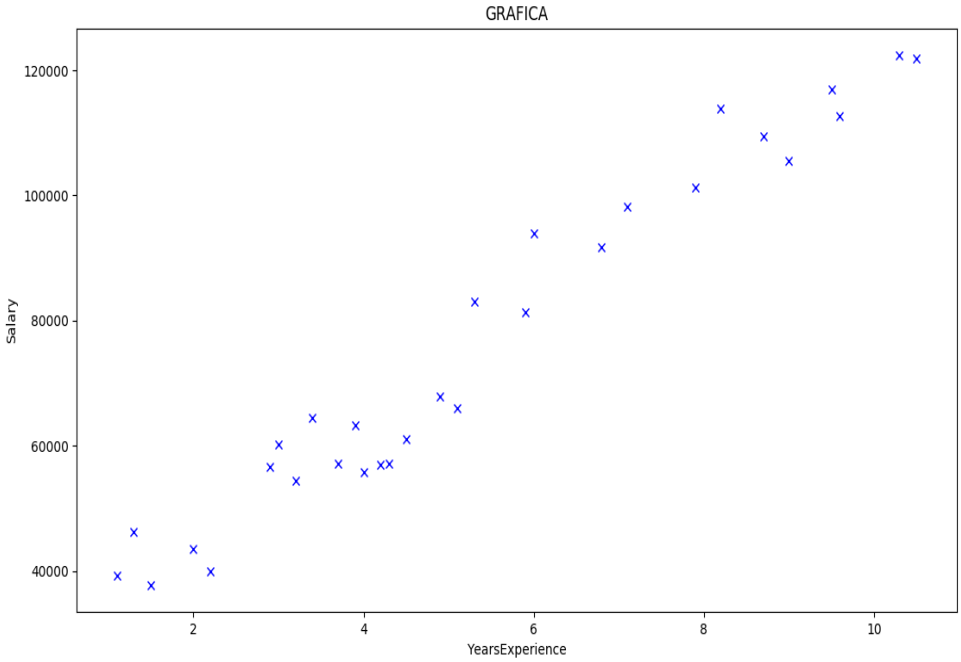

Figure 1: Actividad 1

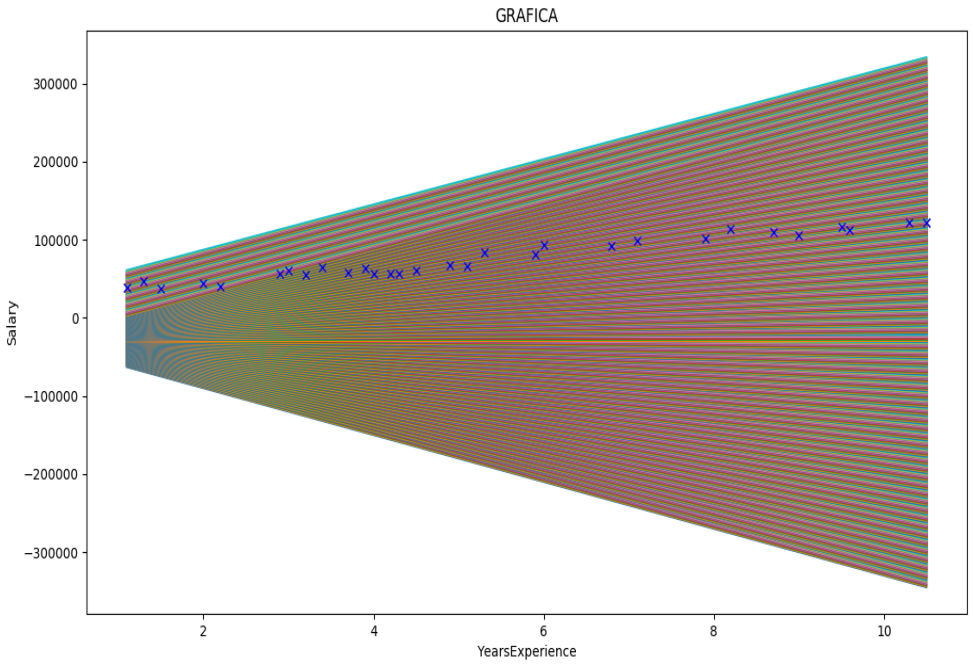

Figure 2: Actividad 2

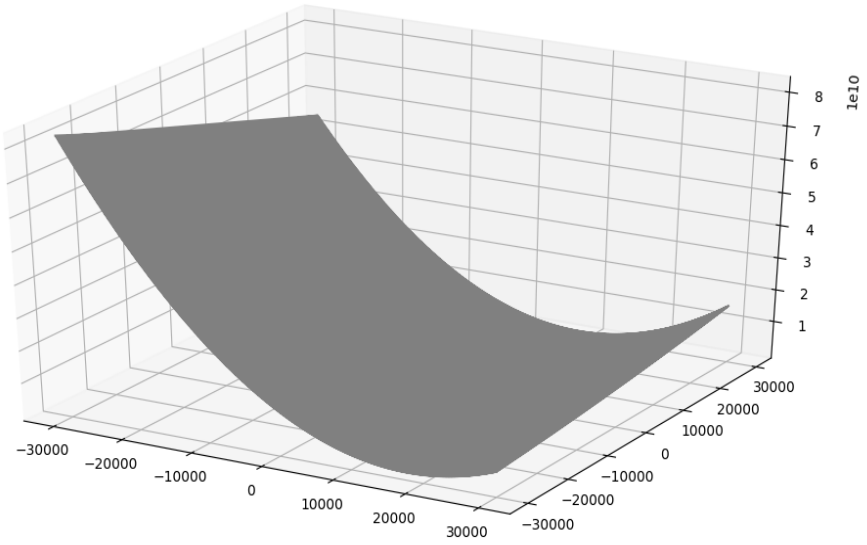

Figure 3: Actividad 3