Congratulations! You passed!

Grade received 80% | Latest Submission Grade 80% | To pass 80% or higher

Go to next item

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**

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

Expand

Incorrect
No. Remember that we are interested in the probability that $y = 1$.

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

- 0.105
- 0.005
- $+\infty$
- $\mathcal{L}(\hat{y}, y) = -(\hat{y}\ \log y + (1 - \hat{y})\ \log(1 - y))$

Loading [MathJax]/jax/output/CommonHTML/jax.js

Expand

Correct
Yes. Since $\mathcal{L}(\hat{y}, y) = -(y\ \log \hat{y} + (1 - y)\ \log(1 - \hat{y}))$, for the given values we get $\mathcal{L}(\hat{y}, y) = -(1\ \log 0.9 + 0\ \log 0.1)$

3. Consider the Numpy array x: **1 / 1 point**

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

What is the shape of x?

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

Expand

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 = (3, 2)

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

Expand

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: **0 / 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 = (3, 3)
- c.shape = (4,2)
- c.shape = (4, 3)
- The computation cannot happen because the sizes don't match. It's going to be "Error"!

Expand

Incorrect
No! 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). Also, the broadcasting cannot happen because of the shape of b. b should have been something like (4, 1) or (1, 3) to broadcast properly. So a*b leads to an error!

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

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

- c.shape = (12288, 150)

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**

$a.shape = (4, 3)$

$b.shape = (4, 1)$

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.T + b
- c = a + b.T
- c = a.T + b.T
- c = a + b

Expand

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: **1 / 1 point**

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

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

$c = a + b$

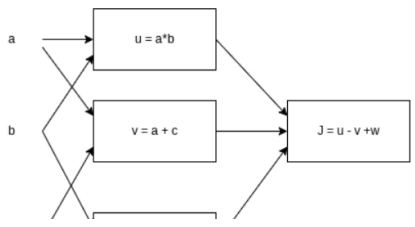
Which of the following arrays is stored in c?

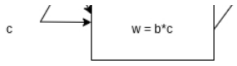
- $\begin{pmatrix} 3 & 3 \\ 3 & 1 \\ 4 & 4 \\ 5 & 2 \end{pmatrix}$
- The computation cannot happen because the sizes don't match. It's going to be an "Error"!
- $3 \quad 4$

Expand

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 computational graph. **1 / 1 point**





What is the output of J?

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

Expand

Correct

Yes.

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