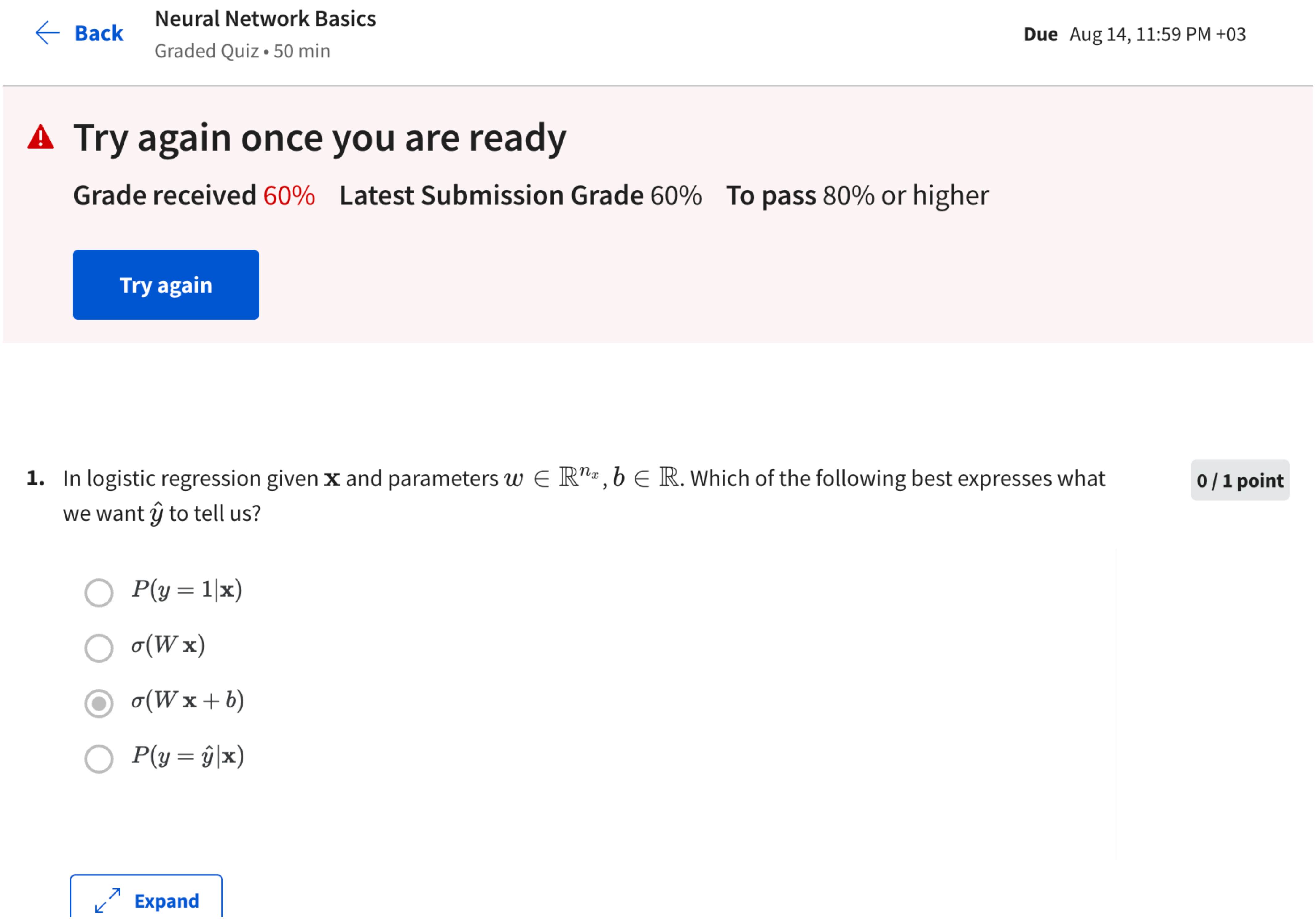


← Back

Neural Network Basics

Graded Quiz • 50 min

Due Aug 14, 11:59 PM +03

Try again once you are ready

Grade received 60% **Latest Submission Grade** 60% **To pass** 80% or higher

Try again

1. In logistic regression given $\mathbf{x}$ and parameters $w \in \mathbb{R}^{n_x}, b \in \mathbb{R}$. Which of the following best expresses what we want $\hat{y}$ to tell us?

