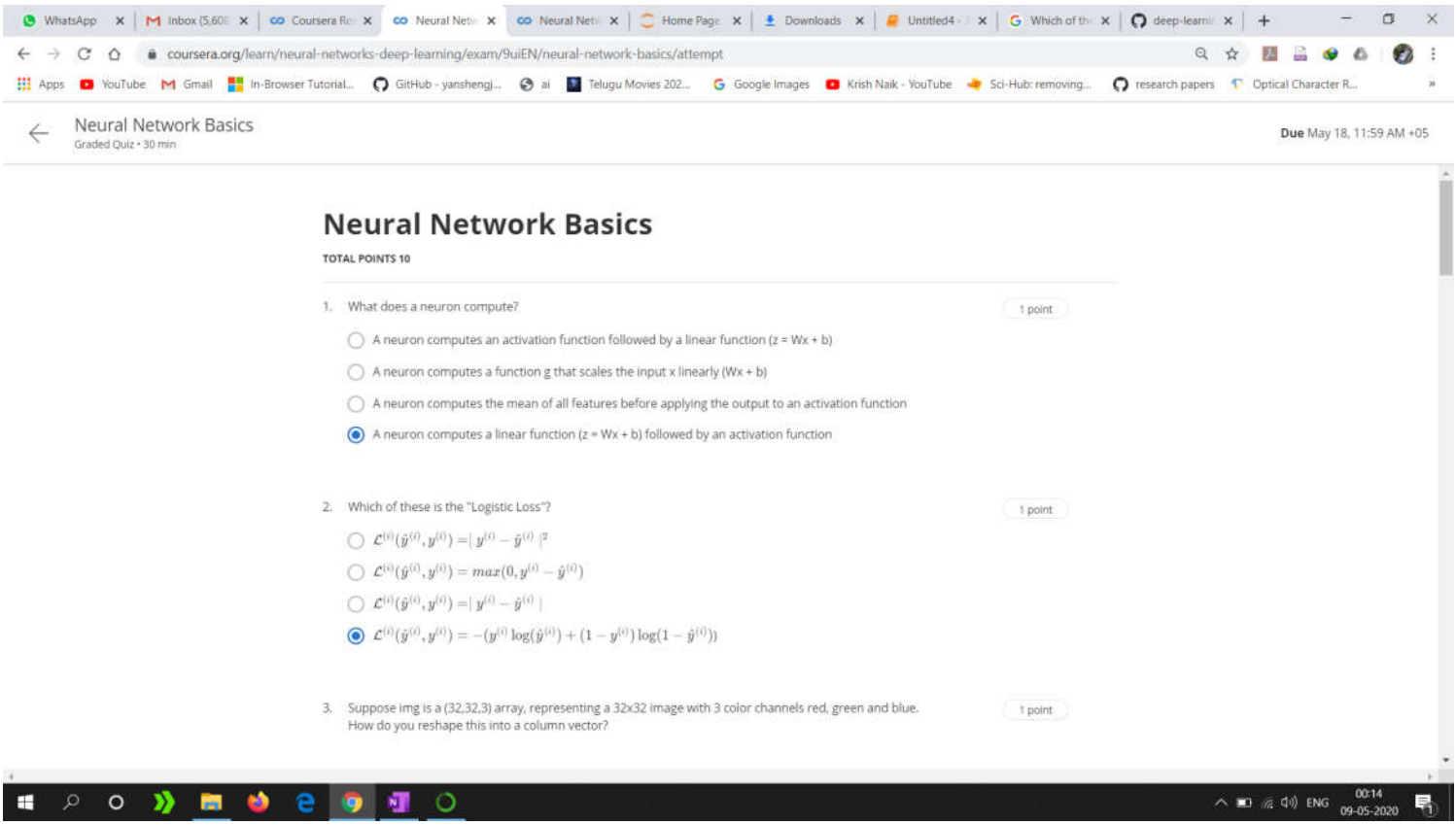
Week2-Quiz-2

08 May 2020 00:38



coursera.org/learn/neural-networks-deep-learning/exam/9uiEN/neural-network-basics/attempt

