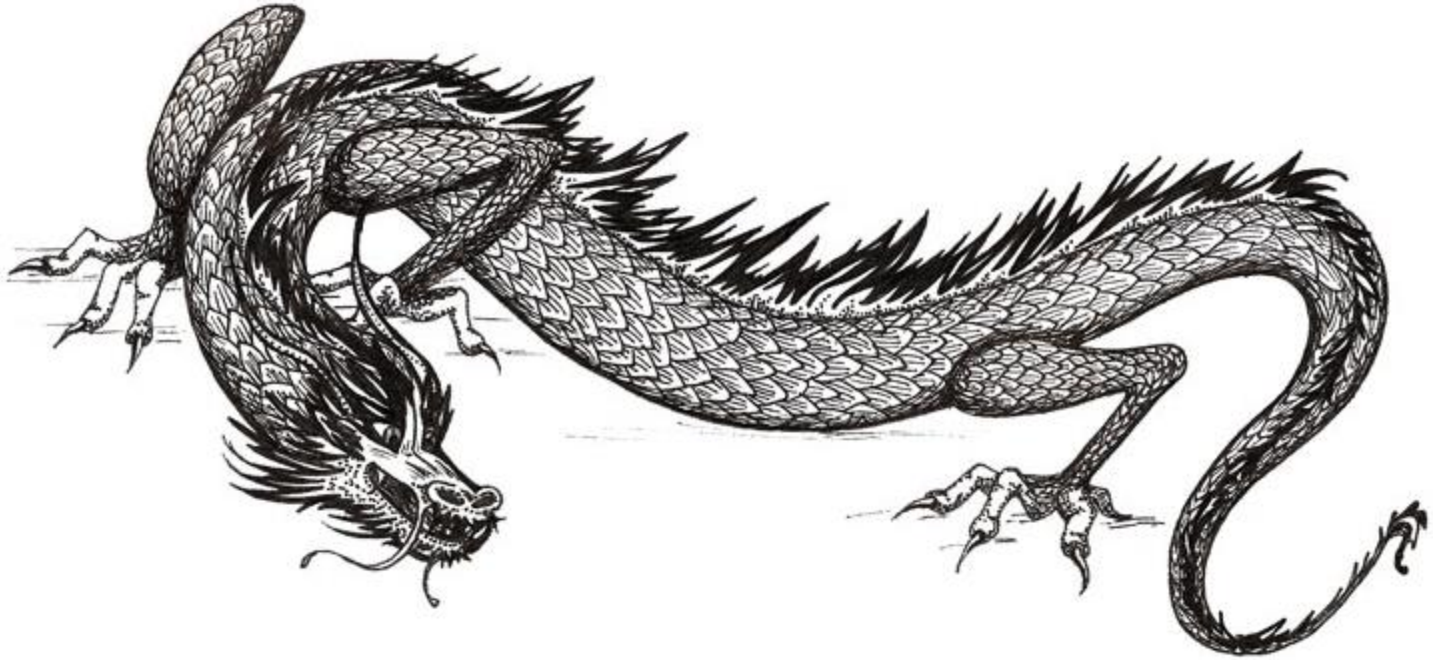
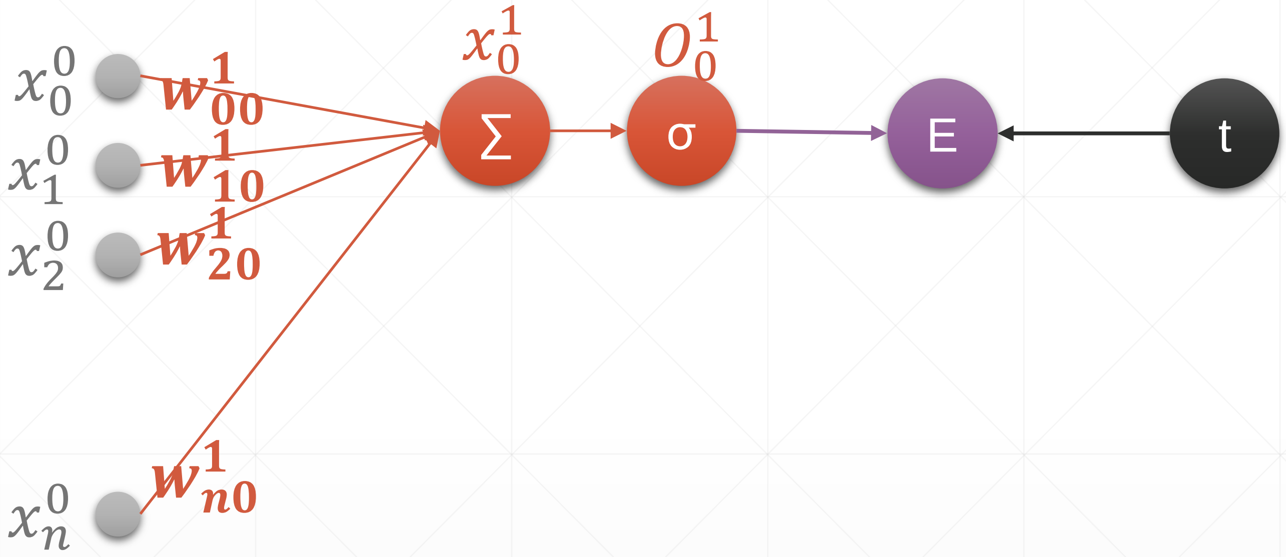


