**np.random.randn(layer\_dims[l], layer\_dims[l-1])**

**np.zeros((layer\_dims[l], 1))**

细节：高斯分布的参数为元组不加括号， zeros、ones等参数为元组加括号。

**nansum(cross\_entropy)**

默认flat方式，也可以指定axis其中0表示行，1表示列，用nansum是因为有时求log可能会出错，这样就用0代替异常值。

**矩阵相乘的若干形式：**

1. np.matmul：矩阵相乘，维度不匹配会报错

2.np.multiply：元素相乘，或者boardcast后元素相乘

3.\*：等价于np.multiply

4.np.dot：矩阵相乘，不会boardcast

总结：

**matmul == dot 而且顺序不可换**

**\* == np.multiply 而且顺序可以换**

**np .ravel() 和np.flatten()**

两者的本质都是想把多维的数组**降为1维**。区别在于numpy.flatten()返回一份拷贝，对数据更改时不会影响原来的数组，而numpy.ravel()则返回视图，对数据更改时会影响原来的数组。

**np.r和np.c**

np.r\_是按列连接两个矩阵，就是把两矩阵上下并置，要求列数相等。

np.c\_是按行连接两个矩阵，就是把两矩阵左右并置，要求行数相等。