Question

What are the parameters of logistic regression?

- ( ) W and b, both n_x dimensional vectors.
- ( ) W and b, both real numbers.
- (•) W, an n_x dimensional vector, and b, a real number.
- ( ) W, an identity vector, and b, a real number.

✓ **Correct**

What is the difference between the cost function and the loss function for logistic regression?

- ○ The cost function computes the error for a single training example; the loss function is the average of the cost functions of the entire training set.
- ◉ The loss function computes the error for a single training example; the cost function is the average of the loss functions of the entire training set.
- ○ They are different names for the same function.

✓ **Correct**

True or false. A convex function always has multiple local optima.

- [ ] True
- [x] False

✓ **Correct**

One step of _______ propagation on a computation graph yields derivative of final output variable.

- ○ Forward
- ◉ Backward

In this class, what does the coding convention dvar represent?

- [ ] The derivative of any variable used in the code.
- [ ] The derivative of input variables with respect to various intermediate quantities.
- [x] The derivative of a final output variable with respect to various intermediate quantities.

Correct

In this video, what is the simplified formula for the derivative of the losswith respect to z?

- a / (1-a)
- a - y
- a (1 - y)

✓ **Correct**

1. What does a neuron compute?

1 / 1 point

- ○ 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)
- ◉ A neuron computes a linear function (z = Wx + b) followed by an activation function
- ○ A neuron computes the mean of all features before applying the output to 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

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

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?

1 / 1 point

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

Correct

4. Consider the two following random arrays a and b:

1 / 1 point

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

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

$c = a + b$

What will be the shape of c?

- ○ c.shape = (3, 2)
- ○ c.shape = (2, 1)
- ○ The computation cannot happen because the sizes don't match. It's going to be "Error"!
- ◉ c.shape = (2, 3)

5. Consider the two following random arrays a and b:

1 / 1 point

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

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

$c = a * b$

What will be the shape of c?

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

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).

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

1 / 1 point

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

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 = (150,150)
- ◉ c.shape = (12288, 45)

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:

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

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

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

Correct

9. Consider the following code:

1 / 1 point

$a = np.random.randn(3, 3)$

$b = np.random.randn(3, 1)$

$c = a * b$

What will be c? (If you're not sure, feel free to run this in python to find out).

- It will lead to an error since you cannot use "*" to operate on these two matrices. You need to instead use np.dot(a,b)

- This will invoke broadcasting, so b is copied three times to become (3, 3), and $*$ invokes a matrix multiplication operation of two 3x3 matrices so c.shape will be (3, 3)

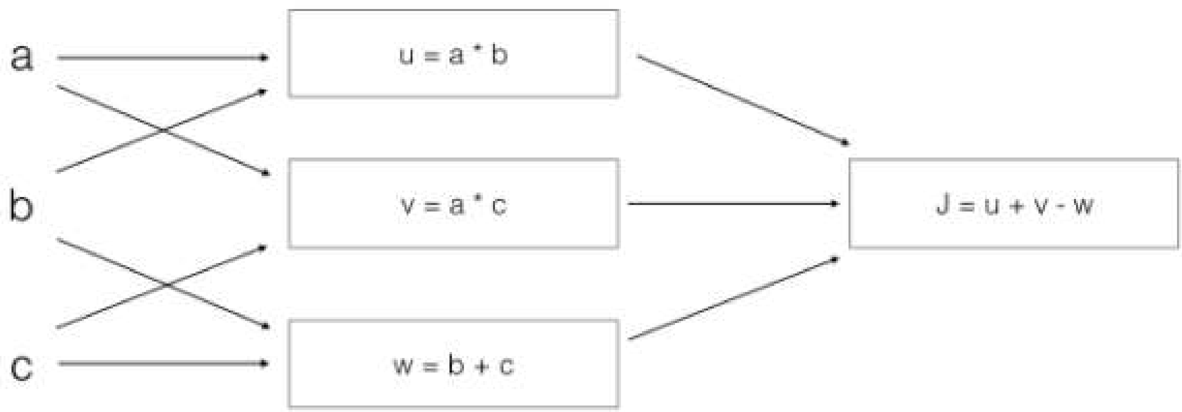
- This will invoke broadcasting, so b is copied three times to become (3,3), and $*$ is an element-wise product so c.shape will be (3, 3)

- This will multiply a 3x3 matrix a with a 3x1 vector, thus resulting in a 3x1 vector. That is, c.shape = (3,1).

Correct

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).