

## 第二周测验-神经网络基础

原文: <https://blog.csdn.net/u013733326/article/details/79865858>

1. 神经元节点计算什么?

- A. 【】 神经元节点先计算激活函数, 再计算线性函数( $z=Wx+b$ )
- B. 【】 神经元节点先计算线性函数 ( $z=Wx+b$ ), 再计算激活。
- C. 【】 神经元节点计算函数  $g$ , 函数  $g$  计算( $Wx+b$ )。
- D. 【】 在将输出应用于激活函数之前, 神经元节点计算所有特征的平均值

请注意: 神经元的输出是  $a=g(Wx+b)$ , 其中  $g$  是激活函数 (sigmoid, tanh, ReLU, ...).

编者: 吴在介绍神经网络时, 使用 Logistic Regression 作为引例。先要计算线性函数  $z=W^T x+b$ (此处  $W$ 、 $b$  为参数,  $x$  为样本构成的矩阵), 但是给出的  $z$  可能分布在整个实数区间, 这个时候需要激活函数将  $z$  映射到 0, 1 范围内。Logistic regression 引例中, 吴使用的激活函数为 sigmoid 函数。

2. 下面哪一个是 Logistic 损失?

[点击这里.](#) (编者: 未发现)

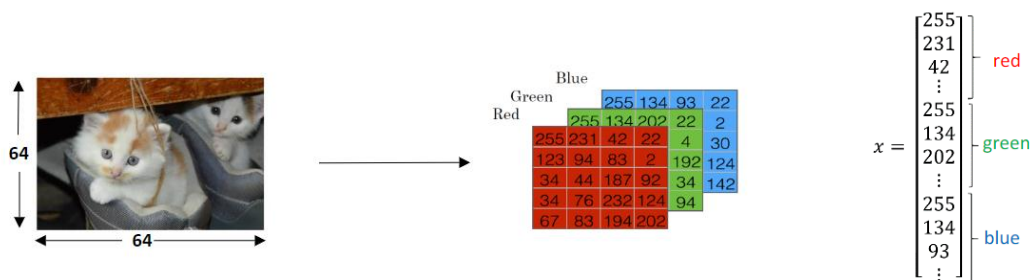
请注意: 我们使用交叉熵损失函数。

3. 假设  $img$  是一个  $(32,32,3)$  数组, 具有 3 个颜色通道: 红色、绿色和蓝色的  $32 \times 32$  像素的图像。如何将其重新转换为列向量?

$x=img.reshape((32*32*3,1))$

编者: 3 个颜色通道, 像素为  $32 \times 32$ , 将每个颜色通道的每个像素依次放在列向量中即可得图像样本。数组->矩阵, 可使用 reshape 方法。

下图为课程讲义。



4.看一下下面的这两个随机数组“a”和“b”:

```
a=np.random.randn(2,3) #a.shape=(2,3)
```

```
b=np.random.randn(2,1) #b.shape=(2,1)
```

```
c=a+b
```

请问数组 c 的维度是多少?

答: B (列向量) 复制 3 次, 以便它可以和 A 的每一列相加, 所以: `c.shape=(2,3)`

编者: a 是 2\*3 的矩阵, b 是 2\*1 的矩阵, 根据 python 中的广播 (broadcasting) 思想, a+b 会得到 2\*3 的矩阵。

5.看一下下面的这两个随机数组“a”和“b”:

```
a=np.random.randn(4,3)#a.shape=(4,3)
```

```
b=np.random.randn(3,2)#b.shape=(3,2)
```

```
c=a*b
```

请问数组“c”的维度是多少?

