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. Suppose that $\hat{y} = 0.5$ and $y = 0$. What is the value of the "Logistic Loss"? Choose the best option. **1 / 1 point**

- 0.5
- 0.693
- $+\infty$
- $$ \mathcal{L}(\hat{y}, y) = - \left(y \log \hat{y} + (1-y) \log (1 - \hat{y}) \right) $$

Loading [MathJax]/jax/output/CommonHTML/jax.js

Expand

Correct
Yes. Given the values of $\hat{y}$ and y we get $\mathcal{L}(0.5, 0) = -(0 \log 0.5 + 1 \log(0.5)) \approx 0.693$.

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**

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

Loading [MathJax]/jax/output/CommonHTML/jax.js

Expand

Correct

4. Consider the following random arrays a and b, and c: **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 = (2, 3)
- The computation cannot happen because the sizes don't match. It's going to be "Error"!

- c.shape = (2, 1)
- c.shape = (3, 2)

Expand

Correct
Yes! This is broadcasting. b (column vector) is copied 3 times so that it can be summed to each column of a.

5. Consider the two following random arrays a and b: **1 / 1 point**

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

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

$c = a * b$

What will be the shape of c?

- The computation cannot happen because it is not possible to broadcast more than one dimension.
- The computation cannot happen because the sizes don't match.
- c.shape = (4, 3)
- c.shape = (1, 3)

Expand

Correct
Yes. Broadcasting is invoked, so row b is multiplied element-wise with each row of a to create c.

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**

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

Expand

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. Consider the following array: **1 / 1 point**

$a = np.array([[2, 1], [1, 3]])$

What is the result of $np.dot(a, a)$?

- $\begin{pmatrix} 5 & 5 \\ 5 & 10 \end{pmatrix}$
- The computation cannot happen because the sizes don't match. It's going to be an "Error"!
- $$\,$$ $$\begin{pmatrix} 4 & 1 \\ 1 & 9 \end{pmatrix}$$

Expand

Correct
Yes, recall that * indicates the element wise multiplication and that np.dot() is the matrix multiplication.
Thus $\begin{pmatrix} (2)(2)+(1)(1) & (2)(1)+(1)(3) \\ \end{pmatrix}$.

$\backslash(1)(2) + (3)(1) \quad (1)(1) + (3)(3)\,/$

8. Consider the following code snippet: **1 / 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.T + b
- c = a.T + b.T
- c = a + b.T
- c = a + b

Expand

Correct
Yes. a[j][i] being used for a[i][j] indicates we are using a.T, and the element in the row j is used in the column j thus we are using b.T.

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

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

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

$c = a * b$

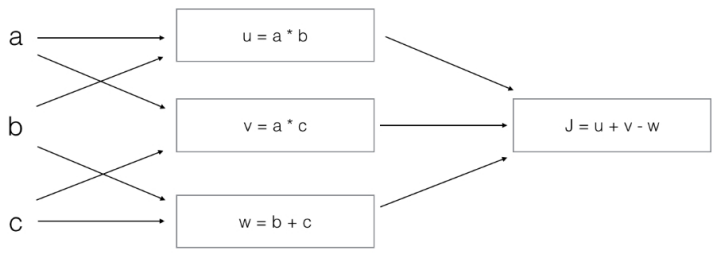
What will be c? (If you're not sure, feel free to run this in python to find out).

- 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).
- 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)

Expand

Correct

10. Consider the following computation graph. **1 / 1 point**



What is the output J?

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

Expand

Correct

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