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

- $P(y = \hat{y}|\mathbf{x})$
- $\sigma(W\,\mathbf{x})$
- $\sigma(W\,\mathbf{x} + b$
- $P(y = 1|\mathbf{x})$

2.Which of these is the "Logistic Loss"? 1 point

- $\mathrm{L}^{(i)}(\hat{y}^{(i)}, y^{(i)}) = -(y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}))$
- $\mathrm{L}^{(i)}(\hat{y}^{(i)}, y^{(i)}) = \mid y^{(i)} - \hat{y}^{(i)} \mid$
- $\mathrm{L}^{(i)}(\hat{y}^{(i)}, y^{(i)}) = max(0, y^{(i)} - \hat{y}^{(i)})$
- $\mathrm{L}^{(i)}(\hat{y}^{(i)}, y^{(i)}) = \mid y^{(i)} - \hat{y}^{(i)} \mid^2$

3.Consider the Numpy array x: 1 point

$x = np.array([[[1], [2]], [[3], [4]]])$

What is the shape of x?

- (4,)
- (2,2,1)
- (1, 2, 2)
- (2, 2)

4.Consider the following random arrays a and b, and c: 1 point

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

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

$c = a + b$

What will be the shape of c?

- The computation cannot happen because the sizes don't match. It's going to be "Error"!
- c.shape = (2, 3)
- c.shape = (2, 1)
- c.shape = (3, 2)

5.Consider the two following random arrays a and b: 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 = (4,2)
- The computation cannot happen because the sizes don't match. It's going to be "Error"!
- c.shape = (4, 3)
- c.shape = (3, 3)

1 point

6.Suppose you have n_x input features per example. If we decide to use row vectors $\mathbf{x}_j$ for the features and $X = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_m \end{bmatrix}$.

What is the dimension of X?

- (n_x, n_x)
- (m, n_x)
- (n_x, m)
- $(1, n_x)$

1 point

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

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?

- The computation cannot happen because the sizes don't match. It's going to be "Error"!
- c.shape = (150,150)
- c.shape = (12288, 150)
- c.shape = (12288, 45)

1 point

8.Consider the following code snippet:

$a.shape = (3, 4)$

$b.shape = (4, 1)$

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.T*b
- c = np.dot(a,b)
- c = a*b.T
- c = a*b

9.Consider the following code: 1 point

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

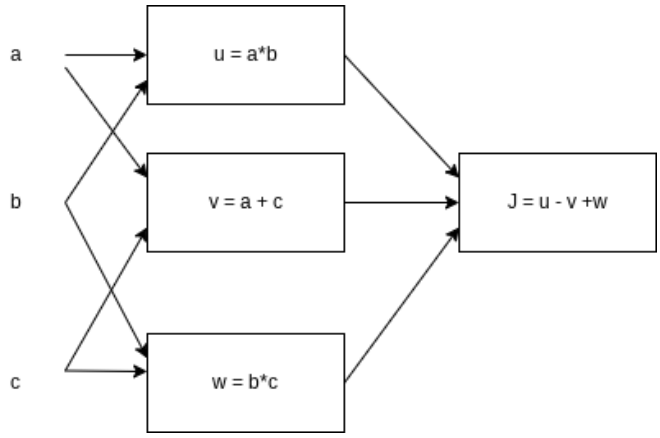
$b = np.random.randn(3, 1)$

$c = a * b$

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

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

10.Consider the following computational graph. 1 point



What is the output of J?

- $(c - 1)(a + c)$
- $ab + bc + ac$
- $(a + c)(b - 1)$
- $(a - 1)(b + c)$