Neural Network Basics
Graded Quiz • 30 min

Due May 18, 11:59 AM +05

Neural Network Basics

TOTAL POINTS 10

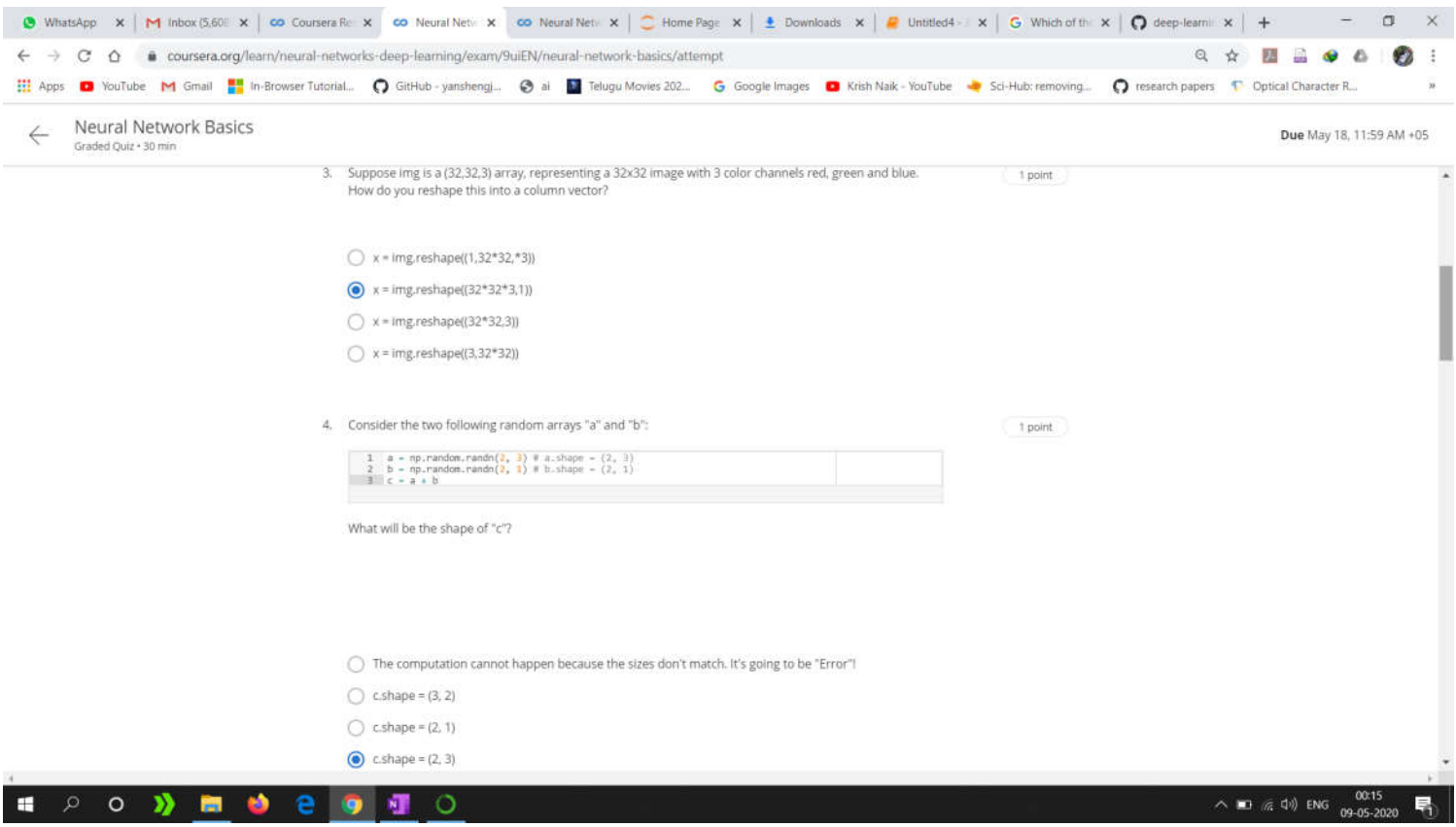
1. What does a neuron compute? 1 point

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

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

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

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 point



coursera.org/learn/neural-networks-deep-learning/exam/9uiEN/neural-network-basics/attempt