0 / 1 point

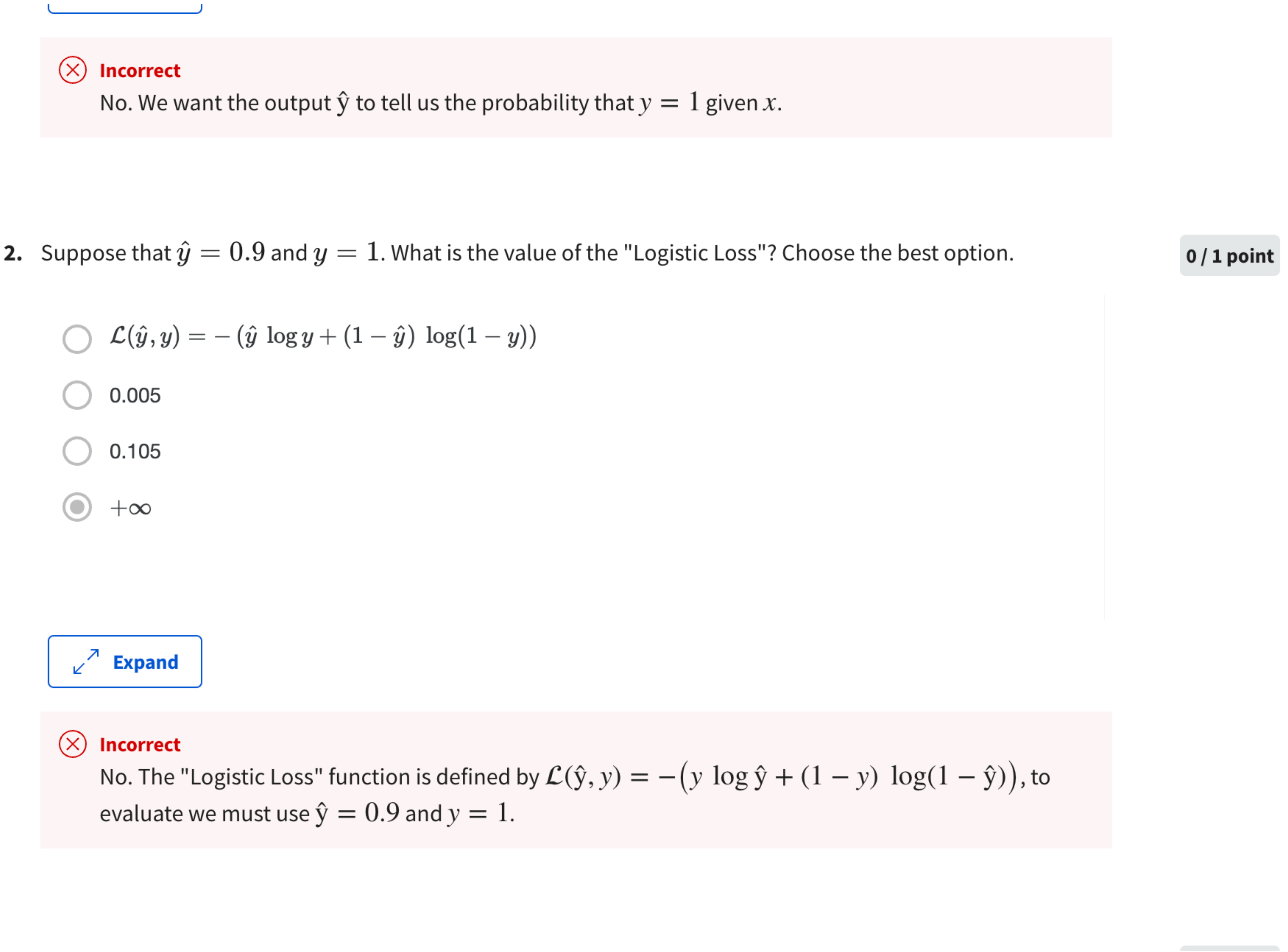
○ $P(y = 1|\mathbf{x})$

○ $\sigma(W\,\mathbf{x})$

● $\sigma(W\,\mathbf{x} + b)$

○ $P(y = \hat{y}|\mathbf{x})$

Expand



☒ **Incorrect**

No. We want the output ŷ to tell us the probability that $y = 1$ given x.

2. Suppose that $\hat{y} = 0.9$ and $y = 1$. What is the value of the "Logistic Loss"? Choose the best option.

0 / 1 point

○ $\mathcal{L}(\hat{y}, y) = -\left(\hat{y}\ \log y + (1-\hat{y})\ \log(1-y)\right)$

○ 0.005

○ 0.105

● $+\infty$

☒ **Incorrect**

No. The "Logistic Loss" function is defined by $\mathcal{L}(\hat{y}, y) = -\big(y\ \log \hat{y} + (1-y)\ \log(1-\hat{y})\big)$, to evaluate we must use $\hat{y} = 0.9$ and $y = 1$.

3. Suppose img is a (32,32,3) array, representing a 32x32 image with 3 color channels red, green and blue. How do you reshape this into a column vector x?

0 / 1 point

○ x = img.reshape((32*32,3))

● x = img.reshape((3,32*32))

○ x = img.reshape((1,32*32,3))

○ x = img.reshape((32*32*3,1))

⤢ **Expand**

⊗ **Incorrect**

4. Consider the following random arrays a and b, and c:

1 / 1 point

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

$$b = np.random.randn(1,4) \,\#\, b.shape = (1,4)$$

$$c = a + b$$

What will be the shape of c?

○ c.shape = (3, 1)

○ The computation cannot happen because it is not possible to broadcast more than one dimension.

