검색

- 설정
- 업데이트
- 성취도
- 도움말 센터
- 로그아웃
- Youngeun In

- as a Neural Network
- ation
- nal)
- **Quiz**
  - ✓ [테스트: Neural Network Basics
    10개의 질문]
- **Programming Assignments**
- **Heroes of Deep Learning (Optional)**

테스트테스트 • 20 min20 minutes

Neural Network Basics

✓
과제 제출
기한년 8월 23일 오후 3:59 KST년 8월 23일 오후 3:59 KST

다시 시도해주십시오

✓
성적 받기
통과 점수:80% 이상
성적
100%

피드백 보기

최고 점수가 유지됩니다.

탐색 확인

이 페이지에서 나가시겠습니까?

이 페이지에 머물기 이 페이지에서 나가기

←

Neural Network Basics
성적 평가 퀴즈 • 20 min

만료 년 8월 23일 오후 3:59 KST

✓

축하합니다! 통과하셨습니다!
통과 점수: 80% 이상

학습 계속하기

성적
100%

Neural Network Basics

최신 제출물 성적
100%
1.
질문 1

What does a neuron compute?

1 / 1점

● ○

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)

○ ○

A neuron computes an activation function followed by a linear function (z = Wx + b)

○ ○

A neuron computes the mean of all features before applying the output to an activation function

✓

맞습니다

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.
질문 2

Which of these is the "Logistic Loss"?

1 / 1점

○ ○

$L^{(i)}(\hat{y}^{(i)}, y^{(i)}) = \mid y^{(i)} - \hat{y}^{(i)} \mid$ L(i)(y^(i), y(i)) =∣ y(i) − y^(i) ∣

○ ○

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

● ○

$L^{(i)}(\hat{y}^{(i)}, y^{(i)}) = -(y^{(i)}\log(\hat{y}^{(i)}) + (1 - y^{(i)})\log(1 - \hat{y}^{(i)})$L(i)(y^(i), y(i)) = −(y(i) log(y^(i)) + (1 − y(i)) log(1 − y^(i)))

○ ○

$L^{(i)}(\hat{y}^{(i)}, y^{(i)}) = \mid y^{(i)} - \hat{y}^{(i)} \mid^2$L(i)(y^(i), y(i)) =∣ y(i) − y^(i) ∣2

✓

맞습니다

Correct, this is the logistic loss you've seen in lecture!

3.
질문 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 / 1점

○○

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

○○

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

●○

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

○○

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

✓
맞습니다
4.
질문 4

Consider the two following random arrays aa and bb:

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

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

$c = a + b$c = a + b

What will be the shape of cc?

1 / 1점

○○

The computation cannot happen because the sizes don't match. It's going to be "Error"!

●○

c.shape = (2, 3)

○○

c.shape = (2, 1)

○○

c.shape = (3, 2)

✓
맞습니다

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

5.
질문 5

Consider the two following random arrays aa and bb:

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

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

$c = a * b$c = a * b

What will be the shape of cc?

1 / 1점

○ ○

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)

✓

맞습니다

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.

질문 6

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

1 / 1점

○ ○

$(m, 1)$(m, 1)

○ ○

$(1, m)$(1, m)

● ○

(n_x, m)(nx, m)

○ ○

(m, n_x)(m, nx)

✓

맞습니다

7.

질문 7

Recall that $np.dot(a,b)$np.dot(a, b) performs a matrix multiplication on aa and bb, whereas $a*b$a ∗ b performs an element-wise multiplication.

Consider the two following random arrays aa and bb:

$a = np.random.randn(12288, 150)$a = np.random.randn(12288, 150) # $a.shape = (12288, 150)$a.shape = (12288, 150)

$b = np.random.randn(150, 45)$b = np.random.randn(150, 45) # b.shape = (150, 45)$$

$c = np.dot(a,b)$c = np.dot(a, b)

What is the shape of cc?

1 / 1점

○ ○

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)

✓
맞습니다

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.
질문 8

Consider the following code snippet:

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

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

```
for i in range(3):
    for j in range(4):
        $c[i][j] = a[i][j] + b[j]$c[i][j] = a[i][j] + b[j]
```

How do you vectorize this?

1 / 1점
○○

c = a.T + b.T

○○

c = a.T + b

○○

c = a + b

◉○

c = a + b.T

✓
맞습니다
9.
질문 9

Consider the following code:

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

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

$c = a*b$c = a ∗ b

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

1 / 1점
◉○

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)

○○

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

○ ○

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)

✓
맞습니다
10.
질문 10

Consider the following computation graph.

What is the output J?

1 / 1점
○ ○

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

● ○

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

○ ○

J = a*b + b*c + a*c

○ ○

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

✓
맞습니다

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