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. Which of these is the "Logistic Loss"?

1 / 1 point

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

Expand

Correct

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

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?

1 / 1 point

- x = img.reshape((32*32*3,1))
- x = img.reshape((3,32*32))

an output y, referred to as ŷ, such that $0 < \hat{y} < 1$.

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

1 / 1 point

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

Expand

Correct

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

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?

1 / 1 point

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

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:

1 / 1 point

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

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

$c = a * b$

What will be the shape of c?

- ○ The computation cannot happen because it is not possible to broadcast more than one dimension.
- ○ The computation cannot happen because the sizes don't match
- ○ The computation cannot happen because it is not possible to broadcast more than one dimension.
- ○ The computation cannot happen because the sizes don't match.
- ◉ c.shape = (4, 3)
- ○ c.shape = (1, 3)

Expand

Correct

Yes. Broadcasting is invoked, so row b is multiplied element-wise with each row of a to create c.

6.

0 / 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?

- ○ (m, n_x)
- ◉ $(1, n_x)$

Expand

Incorrect

No. 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:

1 / 1 point

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

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"!
- ○ $\begin{pmatrix} 4 & 2 \\ 2 & 6 \end{pmatrix}$
- ○ $\begin{pmatrix} 5 & 5 \end{pmatrix}$

Expand

Correct

Yes, recall that * indicates element-wise multiplication.

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]

- ( ) c = a.T + b.T
- (•) c = a.T + b.T

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?

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

Which of the following arrays is stored in c?

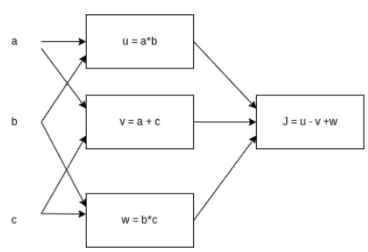
- ( ) The computation cannot happen because the sizes don't match. It's going to be an "Error"!
- (•) $\begin{matrix} 3 & 3 \\ 4 & 2 \end{matrix}$
- ( ) $\begin{matrix} 3 & 4 \\ 3 & 2 \end{matrix}$

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?

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

Expand

Correct

Yes.

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