

# 矩陣求導 $y = ||XW + b||_F^2$ ;

In [1]:

```
import torch
torch.manual_seed(0)
x = torch.randn( 10, 4, requires_grad=True)
w = torch.randn( 4, 4, requires_grad=True)
b = torch.randn( 10, 4, requires_grad=True)
print(x)
print(w)
print(b)
y = (x.mm(w) + b).pow(2).sum()
# y.requires_grad_(True)
print(y)

tensor([[ -1.1258, -1.1524, -0.2506, -0.4339],
        [ 0.8487,  0.6920, -0.3160, -2.1152],
        [ 0.3223, -1.2633,  0.3500,  0.3081],
        [ 0.1198,  1.2377,  1.1168, -0.2473],
        [-1.3527, -1.6959,  0.5667,  0.7935],
        [ 0.5988, -1.5551, -0.3414,  1.8530],
        [-0.2159, -0.7425,  0.5627,  0.2596],
        [-0.1740, -0.6787,  0.9383,  0.4889],
        [ 1.2032,  0.0845, -1.2001, -0.0048],
        [-0.5181, -0.3067, -1.5810,  1.7066]], requires_grad=True)
tensor([[ 0.2055, -0.4503, -0.5731, -0.5554],
        [ 0.5943,  1.5419,  0.5073, -0.5910],
        [-1.3253,  0.1886, -0.0691, -0.4949],
        [-1.4959, -0.1938,  0.4455,  1.3253]], requires_grad=True)
tensor([[ 1.5091,  2.0820,  1.7067,  2.3804],
        [-1.1256, -0.3170, -1.0925, -0.0852],
        [ 0.3276, -0.7607, -1.5991,  0.0185],
        [-0.7504,  0.1854,  0.6211,  0.6382],
        [-0.0033, -0.5344,  1.1687,  0.3945],
        [ 1.9415,  0.7915, -0.0203, -0.4372],
        [-0.2188, -2.4351, -0.0729, -0.0340],
        [ 0.9625,  0.3492, -0.9215, -0.0562],
        [-0.6227, -0.4637,  1.9218, -0.4025],
        [ 0.1239,  1.1648,  0.9234,  1.3873]], requires_grad=True)
tensor(162.5566, grad_fn=<SumBackward0>)
```

In [2]:

```
y.backward()
print(x.grad)
print(w.grad)
print(b.grad)

tensor([[ -5.5354,  2.2788, -7.2747,  4.9560],
        [ 7.0786,  7.9225, -3.9229, -20.8719],
        [ 3.7750, -13.6215,  1.8300,  5.0560],
        [-2.5051,  8.0172,  4.6761,  0.5287],
        [-3.8409, -13.8474,  4.4959, 19.6415],
        [-1.1472, -12.0802, -0.4455, 11.5021],
        [ 1.9607, -13.7048,  3.0314,  8.1723],
        [ 0.3539, -5.1480,  3.1660,  5.3047],
        [ 0.5524,  0.1245, -3.4478, -3.5855],
        [-8.1827, -3.6565, -3.3562, 16.4601]])
tensor([[ 12.8560,  0.9793, -10.9075, -24.5453],
        [ 21.0108, 33.3000, -1.3519, -40.4582],
        [-14.7987, -2.2609, -8.6005, -12.0511],
        [-28.4880, -20.5779, 12.3552, 46.0581]])
```

```

tensor([[ 3.1480,  1.6979,  3.1826,  6.4714],
        [ 6.0862,  1.4365, -4.2966, -7.2248],
        [-2.5635, -5.6951, -4.6232,  1.6427],
        [-2.2008,  4.5967,  1.9861, -2.0806],
        [-6.4545, -5.1746,  2.7957,  5.8385],
        [-2.3584, -4.5992, -0.6065,  5.5481],
        [-3.6774, -6.8541, -0.4983,  1.1807],
        [-2.9029, -1.0739, -2.0263,  1.2502],
        [ 2.5451, -2.2012,  2.7118, -1.0662],
        [-1.2450,  0.5925,  3.8684,  9.8011]])

```

## Practice

In [3]:

```

x = torch.tensor([1.], requires_grad=True)
w = torch.tensor([2.], requires_grad=True)
b = torch.tensor([3.], requires_grad=True)
y = (w*x + b).pow(2)
y.backward()
print(w)
print(x)
print(b)
print(w.grad)
print(x.grad)
print(b.grad)

```

```

tensor([2.], requires_grad=True)
tensor([1.], requires_grad=True)
tensor([3.], requires_grad=True)
tensor([10.])
tensor([20.])
tensor([10.])

```

In [4]:

```

y = w*x + b
y.backward()
print(w.grad)
print(x.grad)
print(b.grad)

```

```

tensor([11.])
tensor([22.])
tensor([11.])

```

In [5]:

```

y = w*x + b
w.grad.zero_()
x.grad.zero_()
b.grad.zero_()
y.backward()
print(w.grad)
print(x.grad)
print(b.grad)

```

```

tensor([1.])
tensor([2.])
tensor([1.])

```

In [6]:

```
# Kan Horst
```

In [ ]:

