Your grade: 100%

Your latest: **100%** • Your highest: **100%** • To pass you need at least 80%. We keep your highest score.

Next item →

1. What does a neuron compute?

1 / 1 point

● A neuron computes a linear function $z = Wx + b$ followed by an activation function

○ A neuron computes a function g that scales the input x linearly (Wx + b)

○ A neuron computes the mean of all features before applying the output to an activation function

○ A neuron computes an activation function followed by a linear function $z = Wx + b$

✓ **Correct**
Correct, we generally say that the output of a neuron is a = g(Wx + b) where g is the activation function (sigmoid, tanh, ReLU, ...).

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

1 / 1 point

○ $\mathcal{L}^{(i)}(\hat{y}^{(i)}, y^{(i)}) = \mid y^{(i)} - \hat{y}^{(i)} \mid^2$

○ $\mathcal{L}^{(i)}(\hat{y}^{(i)}, y^{(i)}) = \mid y^{(i)} - \hat{y}^{(i)} \mid$

○ $\mathcal{L}^{(i)}(\hat{y}^{(i)}, y^{(i)}) = max(0, y^{(i)} - \hat{y}^{(i)})$

● $\mathcal{L}^{(i)}(\hat{y}^{(i)}, y^{(i)}) = -(y^{(i)}\log(\hat{y}^{(i)}) + (1 - y^{(i)})\log(1 - \hat{y}^{(i)}))$

✓ **Correct**
Correct, this is the logistic loss you've seen in lecture!

3. Consider the Numpy array x:

1 / 1 point

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

What is the shape of x?

○ (2, 2)

○ (1, 2, 2)

○ (4,)

● (2,2,1)

✓ **Correct**
Yes. This array has two rows and in each row it has 2 arrays of 1x1.

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

1 / 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?

○ c.shape = (2, 1)

● c.shape = (2, 3)

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

○ c.shape = (3, 2)

✓ **Correct**
Yes! This is broadcasting. b (column vector) is copied 3 times so that it can be summed to each column 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?

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

○ c.shape = (4,2)

○ c.shape = (3, 3)

○ c.shape = (4, 3)

✓ **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. 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?

○ $(1, n_x)$

○ (n_x, n_x)

● (m, n_x)

○ (n_x, m)

Correct

Yes. Each $\mathbf{x}_j$ has dimension $1 \times n_x$, X is built stacking all rows together into a $m \times n_x$ array.

7. Consider the following array:

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

What is the result of $a * a$?

● $\begin{pmatrix} 4 & 1 \\ 1 & 9 \end{pmatrix}$

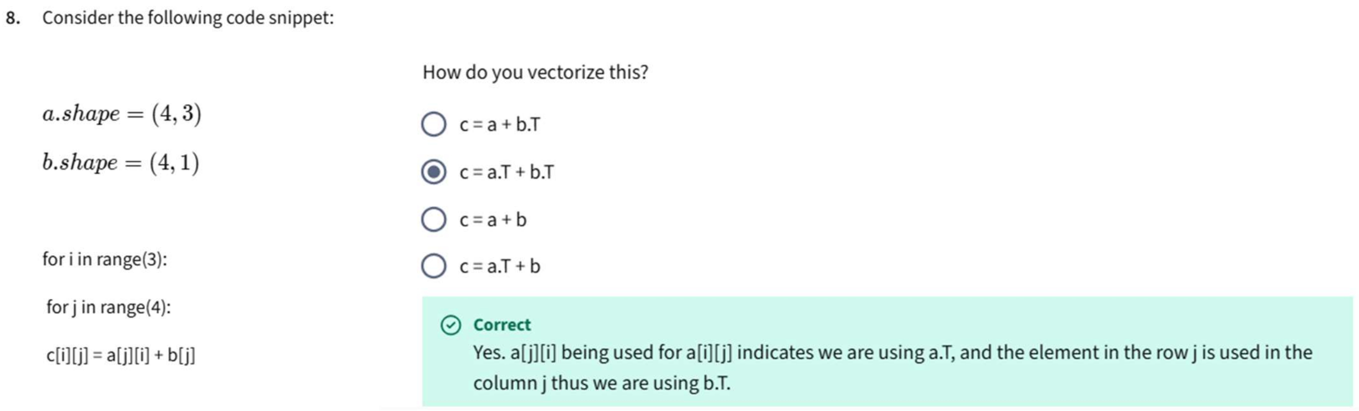
○ The computation cannot happen because the sizes don't match. It's going to be an "Error"!