○ c.shape = (1, 4)

● c.shape = (3, 4)

Expand

✓ **Correct**

Yes. Broadcasting is used, so row b is copied 3 times so it can be summed to each row of a.

5. Consider the two following random arrays a and b:

1 / 1 point

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

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

$$c = a * b$$

What will be the shape of c?

○ c.shape = (4, 3)

○ c.shape = (3, 3)

○ c.shape = (4,2)

● The computation cannot happen because the sizes don't match. It's going to be "Error"!

Expand

✓ **Correct**
Indeed! In numpy the "*" operator indicates element-wise multiplication. It is different from "np.dot()". If you would try "c = np.dot(a,b)" you would get c.shape = (4, 2).

6. Suppose you have n_x input features per example. Recall that $X = [x^{(1)} x^{(2)} ... x^{(m)}]$. What is the dimension of X?

1 / 1 point

○ (m, n_x)

○ $(1, m)$

○ $(m, 1)$

● (n_x, m)

Expand

Correct

7. Recall that $np.dot(a, b)$ performs a matrix multiplication on a and b, whereas $a * b$ performs an element-wise multiplication.

1 / 1 point

Consider the two following random arrays a and b:

$a = np.random.randn(12288, 150)$

$a.shape = (12288, 150)$

$b = np.random.randn(150, 45)$

$b.shape = (150, 45)$

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

What is the shape of c?

○ c.shape = (150,150)

○ The computation cannot happen because the sizes don't match. It's going to be "Error"!

○ c.shape = (12288, 150)

● c.shape = (12288, 45)

Expand

Correct

Correct, remember that a np.dot(a, b) has shape (number of rows of a, number of columns of b). The sizes match because: "number of columns of a = 150 = number of rows of b"

8. Consider the following code snippet:

1 / 1 point

$$a.shape = (3, 4)$$

$$b.shape = (4, 1)$$

```
for i in range(3):

 for j in range(4):

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

How do you vectorize this?

● c = a + b.T

○ c = a.T + b.T

○ c = a + b

○ c = a.T + b

Expand

✓ Correct

9. Consider the following code: **1 / 1 point**

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

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

$$c = a * b$$

What will be c? (If you're not sure, feel free to run this in python to find out).

● This will invoke broadcasting, so b is copied three times to become (3,3), and * is an

● This will invoke broadcasting, so b is copied three times to become (3,3), and ∗ is an element-wise product so c.shape will be (3, 3)

○ It will lead to an error since you cannot use “*” to operate on these two matrices. You need to instead use np.dot(a,b)

○ This will multiply a 3x3 matrix a with a 3x1 vector, thus resulting in a 3x1 vector. That is, c.shape = (3,1).

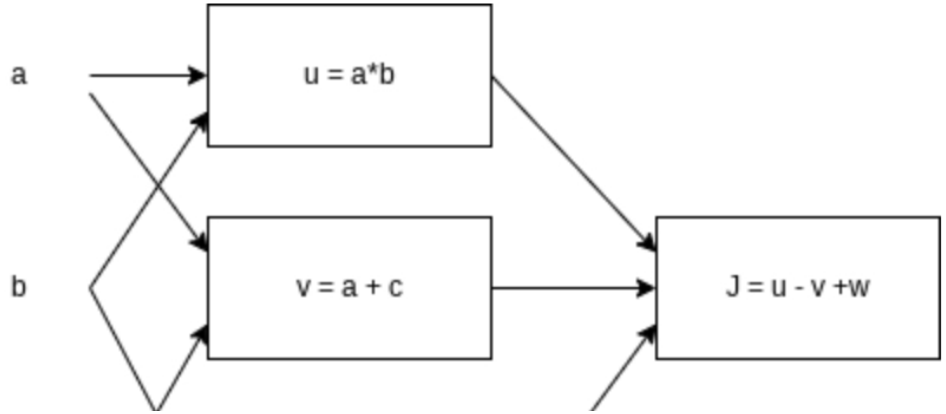
○ This will invoke broadcasting, so b is copied three times to become (3, 3), and ∗ invokes a matrix multiplication operation of two 3x3 matrices so c.shape will be (3, 3)

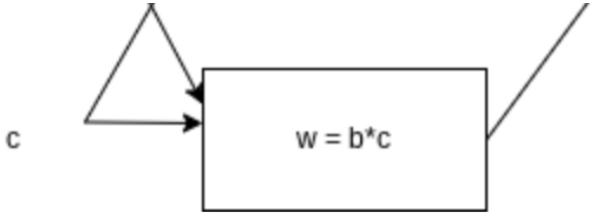
Expand

Correct

10. Consider the following computational graph.

0 / 1 point





What is the output of J?

○ $ab + bc + ac$

● $(a - 1), (b + c)$

○ $(a + c), (b - 1)$

○ $(c - 1), (a + c)$

Expand

Incorrect

No.

$$J = u - v + w = ab - (a + c) + bc = ab - a + bc - c = a\,(b - 1) + c\,(b - 1) = (a + c)\,(b - 1)$$