Neural Network Basics
Graded Quiz • 30 min

Due May 18, 11:59 AM +05

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 point

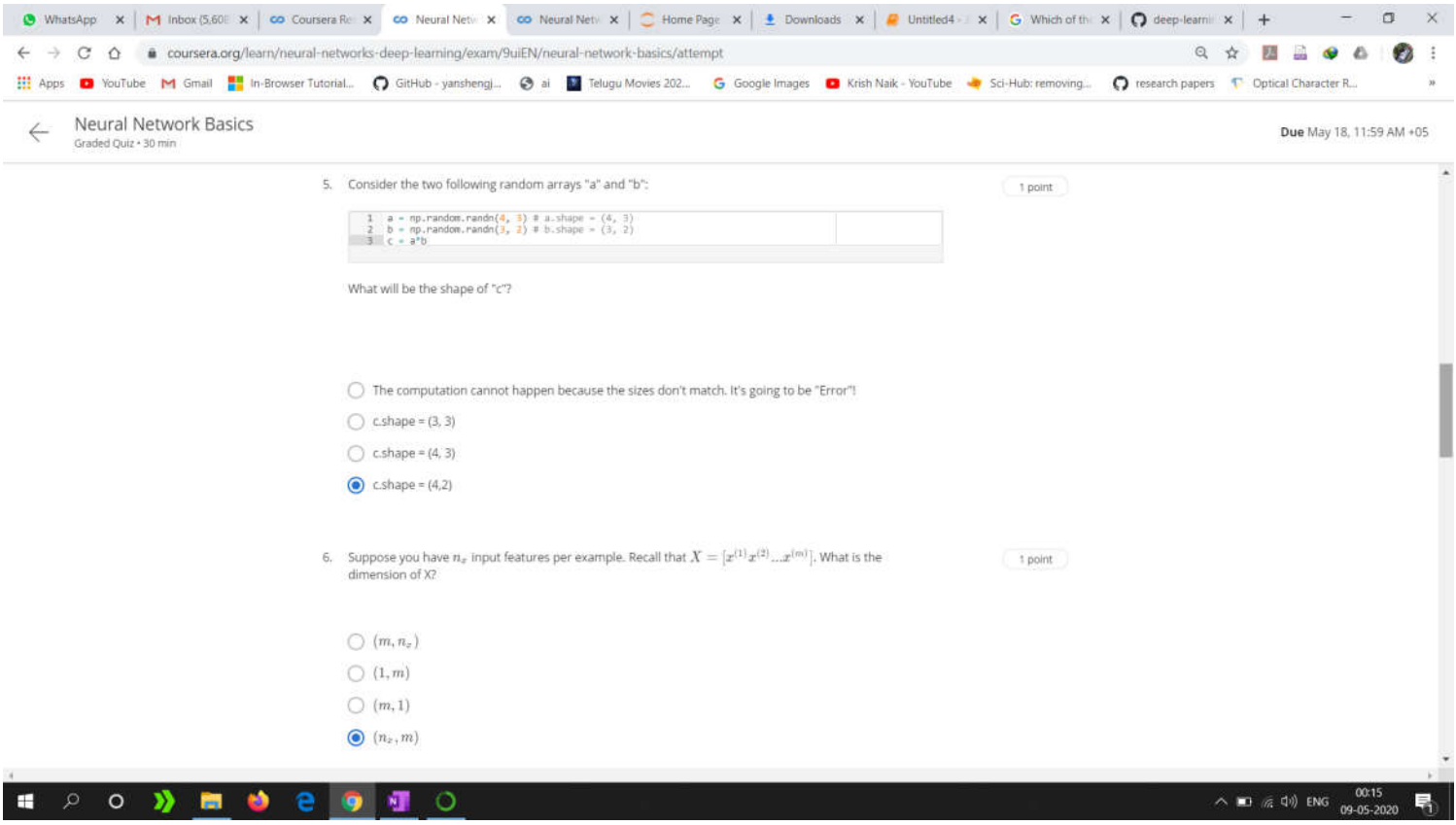
- x = img.reshape((1,32*32,*3))
- (selected) x = img.reshape((32*32*3,1))
- x = img.reshape((32*32,3))
- x = img.reshape((3,32*32))