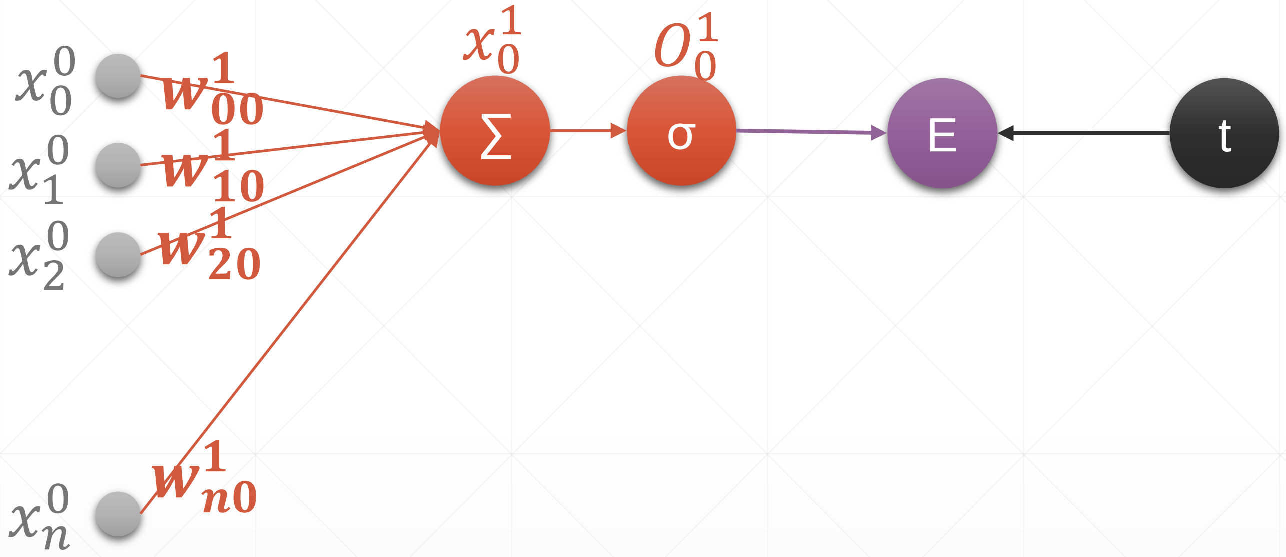
多输出感知机及其梯度

主讲：龙良曲

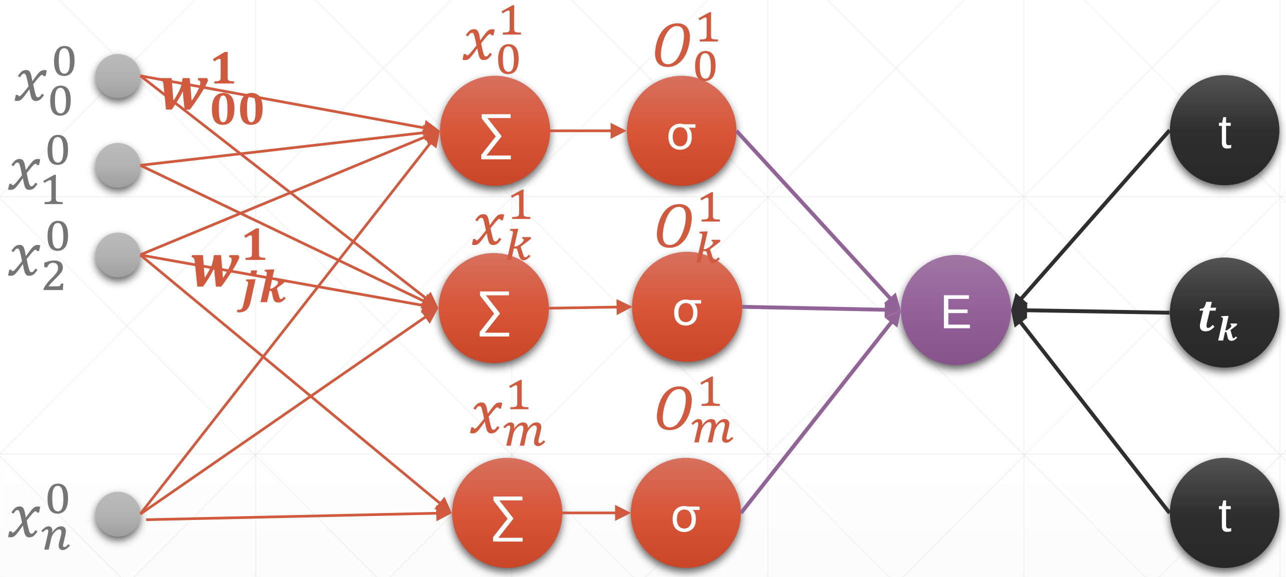
Perceptron



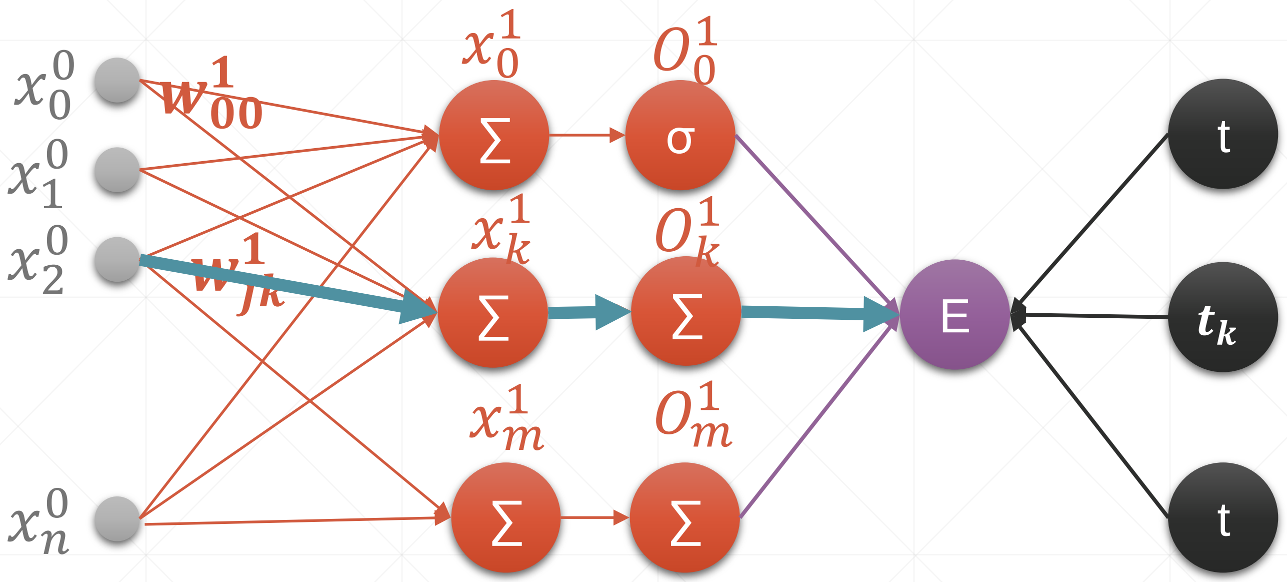
$$\frac{\partial E}{\partial w_{j0}} = (O_0 \ - t)O_0\,(1 - O_0)\ x_j^0$$


x_0^0
x_1^0
x_2^0
x_n^0
w_{00}^1
w_{10}^1
w_{20}^1
w_{n0}^1
x_0^1
O_0^1
Σ
σ
E
t

Multi-output Perceptron



Derivative



$$E = \frac{1}{2}\sum\left(O_i^1 - t_i\right)^2$$

$$\frac{\partial E}{\partial w_{jk}} = \left(O_k - t_k\right)\frac{\partial O_k}{\partial w_{jk}}$$

