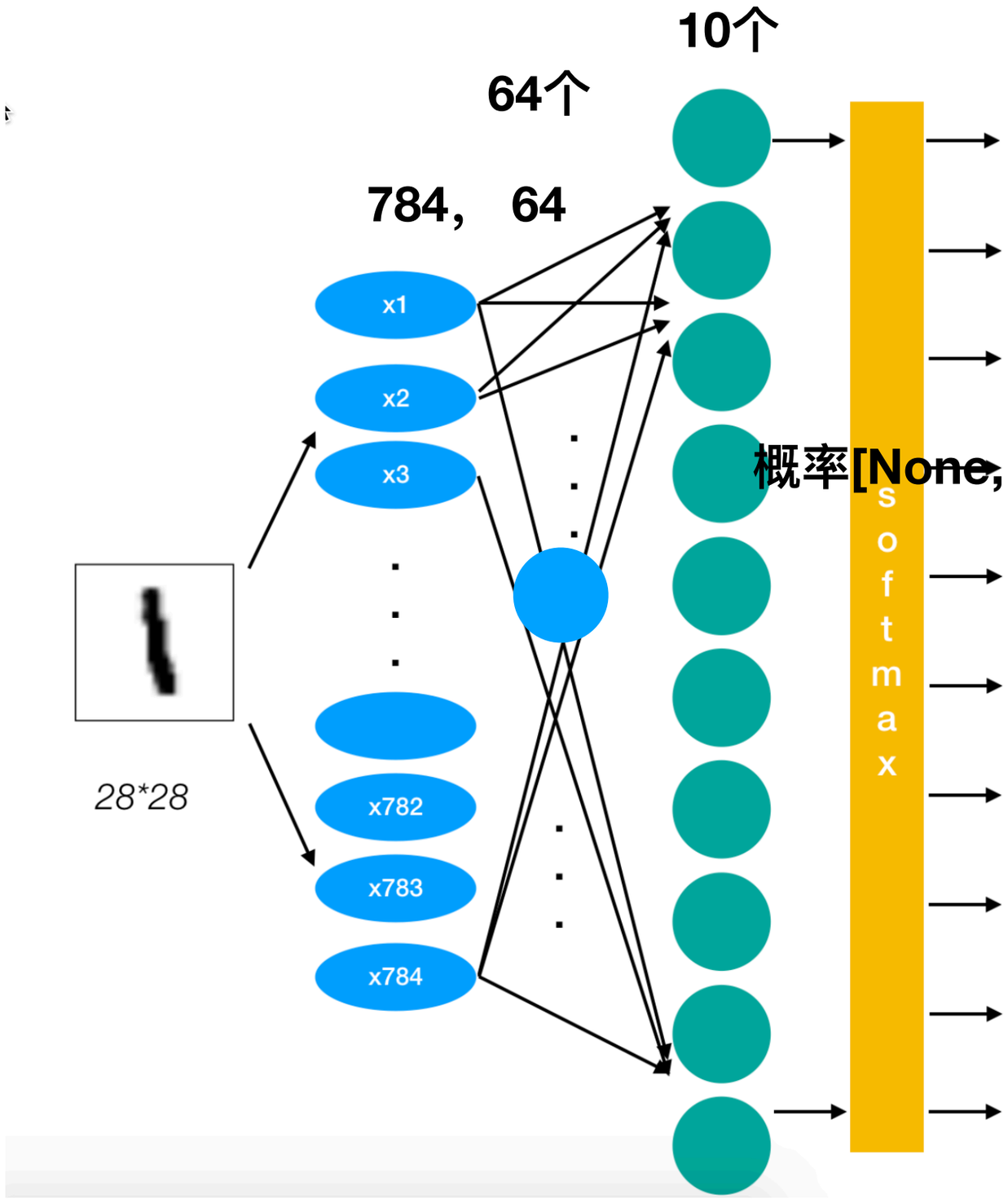


隐层的参数：W [784, 64]，b [64]

