Which of these is the "Logistic Loss"?

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

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

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

Expand

Correct

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

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

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

$b = np.random.randn(1, 4)$ # $b.shape = (1, 4)$

$c = a + b$

What will be the shape of c?

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

Consider the two following random arrays a and b:

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

$b = np.random.randn(3, 2)$ # $b.shape = (3, 2)$

$c = a * b$

What will be the shape of c?

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

Suppose our input batch consists of 8 grayscale images, each of dimension 8x8. We reshape these images into feature column vectors $\mathbf{x}^j$. Remember that $X = \left[\mathbf{x}^{(1)}\mathbf{x}^{(2)} \cdots \mathbf{x}^{(8)}\right]$. What is the dimension of X?

- ○ (8, 64)
- ◉ (64, 8)
- ○ (512, 1)
- ○ (8, 8, 8)

Expand

Correct

Yes. After converting the 8x8 gray scale images to a column vector we get a vector of size 64, thus X has dimension $(64, 8)$.

Consider the following array:

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

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

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

Expand

Correct

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

Suppose you have n_x input features per example. Recall that X=[x^(1), x^(2)...x^(m)]. What is the dimension of X?

`(n_x, m)`

Note: A stupid way to validate this is use the formula Z^(l) = W^(l)A^(l) when l = 1, then we have

- A^(1) = X
- X.shape = (n_x, m)
- Z^(1).shape = (n^(1), m)
- W^(1).shape = (n^(1), n_x)

Consider the following code snippet:

$$a.shape = (3, 4)$$

$$b.shape = (4, 1)$$

for i in range(3):

for j in range(4):

c[i][j] = a[i][j]*b[j]

How do you vectorize this?

How do you vectorize this?

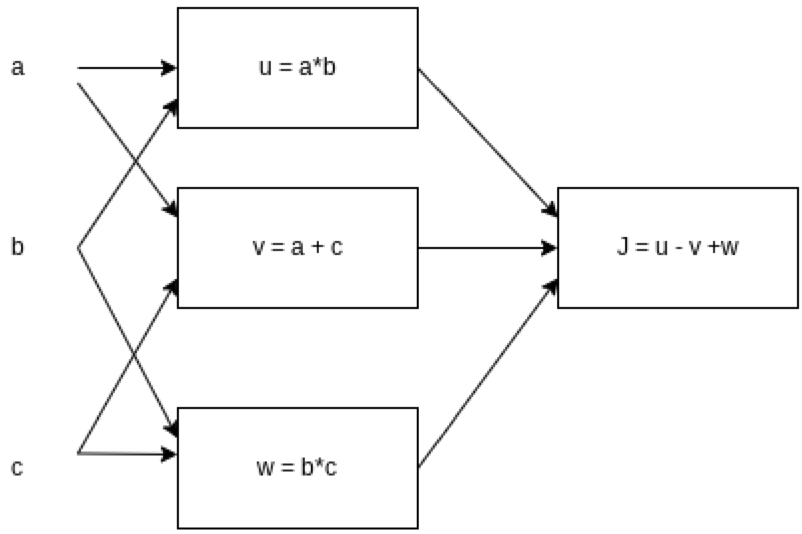
- c = a*b.T
- c = np.dot(a,b)
- c = a*b
- c = a.T*b

Expand

Correct

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

Consider the following computational graph.



What is the output of J?

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

Correct

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