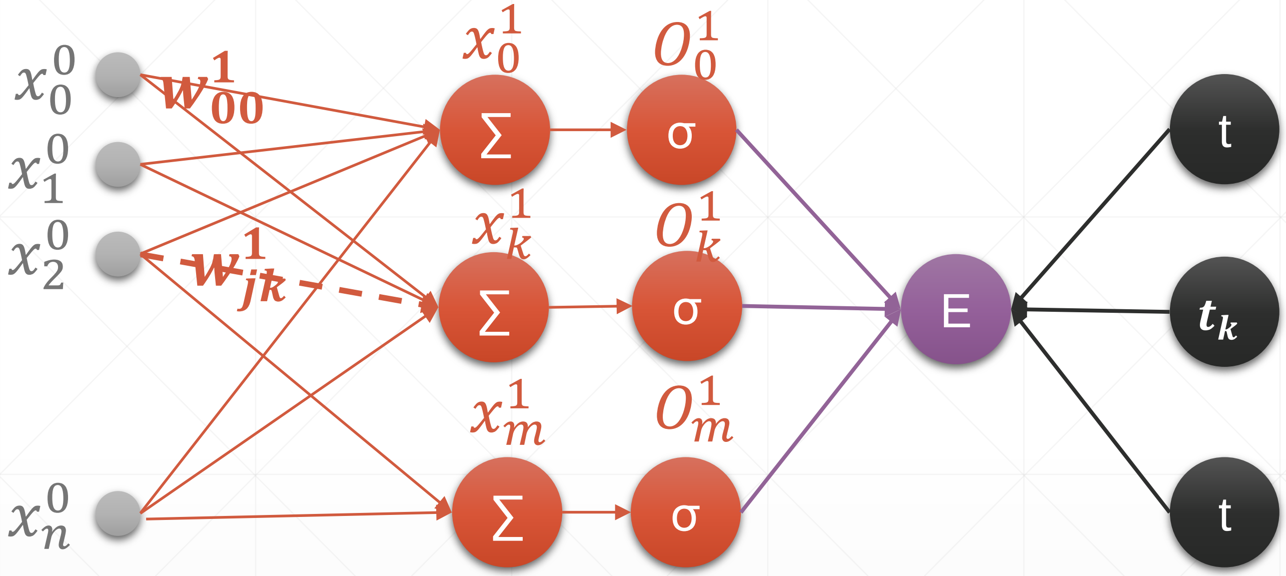
$$\frac{\partial E}{\partial w_{jk}} = \left(O_k - t_k\right)\frac{\partial \sigma(x_k)}{\partial w_{jk}}$$

$$\frac{\partial E}{\partial w_{jk}} = \left(O_k - t_k\right)\sigma(x_k)(1 - \sigma(x_k))\frac{\partial x_k^1}{\partial w_{jk}}$$

$$\frac{\partial E}{\partial w_{jk}} = \left(O_k - t_k\right)\mathrm{O_k}\left(1 - O_k\right)\frac{\partial x_k^1}{\partial w_{jk}}$$

$$\frac{\partial E}{\partial w_{jk}} = \left(O_k - t_k\right)\mathrm{O_k}\left(1 - O_k\right)x_j^0$$

Multi-output Perceptron



$$\frac{\partial E}{\partial w_{jk}} = (O_k - t_k)O_k(1 - O_k)\, x_j^0$$

```
In [3]: x=tf.random.normal([2,4])
In [4]: w=tf.random.normal([4,3])
In [5]: b=tf.zeros([3])
In [6]: y=tf.constant([2,0])

In [9]: with tf.GradientTape() as tape:
   ...:     tape.watch([w,b])
   ...:     prob = tf.nn.softmax(x@w+b, axis=1)
   ...:     loss = tf.reduce_mean(tf.losses.MSE(tf.one_hot(y,depth=3), prob))

In [10]: grads = tape.gradient(loss, [w,b])
In [11]: grads[0]
[[-0.00967887, -0.00335512,  0.01303399],
       [-0.04446869,  0.06194263, -0.01747394],
       [-0.04530644,  0.01043231,  0.03487412],
       [ 0.02006017, -0.03638988,  0.0163297 ]]
In [12]: grads[1] # [-0.02585024,  0.06217915, -0.03632889]
```

下一课时

链式法则

Thank You.