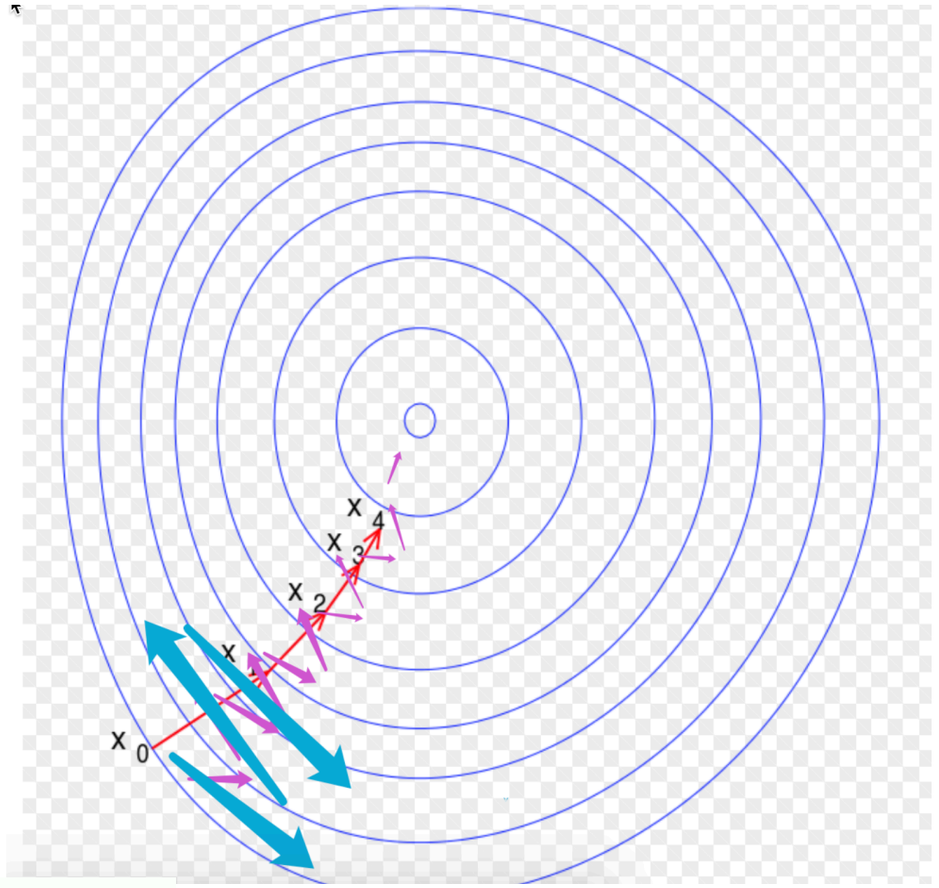
输出层的参数：W [64, 10]，b [10]

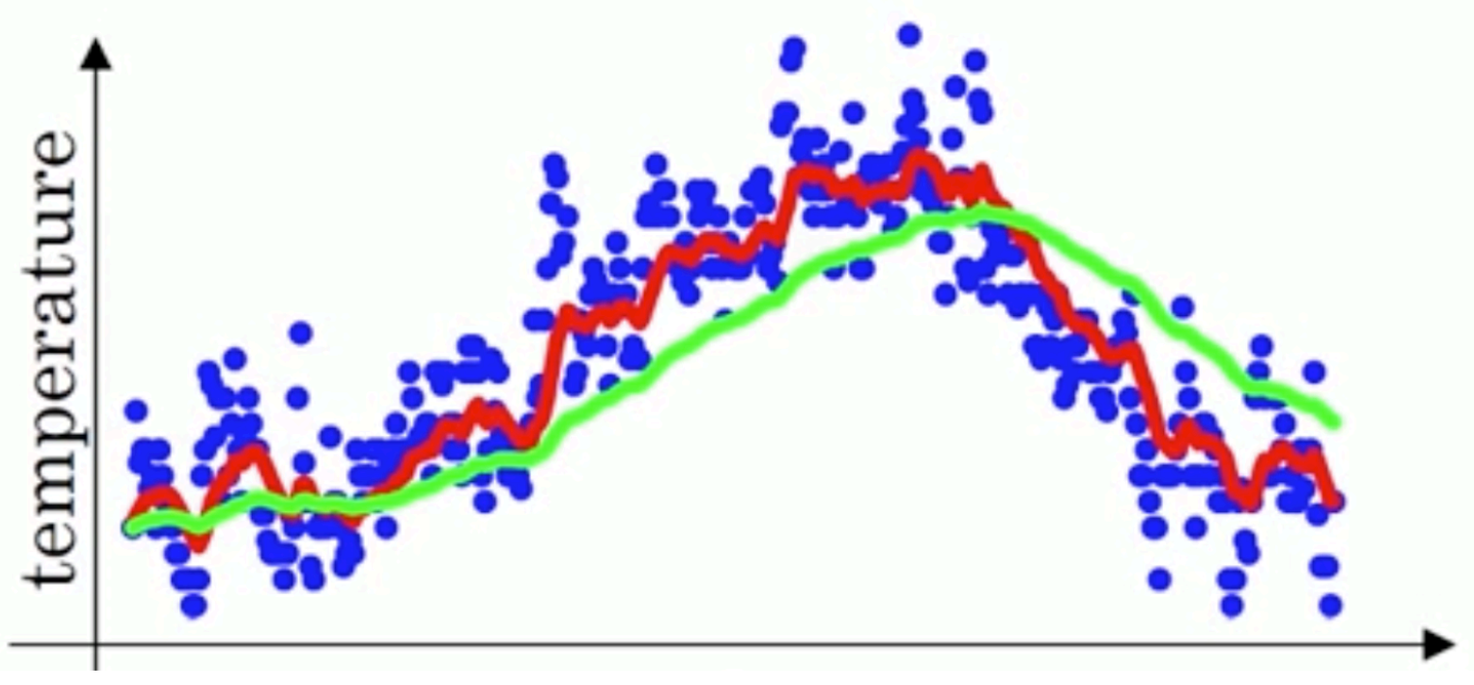
$$S_{dW}[l] = \beta S_{dW}[l] + (1-\beta) dW^{[l]}$$

