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

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

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 $\hat{y}$, such that $0 < \hat{y} < 1$.

2. Which of these is the "Logistic Loss"? 1 / 1 point

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

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

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

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

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

$c = a + b$

What will be the shape of c?

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

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

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

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

$c = a * b$

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.
- (selected) c.shape = (3, 3)
- c.shape = (1, 3)

Expand

Correct
Yes. Broadcasting allows row a to be multiplied element-wise with each row of b to from c.

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

Expand

Correct
Yes. 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 $a * a$?

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

Expand

Correct
Yes, recall that * indicates element-wise multiplication.

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

Expand

Correct

9. Consider the following arrays: 0 / 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?

$\begin{pmatrix} 3 & 3 \\ 4 & 4 \\ 5 & 2 \end{pmatrix}$

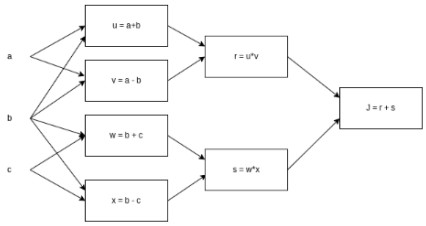
$\begin{matrix} 3 & 3 \\ 4 & 2 \end{matrix}$

$\begin{matrix} 3 & 4 \\ 3 & 2 \end{matrix}$

Expand

Incorrect
No. The array b is a column vector. This is copied two times and added to the array a to construct the array c.

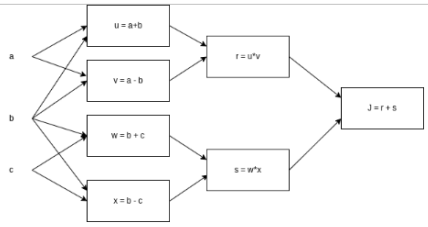
10. Consider the following computational graph. 1 / 1 point



What is the output of J?

- $a^2 + b^2 - c^2$
- $a^2 - b^2$
- $a^2 - c^2$
- $(a - b) * (a - c)$

Expand



What is the output of J?

- $a^2 + b^2 - c^2$
- $a^2 - b^2$
- $a^2 - c^2$
- $(a - b) * (a - c)$

Expand

Correct

Yes.