Your grade: 80%

Your latest: **80%** • Your highest: **80%** • To pass you need at least 80%. We keep your highest score.

Next item

1. What does a neuron compute? **1 / 1 point**

- A neuron computes the mean of all features before applying the output to an activation function
- A neuron computes an activation function followed by a linear function $z = Wx + b$
- A neuron computes a function g that scales the input x linearly (Wx + b)
- (selected) A neuron computes a linear function $z = Wx + b$ followed by an activation function

Correct
Correct, we generally say that the output of a neuron is a = g(Wx + b) where g is the activation function (sigmoid, tanh, ReLU, ...).

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

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

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

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

- x.reshape(-1, 3)
- (selected) x.reshape(2, 2, 2)
- x.reshape(1, 4, 3)
- x.reshape(2, 4, 4)

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

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

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: **0 / 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?

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

Incorrect
No. Broadcasting is invoked, multiplying each row of b, element-wise with a.

6. 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? 1 / 1 point

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

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

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?

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

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: 0 / 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?

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

Incorrect
No. The a[j][i] being assigned to a[i][j] indicates that we are using a.T.

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"!

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

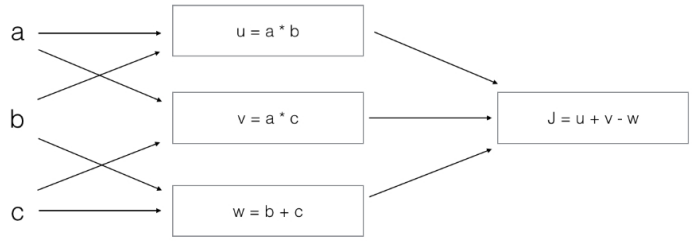
```
for i in range(3):
    c[i] = a[i]**2 + b[i]**2
```

```
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 computation graph. **1 / 1 point**



What is the output J?

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

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