Congratulations! You passed!

Grade received 90% To pass 80% or higher

Go to next item

Neural Network Basics

Latest Submission Grade 90%

1. In logistic regression given $\mathbf{x} \in \mathbb{R}^{n_x}$, $b \in \mathbb{R}$. Which of the following best expresses what we want $\hat{y}$ to tell us? **1 / 1 point**

- $\sigma(W\,\mathbf{x} + b)$
- $P(y = 1|\mathbf{x})$
- $\sigma(W\,\mathbf{x})$

Correct
Yes. We want the output $\hat{y}$ to tell us the probability that $y = 1$ given $\mathbf{x}$.

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

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

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?

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

Yes. This array has two rows and in each row it has 2 arrays of 1s.

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

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.
- [x] c.shape = (4, 3)
- [ ] c.shape = (1, 3)
- [ ] The computation cannot happen because the sizes don't match.

6. **1 / 1 point**

Suppose we use as input grayscale images of 8x8. We reshape these images into feature column vectors $\mathbf{x}^{(i)}$. Remember that $X = [\mathbf{x}^{(1)}\mathbf{x}^{(2)} \cdots \mathbf{x}^{(m)}]$. What is the dimension of X?

- (512, 1)
- (8, 8, 8)
- (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. Consider the following array: **1 / 1 point**

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

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

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

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

8. **0 / 1 point**

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

Incorrect
No. Notice that b is a column vector; but we are using it to fill the row i of c.

9. Consider the following arrays:

&&a = np.array([[1, 1], [1, -1]])$$

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

$c = a + b$

Which of the following arrays is stored in c?

- The computation cannot happen because the sizes don't match. It's going to be an "Error"!
- $\begin{matrix} 3 & 3 \\ 4 & 2 \end{matrix}$
- $\begin{pmatrix} 3 & 3 \\ 3 & 1 \\ 4 & 4 \\ 5 & 2 \end{pmatrix}$
- $\begin{matrix} 3 & 4 \end{matrix}$

Correct
Yes. The array b is a column vector. This is copied two times and added to the array a to construct the array c.

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

What is the output of J?

- $(a - 1)\,(b + c)$
- $(a + c)\,(b - 1)$
- $ab + bc + ac$
- $(c - 1)\,(a + c)$

Correct
Yes. $J = u - v + w = ab - (a + c) + bc = ab - a + bc - c = a\,(b - 1) + c\,(b - 1) = (a + c)\,(b - 1)$