Your grade: 90%

Your latest: **90%** • Your highest: **90%**

To pass you need at least 80%. We keep your highest score.

Next item →

1. In logistic regression given $\mathbf{x}$ and parameters $w \in \mathbb{R}^{n_x}$, $b \in \mathbb{R}$. Which of the following best expresses what we want $\hat{y}$ to tell us?

1 / 1 point

- ○ $P(y = \hat{y} | \mathbf{x})$
- ○ $\sigma(W\,\mathbf{x} + b$
- ◉ $P(y = 1 | \mathbf{x})$
- ○ $\sigma(W\,\mathbf{x})$

Correct

Yes. We want the output $\hat{y}$ to tell us the probability that $y = 1$ given $\mathbf{x}$
.

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

1 / 1 point

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

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

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

Correct

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?

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

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

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

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

$c = a * b$

What will be the shape of c?

- ○ c.shape = (1, 3)
- ○ 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 = (3, 3)

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

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

Correct

7. Recall that $np.dot(a, b)$ performs a matrix multiplication on a and b, whereas $a * b$ performs an element-wise multiplication.

1 / 1 point

Consider the two following random arrays a and b:

$$a = np.random.randn(12288, 150) \# a.shape = (12288, 150)$$

$$b = np.random.randn(150, 45) \# b.shape = (150, 45)$$

$$c = np.dot(a, b)$$

What is the shape of c?

- ○ The computation cannot happen because the sizes don't match. It's going to be "Error"!
- ○ c.shape = (150,150)
- ● c.shape = (12288, 45)
- ○ c.shape = (12288, 150)

Correct

Correct, remember that a np.dot(a, b) has shape (number of rows of a, number of columns of b). The sizes match because :

"number of columns of a = 150 = number of rows of b"

8. Consider the following code snippet:

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]

How do you vectorize this?

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

Incorrect
No. Notice that b is a column vector; but we are using it to fill the row i of c.

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

$a.shape = (3, 3)$

$b.shape = (3, 3)$

$c = a * *2 + b.T * *2$

Which of the following gives an equivalent output for c?

- The computation cannot happen because the sizes don't match. It's going to be an "Error"!
-

  for i in range(3):

  c[i] = a[i]**2 + b[i]**2

- [x]

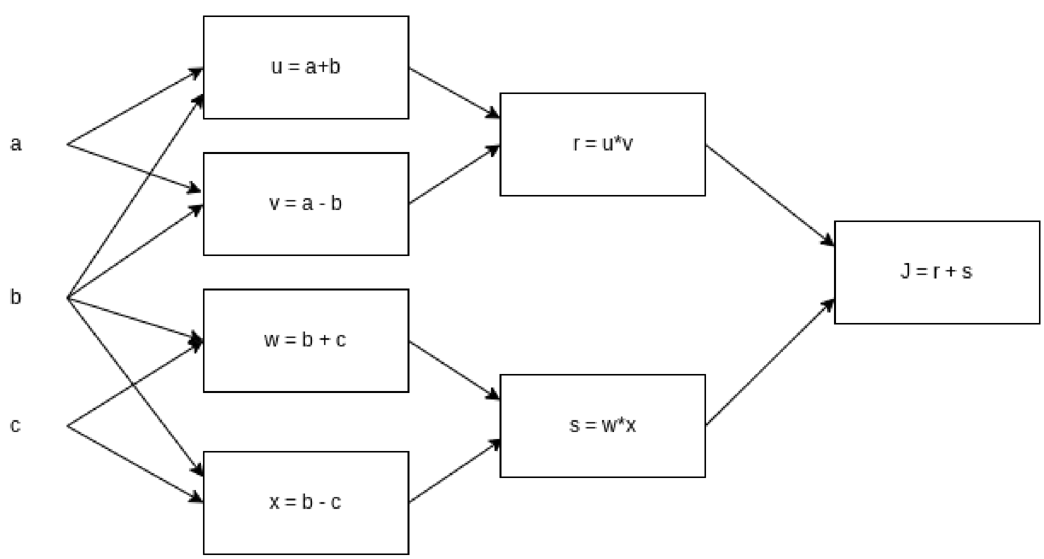
```
for i in range(3):
 for j in range(3):
 c[i][j] = a[i][j]**2 + b[j][i]**2
```

- [ ]

```
for i in range(3):
 for j in range(3):
 c[i][j] = a[i][j]**2 + b[i][j]**2
```

Correct
Yes. This code squares each entry of a and adds it to the transpose of b square.

10. Consider the following computational graph.



What is the output of J?

- [ ] $a^2 + b^2 - c^2$

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

Correct

Yes.

$J = r + s = u * v + w * x = (a+b)*(a-b) + (b+c)*(b-c) = a^2 - b^2 + b^2 - c^2 = a^2 - c^2$