Pattern and Anomaly Detection Lab

EXP - 10

GitHub LINK - https://github.com/ishikkkkaaaa/UPES/blob/master/Pattern-and-Anomoly-Detection/LAB%2010/main.ipynb

NAME - ISHIKA KESARWANI
ROLL NO. - 92
SAP ID- 500076373

In [1]:
```
import numpy as np
```

In [2]:
```
a1 = np. random. random( (100, 1))
b= 4 + 3*a1 + np. random. randn(100, 1)
```

In [3]:
```
a0 = np. ones ( (100, 1))
A = np. concatenate( (a0, a1), axis = 1)
```

In [4]:
```
temp1 = np.linalg.inv(np.dot (A.T, A))
temp2 = np.dot(temp1,A.T)
w = np. dot(temp2, b)
print("------------------------------")
print("Least squares method(Direct) Single Input")
print("------------------------------")
print("W0",w[0])
print("W1",w[1])
```

```
------------------------------
Least squares method(Direct) Single Input
------------------------------
W0 [4.18949492]
W1 [2.9046634]
```

In [5]:
```
A = np.random.rand (100,1)
y = 4 + 3*A + np.random.randn (100,1)
AWb = np.c_[np.ones ((100,1)), A]
```

```
W = AWb.T.dot (AWb)
W = np. linalg.inv(W)
W= (W.dot (AWb.T)).dot (y)
print (W)


A_ = np.random.rand (100,3)
b1 = 4 + 3*A_ +np.random.randn(100,1)
b2 = 2 + 5*A_ +np.random.randn(100,1)
b3 = 6 + 7*A_ +np.random.randn(100, 1)
b4 = 3 + 6*A_ +np.random.randn(100, 1)
AWb_ = np.c_[np.ones((100,1)), A_]


W_ = AWb_.T.dot (AWb_)
W_ = np.linalg.inv(W_)
W_1 = (W_.dot (AWb_.T)).dot (b1)
W_2 = (W_.dot (AWb_.T)).dot (b2)
W_3 = (W_.dot (AWb_.T)).dot (b3)
W_4 = (W_.dot (AWb_.T)).dot (b4)


print("W1\n", W_1)
print("W2\n", W_2)
print("W3\n", W_3)
print("W4\n", W_4)


np.concatenate((W_1, W_2, W_3, W_4))
```

```python
np.concatenate((W_1, W_2, W_3, W_4))
```

```
[[3.97481659]
 [2.92020388]]
W1
 [[3.5589851  3.5589851  3.5589851 ]
 [3.67060408 0.67060408 0.67060408]
 [0.12270192 3.12270192 0.12270192]
 [0.08738517 0.08738517 3.08738517]]
W2
 [[ 1.72433569  1.72433569  1.72433569]
 [ 5.07241221  0.07241221  0.07241221]
 [ 0.35179835  5.35179835  0.35179835]
 [-0.18515694 -0.18515694  4.81484306]]
W3
 [[ 6.34341116  6.34341116  6.34341116]
 [ 6.12268593 -0.87731407 -0.87731407]
 [ 0.6141725   7.6141725   0.6141725 ]
 [-0.33562452 -0.33562452  6.66437548]]
W4
 [[ 2.54284351  2.54284351  2.54284351]
 [ 6.28114144  0.28114144  0.28114144]
 [-0.10435375  5.89564625 -0.10435375]
 [ 0.59046322  0.59046322  6.59046322]]
```

```
Out[5]: array([[ 3.5589851 ,  3.5589851 ,  3.5589851 ],
       [ 3.67060408,  0.67060408,  0.67060408],
       [ 0.12270192,  3.12270192,  0.12270192],
       [ 0.08738517,  0.08738517,  3.08738517],
       [ 1.72433569,  1.72433569,  1.72433569],
       [ 5.07241221,  0.07241221,  0.07241221],
       [ 0.35179835,  5.35179835,  0.35179835],
       [-0.18515694, -0.18515694,  4.81484306],
       [ 6.34341116,  6.34341116,  6.34341116],
       [ 6.12268593, -0.87731407, -0.87731407],
       [ 0.6141725 ,  7.6141725 ,  0.6141725 ],
       [-0.33562452, -0.33562452,  6.66437548],
       [ 2.54284351,  2.54284351,  2.54284351],
       [ 6.28114144,  0.28114144,  0.28114144],
       [-0.10435375,  5.89564625, -0.10435375],
       [ 0.59046322,  0.59046322,  6.59046322]])
```

<u>Least squares method(Direct)</u>

$$\boxed{\mathbf{w}_{\mathrm{ML}} = \left(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}}$$

np.linalg.inv(x.T.dot(x)

Use np.c_ to concatenate bias

Regularized least squares

$$\boxed{\mathbf{w} = \left(\lambda\mathbf{I} + \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}.}$$

Sequential

$$\boxed{\begin{aligned}\mathbf{w}^{(\tau+1)} &= \mathbf{w}^{(\tau)} - \eta\nabla E_n \\ &= \mathbf{w}^{(\tau)} + \eta(t_n - \mathbf{w}^{(\tau)\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n))\boldsymbol{\phi}(\mathbf{x}_n).\end{aligned}}$$

$$\boxed{\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} + \eta(t_n - \mathbf{w}^{(\tau)\mathrm{T}}\phi_n)\phi_n}$$

Multiple Outputs

$$\boxed{\mathbf{W}_{\mathrm{ML}} = \left(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{T}.}$$