○ $\begin{pmatrix} 5 & 5 \\ 5 & 10 \end{pmatrix}$

○ $\begin{pmatrix} 4 & 2 \\ 2 & 6 \end{pmatrix}$

Correct

Yes, recall that * indicates element-wise multiplication.



8. Consider the following code snippet:

$a.shape = (4,3)$

$b.shape = (4,1)$

```
for i in range(3):
 for j in range(4):
 c[i][j] = a[j][i] + b[j]
```

How do you vectorize this?

○ c = a + b.T

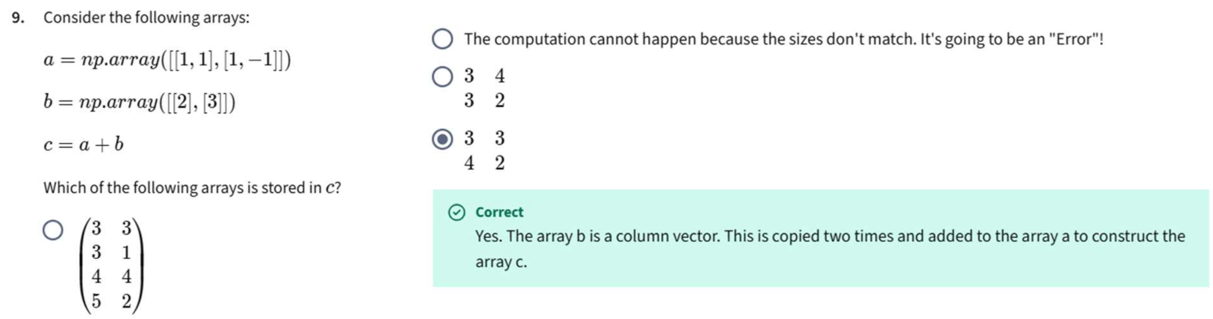
● c = a.T + b.T

○ c = a + b

○ c = a.T + b

Correct

Yes. a[j][i] being used for a[i][j] indicates we are using a.T, and the element in the row j is used in the column j thus we are using b.T.



9. Consider the following arrays:

$a = np.array([[1,1],[1,-1]])$

$b = np.array([[2],[3]])$

$c = a + b$

Which of the following arrays is stored in c?

○ $\begin{pmatrix} 3 & 3 \\ 3 & 1 \\ 4 & 4 \\ 5 & 2 \end{pmatrix}$

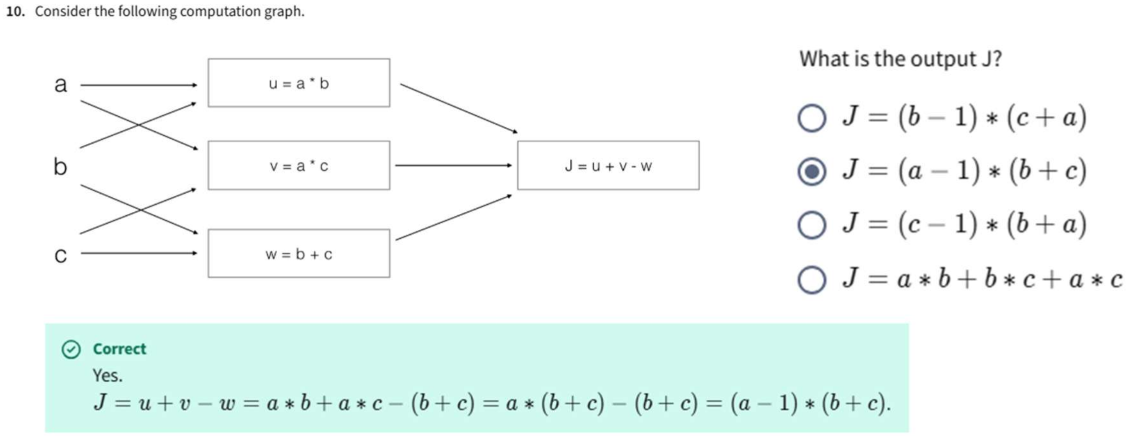
○ The computation cannot happen because the sizes don't match. It's going to be an "Error"!

○ $\begin{matrix} 3 & 4 \\ 3 & 2 \end{matrix}$

● $\begin{matrix} 3 & 3 \\ 4 & 2 \end{matrix}$

Correct

Yes. The array b is a column vector. This is copied two times and added to the array a to construct the array c.



10. Consider the following computation graph.

What is the output J?

○ $J = (b-1)*(c+a)$

● $J = (a-1)*(b+c)$

○ $J = (c-1)*(b+a)$

○ $J = a*b + b*c + a*c$

Correct

Yes.

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