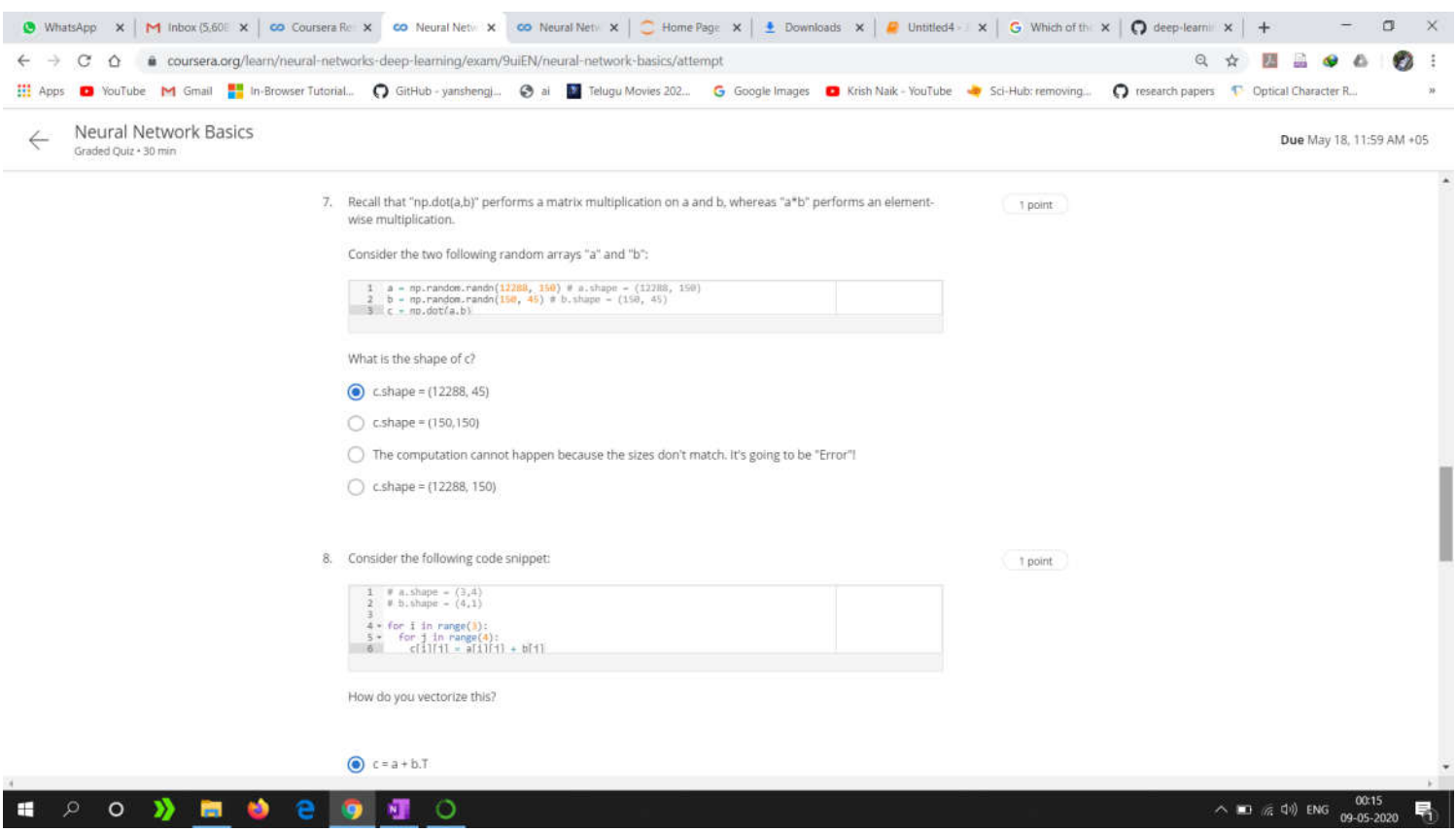
4. Consider the two following random arrays "a" and "b": 1 point

```
a = np.random.randn(2, 3) # a.shape = (2, 3)
b = np.random.randn(3, 1) # b.shape = (3, 1)
c = a + b
```

