Neural Network Basics

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

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

Expand

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

1. In logistic regression given the input $\mathbf{x}$, and parameters $w \in \mathbb{R}^{n_x}, b \in \mathbb{R}$, how do we generate the output $\hat{y}$? 1 / 1 point

- $\sigma(W\mathbf{x})$
- $\tanh(W\mathbf{x} + b)$
- $W\mathbf{x} + b$
- $\sigma(W\mathbf{x} + b)$.

Expand

Correct
Right, in logistic regression we use a linear function $W\mathbf{x} + b$ followed by the sigmoid function σ, to get an output y, referred to as ŷ, such that $0 < \hat{y} < 1$.

2. Suppose that $\hat{y} = 0.9$ and $y = 1$. What is the value of the "Logistic Loss"? Choose the best option. 1 / 1 point

- $+\infty$
- $\mathcal{L}(\hat{y}, y) = -(\hat{y} \log y + (1 - \hat{y}) \log(1 - y))$
- 0.105
- 0.005

Expand

Correct
Yes. Since $\mathcal{L}(\hat{y}, y) = -(y \log \hat{y} + (1 - y) \log(1 - \hat{y}))$, for the given values we get $\mathcal{L}(\hat{y}, y) = -(1 \log 0.9 + 0 \log 0.1)$

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

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

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

Expand

Correct

3. Consider the Numpy array x: 1 / 1 point

$x = np.array([[[1], [2]], [[3], [4]]])$

What is the shape of x?

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

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

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

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

$c = a + b$

What will be the shape of c?

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

Expand

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

4. Consider the following random arrays a and b, and c: 1 / 1 point

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

$b = np.random.randn(1,4)$ # $b.shape = (1,4)$

$c = a + b$

What will be the shape of c?

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

Expand

Correct
Yes. Broadcasting is used, so row b is copied 3 times so it can be summed to each row of a.

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?

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

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

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

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

$c = a * b$

What will be the shape of c?

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

Expand

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

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

Expand

Correct
Yes. Broadcasting allows row a to be multiplied element-wise with each row of b to from c.

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

Expand

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).

1 / 1 point

6. 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.

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 / 1 point

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

Expand

Correct

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?

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

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"

7. Consider the following array: 1 / 1 point

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

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

- $\begin{pmatrix} 4 & 1 \\ 1 & 9 \end{pmatrix}$
- $\begin{pmatrix} 4 & 2 \\ 2 & 6 \end{pmatrix}$
- $\begin{pmatrix} 5 & 5 \\ 5 & 10 \end{pmatrix}$
- The computation cannot happen because the sizes don't match. It's going to be

Expand

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

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 = np.dot(a,b)
- c = a*b
- c = a.T*b

Expand

Correct
Yes. b.T gives a column vector with shape (1, 4). The result of c is equivalent to broadcasting a*b.T.

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

Expand

Correct

9. Consider the following arrays: 1 / 1 point

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

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

$c = a + b$

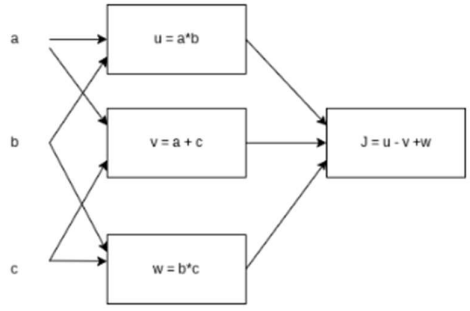
Which of the following arrays is stored in c?

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

Expand

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 computational graph. 1 / 1 point



What is the output of J?

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

Expand

Correct

Yes.

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