答: 运算符“\*”说明了按元素乘法来相乘, 但是元素乘法需要两个矩阵之间的维数相同, 所以这将报错, 无法计算。

编者: 似乎正确答案是无法计算, 但是元素乘法与我了解的矩阵乘法有差异。。因为下图所示的 4x3 矩阵乘 3x2 矩阵是成立的。

$$C = AB = \begin{pmatrix} 5 & 2 & 4 \\ 3 & 8 & 2 \\ 6 & 0 & 4 \\ 0 & 1 & 6 \end{pmatrix} \begin{pmatrix} 2 & 4 \\ 1 & 3 \\ 3 & 2 \end{pmatrix} = \begin{pmatrix} 24 & 34 \\ 20 & 40 \\ 24 & 32 \\ 19 & 15 \end{pmatrix}$$

在朱皮特内实验后, “\*” 运算符代表维度相同的矩阵 a、b 内, 相同位置的元素相乘。

```
In [8]: a = np.random.randn(3,2)
        b = np.random.randn(3,2)

In [9]: a
Out[9]: array([[ 1.42600914, -0.25804211],
               [-0.33873856, -0.11304133],
               [-0.45558265, -0.28655791]])

In [10]: b
Out[10]: array([[ 1.13328228, -1.53396585],
               [ 0.62466824, -1.23930366],
               [-0.6283364 ,  0.3730306 ]])

In [11]: a*b
Out[11]: array([[ 1.61607088,  0.39582778],
               [-0.21159922,  0.14009254],
               [ 0.28625917, -0.10689487]])
```

6.假设你的每一个实例有  $n_x$  个输入特征,想一下在  $X=[x^{(1)},x^{(2)}\dots x^{(m)}]$  中, $X$  的维度是多少?

答:  $(n_x,m)$

请注意: 一个比较笨的方法是当  $l=1$  的时候,那么计算一下  $Z^{(l)}=W^{(l)}A^{(l)}Z^{(l)}=W^{(l)}A^{(l)}$ , 所以我们就有:

$$A^{(1)}A^{(1)}=X$$

$$X.shape=(n_x,m)$$

$$Z^{(1)}Z^{(1)}.shape=(n^{(1)}n^{(1)},m)$$

$$W^{(1)}W^{(1)}.shape=(n^{(1)}n^{(1)},n_x)$$

编者:  $x(i)$ 代表一个样本,一个样本(实例)包含  $n_x$  个特征,一共有  $m$  个样本。

那么  $X$  的维度为(特征,样本个数),即是  $(n_x, m)$

7.回想一下,  $np.dot(a, b)$  在  $a$  和  $b$  上执行矩阵乘法,而 ' $a*b$ ' 执行元素方式的乘法。

看一下下面的这两个随机数组 " $a$ " 和 " $b$ ":

```
a=np.random.randn(12288,150)#a.shape=(12288,150)
```

```
b=np.random.randn(150,45)#b.shape=(150,45)
```

```
c=np.dot(a,b)
```

请问  $c$  的维度是多少?

答:  $c.shape=(12288,45)$ ,这是一个简单的矩阵乘法例子。

编者: 首先矩阵乘法  $np.dot(a, b)$ 与元素乘法  $a*b$  是不一样的。其次在  $a-m \times n$ ,  $b-n \times k$  经过矩阵乘法后的到的矩阵  $c$  维度为  $m \times k$

8.看一下下面的这个代码片段:

```
#a.shape=(3,4)
```

```
#b.shape=(4,1)
```

```
for l in range(3):
```

```
    for j in range(4):
```

```
        c[i][j]=a[i][j]+b[j]
```

请问要怎么把它们向量化?

答:  $c=a+b.T$

编者: 先看下这两层循环在干什么。这两层循环要做的就是将  $b$  中的四个元素与  $a$  中的每行元素相加,一共加三层。向量化来看,将  $b$  转置为行元素,与  $a$  相加即可。python 有广播一说。

9.看一下下面的代码:

```
a=np.random.randn(3,3)
```

```
b=np.random.randn(3,1)
```

```
c=a*b
```

请问 c 的维度会是多少?

答: 这将会使用广播机制, b 会被复制三次, 就会变成(3,3), 再使用元素乘法。所以:

```
c.shape=(3,3).
```

编者: 似乎广播机制条件如下: 矩阵格式:(1,1) (1,n) (n,1) n 与进行运算的矩阵有关联(相等?)

10.看一下下面的计算图:

```
J=u+v-w
```

```
=a*b+a*c-(b+c)
```

```
=a*(b+c)-(b+c)
```

```
=(a-1)*(b+c)
```

答:(a-1)\*(b+c)

博主注: 由于弄不到图, 所以很抱歉。

正确答案

1	2	3	4	5
B				
6	7	8	9	10