What will be the shape of "c"?

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


coursera.org/learn/neural-networks-deep-learning/exam/9uiEN/neural-network-basics/attempt
Neural Network Basics
Graded Quiz • 30 min
Due May 18, 11:59 AM +05
5. Consider the two following random arrays "a" and "b":
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"?
The computation cannot happen because the sizes don't match. It's going to be "Error"!
c.shape = (3, 3)
c.shape = (4, 3)
c.shape = (4,2)
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 point
(m, n_x)
$(1, m)$
$(m, 1)$
(n_x, m)


coursera.org/learn/neural-networks-deep-learning/exam/9uiEN/neural-network-basics/attempt
Neural Network Basics
Graded Quiz • 30 min
Due May 18, 11:59 AM +05
7. Recall that "np.dot(a,b)" performs a matrix multiplication on a and b, whereas "a*b" performs an element-wise multiplication.
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, 45)
c.shape = (150,150)
The computation cannot happen because the sizes don't match. It's going to be "Error"!
c.shape = (12288, 150)
8. Consider the following code snippet:
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

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

9. Consider the following code: 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).

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

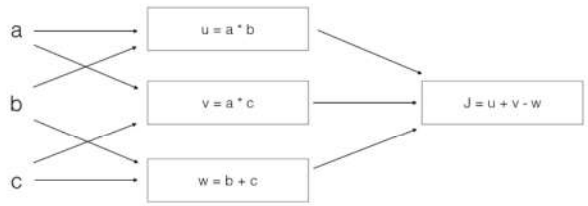
coursera.org/learn/neural-networks-deep-learning/exam/9uiEN/neural-network-basics/attempt