$$S_{db}[l] = \beta S_{db}[l] + (1-\beta) db^{[l]}$$

$$W^{[l]} := W^{[l]} - \alpha S_{dW}[l]$$

$$b^{[l]} := b^{[l]} - \alpha S_{db}[l]$$





假设用每一个 mini-batch 计算 dW、db，第t次迭代时：

$$v_{dW} = \beta_1 v_{dW} + (1 - \beta_1) dW$$

$$v_{db} = \beta_1 v_{db} + (1 - \beta_1) db$$

动量梯度结果

$$v_{dW[l]}^{corrected} = \frac{v_{dW[l]}}{1 - (\beta_1)^t}$$

修正　目的：稍微增大一些调整后的值

$$s_{dW} = \beta_2 s_{dW} + (1 - \beta_2)(dW)^2$$

$$s_{db} = \beta_2 s_{db} + (1 - \beta_2)(db)^2$$

RMSProp梯度接过

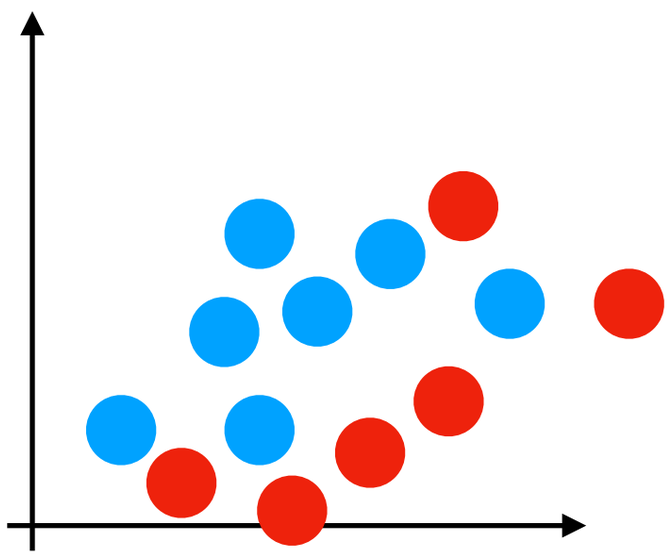
$$s_{dW[l]}^{corrected} = \frac{s_{dW[l]}}{1 - (\beta_1)^t}$$

修正　目的：稍微增大一些调整后的值

其中l为某一层，t为移动平均第次的值

Adam 算法的参数更新：

训练集	测试集		
94%	70%	方差较大	过拟合
75%	78%	偏差较大	欠拟合



$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m} L(\hat{y}^{(i)}, y^{(i)}) + \frac{\lambda}{2m}||w||_2^2$$

dw = dL/dw + lambd/m(W)

