

1. ¿Qué calcula una neurona?

1 / 1 punto

- ☐ Una neurona calcula una función de activación seguida de una función lineal ($z = Wx + b$)
- ☐ Una neurona calcula la media de todas las características antes de aplicar la salida a una función de activación
- ☒ Una neurona calcula una función lineal ($z = Wx + b$) seguida de una función de activación
- ☐ Una neurona calcula una función g que escala la entrada x linealmente ($Wx + b$)
- ☒ **Correcto**
Correcto, generalmente decimos que la salida de una neurona es $a = g(Wx + b)$ donde g es la función de activación (sigmoide, tanh, ReLU, ...).

2. ¿Cuál de estas es la "Pérdida Logística"?

1 / 1 punto

- ☐ $L^{(yo)}(\hat{y}^{(yo)}, y^{(yo)}) = |y^{(yo)} - \hat{y}^{(yo)}|^2$
- ☒ $L^{(yo)}(\hat{y}^{(yo)}, y^{(yo)}) = -(y^{(yo)} \log g(\hat{y}^{(yo)}) + (1 - y^{(yo)}) \log (1 - \hat{y}^{(yo)}))$
- ☐ $L^{(yo)}(\hat{y}^{(yo)}, y^{(yo)}) = |y^{(yo)} - \hat{y}^{(yo)}|$
- ☐ $L^{(yo)}(\hat{y}^{(yo)}, y^{(yo)}) = \max(0, y^{(yo)} - \hat{y}^{(yo)})$
- ☒ **Correcto**
¡Correcto, esta es la pérdida logística que has visto en la conferencia!

3. Supongamos que `img` es una matriz (32,32,3), que representa una imagen de 32x32 con 3 canales de color rojo, verde y azul. ¿Cómo transformas esto en un vector de columna?

1 / 1 punto

- ☐ `x = img.reforma((1,32*32,*3))`
- ☐ `x = img.reforma((3,32*32))`
- ☒ `x = img.reforma((32*32*3,1))`
- ☐ `x = img.reforma((32*32,3))`

✓ **Correcto**

4.Considere las dos matrices aleatorias siguientes "a" y "b":

1 / 1 punto

```
1 a = np.random.randn(2, 3) # a.shape = (2, 3)
2 b = np.random.randn(2, 1) # b.shape = (2, 1)
3 c = a + b
```

¿Cuál será la forma de "c"?

- ☐ c.forma = (3, 2)
- ☒ c.forma = (2, 3)
- ☐ c.forma = (2, 1)
- ☐ El cálculo no puede ocurrir porque los tamaños no coinciden. Va a ser "Error"!
- ✓ **Correcto**
¡Sí! Esto es radiodifusión. b (vector de columna) se copia 3 veces para que pueda sumarse a cada columna de a.

5.Considere las dos matrices aleatorias siguientes "a" y "b":

1 / 1 punto

```
1 a = np.random.randn(4, 3) # a.shape = (4, 3)
2 b = np.random.randn(3, 2) # b.shape = (3, 2)
3 c = a*b
```

¿Cuál será la forma de "c"?

- ☐ c.forma = (4, 3)
- ☐ c.forma = (4,2)
- ☐ c.forma = (3, 3)

☐ El cálculo no puede ocurrir porque los tamaños no coinciden. Va a ser "Error"!

☒ **Correcto**
¡Por cierto! En numpy, el operador "*" indica la multiplicación por elementos. Es diferente de "np.dot()". Si intentara "c = np.dot(a,b)" obtendría c.shape = (4, 2).

6. Supongamos que tienes n_{X} características de entrada por ejemplo. Recordar que $X = [X^{(1)} X^{(2)} \dots X^{(m)}]$. ¿Cuál es la dimensión de X?