Neural Network Basics
Graded Quiz • 30 min

Due May 18, 11:59 AM +05

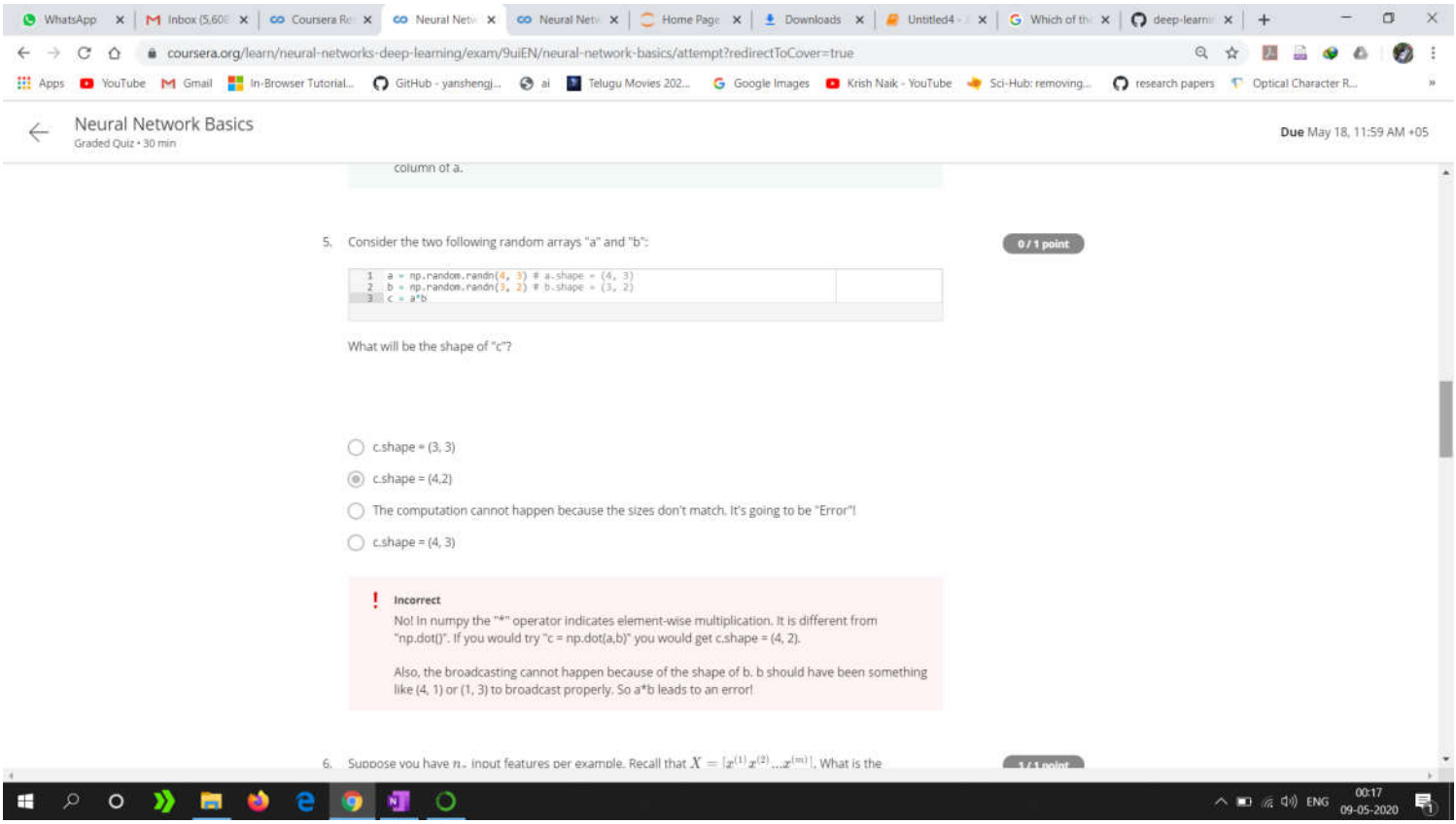
use np.dot(a,b)

10. Consider the following computation graph. 1 point



What is the output J?

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


coursera.org/learn/neural-networks-deep-learning/exam/9uiEN/neural-network-basics/attempt?redirectToCover=true
Neural Network Basics
Graded Quiz • 30 min
Due May 18, 11:59 AM +05
column of a.
5. Consider the two following random arrays "a" and "b":
0 / 1 point
1 a = np.random.randn(4, 3) # a.shape = (4, 3)
2 b = np.random.randn(3, 2) # b.shape = (3, 2)
3 c = a*b
What will be the shape of "c"?
c.shape = (3, 3)
c.shape = (4,2)
The computation cannot happen because the sizes don't match. It's going to be "Error"!
c.shape = (4, 3)
Incorrect
No! 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).
Also, the broadcasting cannot happen because of the shape of b. b should have been something like (4, 1) or (1, 3) to broadcast properly. So a*b leads to an error!
6. Suppose you have n_x input features per example. Recall that $X = [x^{(1)} x^{(2)} ... x^{(m)}]$. What is the
1 / 1 point
ENG
00:17
09-05-2020