W := W - (alpha)dW
= W - (alpha)dL/dw -(alpha)(lambd/m)(W)

```python
# 假设设置神经元保留概率
keep_prob = 0.8
# 随机建立一个标记1 or 0的矩阵，表示随机失活的单元，占比20%
dl = np.random.rand(al.shape[0], al.shape[1]) < keep_prob
# 让a1对应d1的为0地方结果为0
al = np.multiply(al, dl)

# 为了测试的时候，每一个单元都参与进来
al /= keep_prob
```

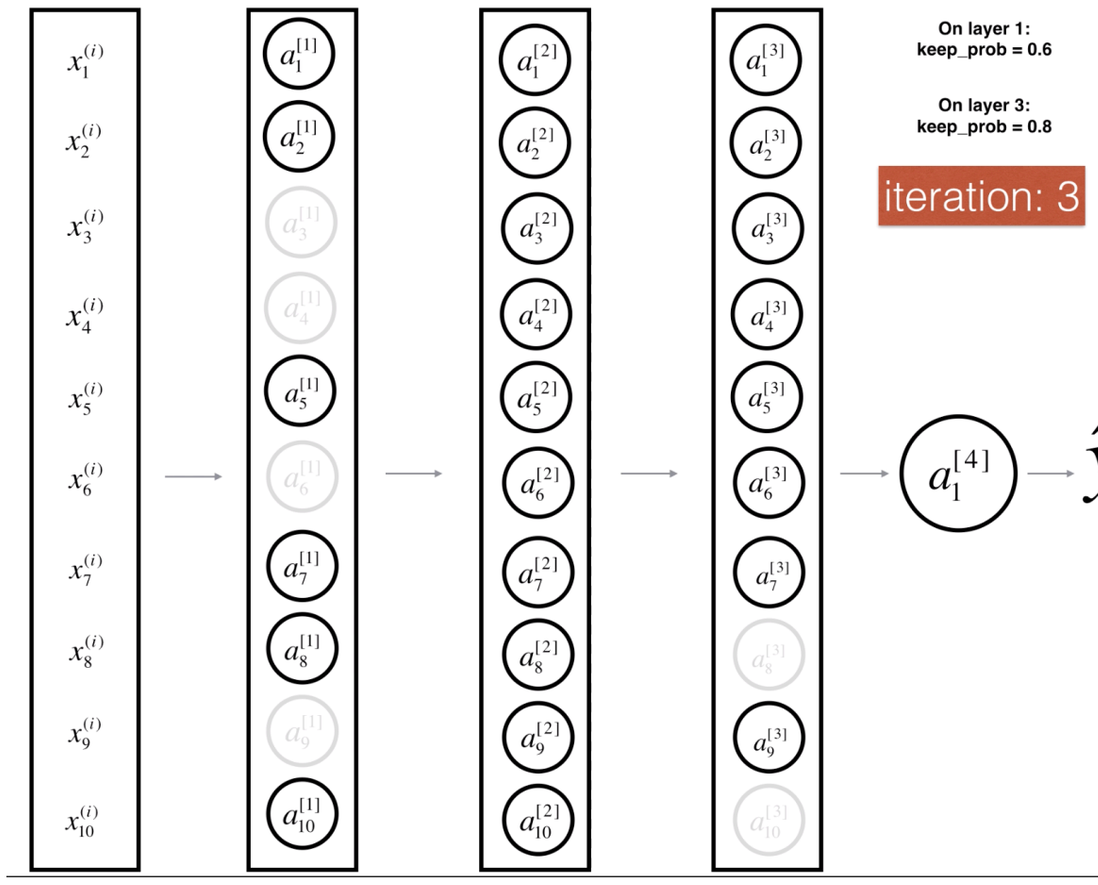
测试期间，平均的量没变

keep_prob = 0.8

第一层10个神经元， 参数：W = 6 * 10

[1, 10]　　[6个1， 4个0]

6个值保留下来，4个值直接没了



训练集的准确率:
Accuracy: 0.938388625924171
测试集的准确率:
Accuracy: 0.93

L2

训练集的准确率:
Accuracy: 0.928099526066351
测试集的准确率:
Accuracy: 0.95

droupout