1 / 1 punto

☐ (m , n_{X})

☐ (m , 1)

☒ (n_X , m)

☐ (1 , m)

☒ **Correcto**

7. Recuerde que "np.dot(a,b)" realiza una multiplicación de matrices en a y b, mientras que "a*b" realiza una multiplicación por elementos. 1 / 1 punto

Considere las dos matrices aleatorias siguientes "a" y "b":

```
1 a = np.random.randn(12288, 150) # a.shape = (12288, 150)
2 b = np.random.randn(150, 45) # b.shape = (150, 45)
3 c = np.dot(a,b)
```

¿Cuál es la forma de c?

☐ El cálculo no puede ocurrir porque los tamaños no coinciden. Va a ser "Error"!

☐ c.forma = (150,150)

☐ c.forma = (12288, 150)

☒ c.forma = (12288, 45)

☒ **Correcto**
Correcto, recuerda que un np.dot(a, b) tiene forma (número de filas de a, número de columnas de b). Los tamaños coinciden porque:

"número de columnas de a = 150 = número de filas de b"

8.Considere el siguiente fragmento de código:

1 / 1 punto

```
1 # a.shape = (3,4)
2 # b.shape = (4,1)
3
4 for i in range(3):
5     for j in range(4):
6         c[i][j] = a[i][j] + b[j]
```

¿Cómo se vectoriza esto?

- ☐ c = a + b
- ☒ c = a + bT
- ☐ c = aT + b
- ☐ c = aT + bT
- ☒ **Correcto**

9.Considere el siguiente código:

1 / 1 punto

```
1 a = np.random.randn(3, 3)
2 b = np.random.randn(3, 1)
3 c = a*b
```

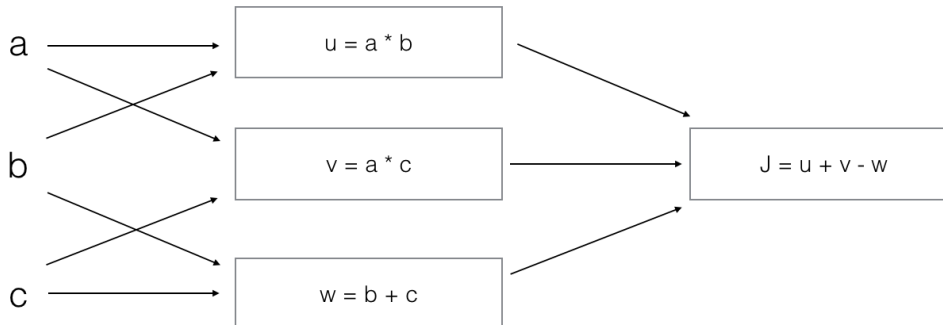
¿Cuál será c? (Si no está seguro, siéntase libre de ejecutar esto en python para averiguarlo).

- ☒ Esto invocará la transmisión, por lo que b se copia tres veces para convertirse en (3,3), y*es un producto de elementos, por lo que c.shape será (3, 3)
- ☐ Esto invocará la transmisión, por lo que b se copia tres veces para convertirse en (3, 3), y*invoca una operación de multiplicación de matrices de dos matrices de 3x3, por lo que c.shape será (3, 3)
- ☐ Esto multiplicará una matriz a de 3x3 con un vector de 3x1, lo que resultará en un vector de 3x1. Es decir, c.forma = (3,1).
- ☐ Conducirá a un error ya que no puede usar "*" para operar en estas dos matrices. En su lugar, debe usar np.dot (a, b)

✓ **Correcto**

10. Considere el siguiente gráfico de cálculo.

1 / 1 punto



¿Cuál es la salida J?

- ☐ $J = (c - 1) * (b + a)$
- ☒ $J = (a - 1) * (b + c)$
- ☐ $J = a * b + b * c + a * c$
- ☐ $J = (b - 1) * (c + a)$

✓ **Correcto**

Sí. $J = u + v - w = a * b + a * c - (b + c) = a * (b + c) - (b + c) = (a - 1) * (b + c)$.