Congratulations! You passed!

Grade received 90% **Latest Submission Grade** 90% **To pass** 80% or higher

Go to next item

1. In logistic regression given the input $\mathbf{x}$, and parameters $w \in \mathbb{R}^{n_x}, b \in \mathbb{R}$, how do we generate the output $\hat{y}$? **1 / 1 point**

- $W\mathbf{x} + b$
- $\sigma(W\mathbf{x} + b)$.
- $\sigma(W\mathbf{x})$
- $\tanh(W\mathbf{x} + b)$

Expand

Correct
Right, in logistic regression we use a linear function $W\mathbf{x} + b$ followed by the sigmoid function σ, to get an output y, referred to as ŷ, such that $0 < \hat{y} < 1$.

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**

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

Expand

Incorrect
No. This is not the definition of the Logistic Loss function.

3. Suppose x is a (8, 1) array. Which of the following is a valid reshape? **1 / 1 point**

- x.reshape(1, 4, 3)
- x.reshape(2, 4, 4)
- x.reshape(2, 2, 2)
- x.reshape(-1, 3)

Expand

Correct
Yes. This generates uses 2*2*2 = 8 entries.

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, 4)
- ( ) c.shape = (1, 4)
- ( ) c.shape = (3, 1)
- ( ) The computation cannot happen because it is not possible to broadcast more than one dimension.

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. **1 / 1 point**

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

Expand

Correct
Yes. Each $\mathbf{x}_i$ 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: **1 / 1 point**

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

What is the result of $np.dot(a, a)$?

- $\begin{pmatrix} 4 & 2 \\ 2 & 6 \end{pmatrix}$
- The computation cannot happen because the sizes don't match. It's going to be an "Error"!
- $\begin{pmatrix} 5 & 5 \\ 5 & 10 \end{pmatrix}$
- $\begin{pmatrix} 4 & 1 \\ 1 & 9 \end{pmatrix}$

Expand

Correct

Yes, recall that * indicates the element wise multiplication and that np.dot() is the matrix multiplication. Thus $\begin{pmatrix} (2)(2)+(1)(1) & (2)(1)+(1)(3) \\ (1)(2)+(3)(1) & (1)(1)+(3)(3) \end{pmatrix}$.

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

Expand

Correct

Yes. b.T gives a column vector with shape (1, 4). The result of c is equivalent to broadcasting a*b.T.

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

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

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

$c = a * b$

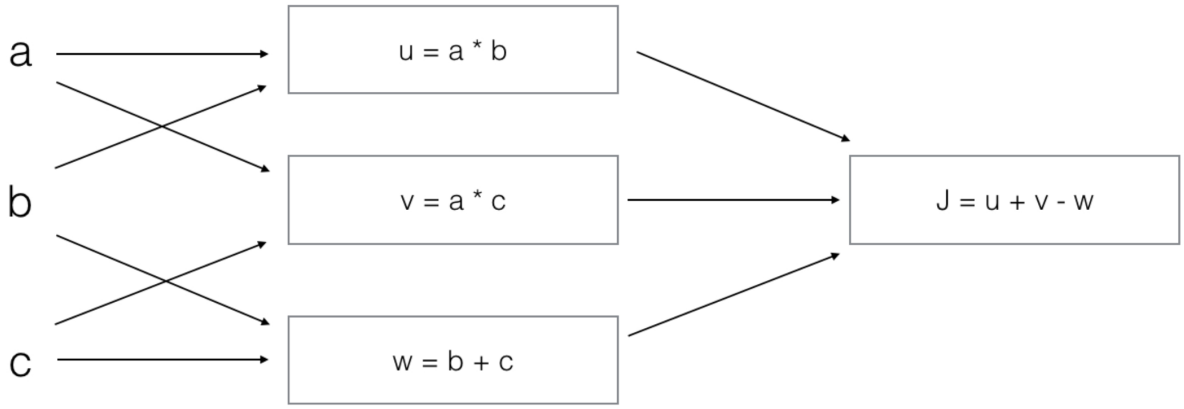
What will be c? (If you're not sure, feel free to run this in python to find out).

- This will 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 $*$ 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 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 computation graph. **1 / 1 point**



What is the output J?

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

Expand

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