Congratulations! You passed!

Grade received 100% **Latest Submission Grade** 100% **To pass** 80% or higher

Go to 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 linear function $z = Wx + b$ followed by an activation function
- A neuron computes a function g that scales the input x linearly (Wx + b)

Expand

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

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

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?

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

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

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

$c = a + b$

What will be the shape of c?

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

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**

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

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

$c = a * b$

What will be the shape of c?

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

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?

- (m, n_x)
- $(1, n_x)$
- (n_x, m)
- (n_x, 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. 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 = (150,150)
- c.shape = (12288, 45)
- c.shape = (12288, 150)
- The computation cannot happen because the sizes don't match. It's going to be "Error"!

Expand

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

Expand

Correct

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

$a.shape = (3, 3)$

$b.shape = (3, 3)$

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

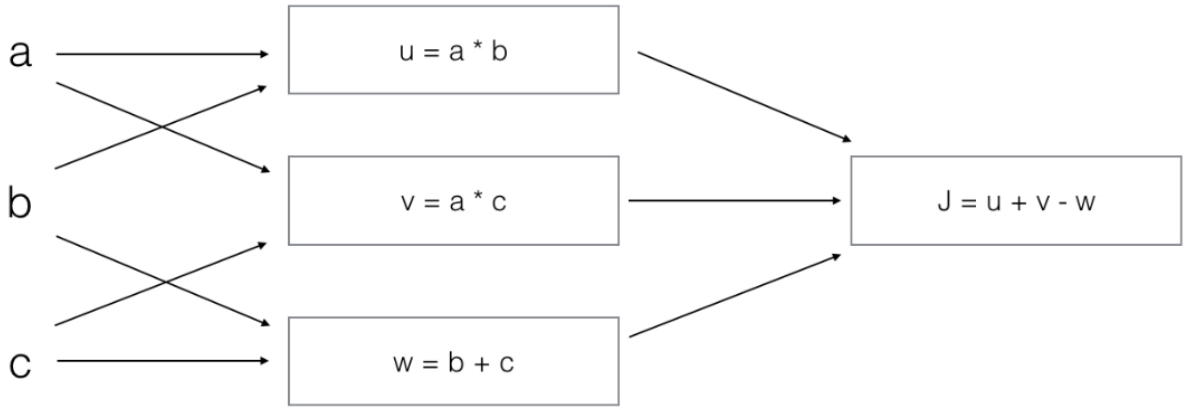
Which of the following gives an equivalent output for c?

- 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[i][j]**2
- for i in range(3):
  for j in range(3):
  c[i][j] = a[i][j]**2 + b[j][i]**2
- The computation cannot happen because the sizes don't match. It's going to be an "Error"!

Expand

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?

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

Expand

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