Congratulations! You passed!

Grade received 100% Latest Submission Grade 100% 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**

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

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**

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

Expand

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?

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

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

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

$c = a + b$

What will be the shape of c?

- c.shape = (2, 1)
- c.shape = (2, 3, 3)
- The computation cannot happen because it is not possible to broadcast more than one dimension
- c.shape = (3,3)

Expand

Correct
Yes. It is not possible to broadcast together a and b. In this case there is no way to generate copies of one of the arrays to match the size of the other.

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 = (3, 3)
- c.shape = (4, 3)

- c.shape = (4,2)

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. 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**

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

Expand

Correct

7. Consider the following array: **1 / 1 point**

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

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

- $\begin{pmatrix} 4 & 1 \\ 1 & 9 \end{pmatrix}$
- 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 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 = np.dot(a,b)
- c = a*b
- c = a*b.T
- c = a.T*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).

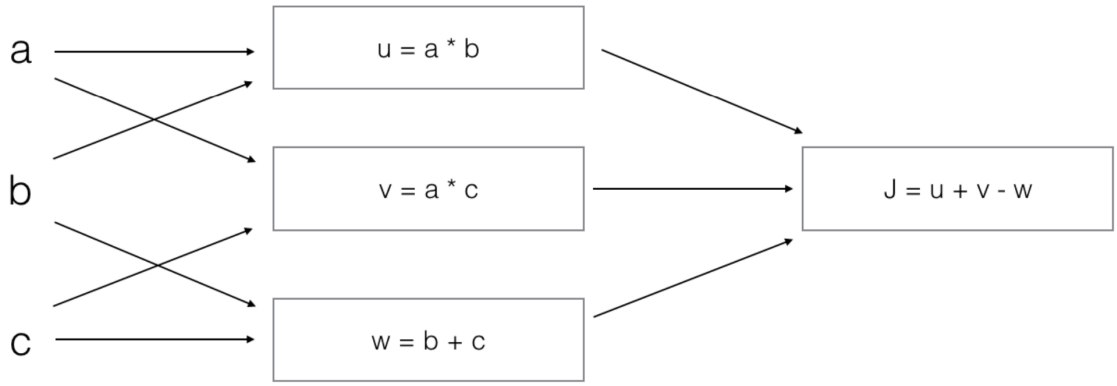
- This will invoke broadcasting, so b is copied three times to become (3,3), and $*$ is an element-wise product so c.shape will be (3, 3)
- This will multiply a 3x3 matrix a with a 3x1 vector, thus resulting in a 3x1 vector. That is, c.shape = (3,1).
- This will invoke broadcasting, so b is copied three times to become (3, 3), and $*$ invokes a matrix multiplication operation of two 3x3 matrices 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)

Expand

Correct

10. Consider the following computation graph.

1 / 1 point



What is the output J?

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

Expand

Correct

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