

# [AI project 중간시험] 2021.10.26(화) 16:30

학번 ( ) 이름 ( )

## (시험 요령)

1. 답안은 파일이름이 자기이름.ipynb 인 IPython 파일로 제출합니다
2. 안전을 위하여 모든 답안지 내용을 자기이름.pdf 파일로 인쇄하여 추가로 제출합니다
3. 첫번째 cell에 자기이름을 적고 cell type을 markdown으로 설정합니다
4. 답안에는 코드, 실행결과, 서술문, 그림을 모두 포함합니다
5. 그림은 markdown cell안에 복사붙여넣기 하고 실행시켜서 작성합니다

※ 문제 풀이 코드 맨 앞에 넣어주세요

```
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"
```

I. 이러닝의 [Quiz2]에서 문장이 맞으면 True 틀리면 False로 답하세요. [각1점]

본 T/F 문제는 이러닝 학습관리시스템 INU LMS(learning management system) 문제에 직접 답을 선택합니다.

II. 다음 함수에 대하여 변수  $x, y, z$ 의 값이 모두 2 일 때 미분한 값을 구하려고 합니다 [각10점]

1.  $x$  대한 미분 값을 구하는  $\partial f / \partial x$  식을 구해보세요.
2.  $x, y, z$ 의 값이 모두 2일 때의  $\partial f / \partial x$  미분 값을 파이썬 코드로 계산해보세요.

$$f = 2xy + 3x^2z + 4z$$

(개선) 변수  $x, y, z$ 의 값이 모두 2 일 때 다음 함수의  $x$ 에 대하여 미분한 값  $\partial f / \partial x$ 를 파이썬 코드로 구해보세요. [각10점]

$$f = 2xy + 3x^2z + 4z$$

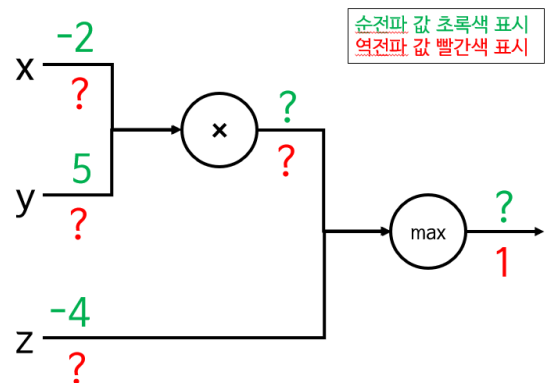
1.  $x, y, z$ 가 서로 독립적인 경우
2.  $x$ 와  $y$ 가 아래의 관계가 있는 경우

$$y = x^3 + x$$

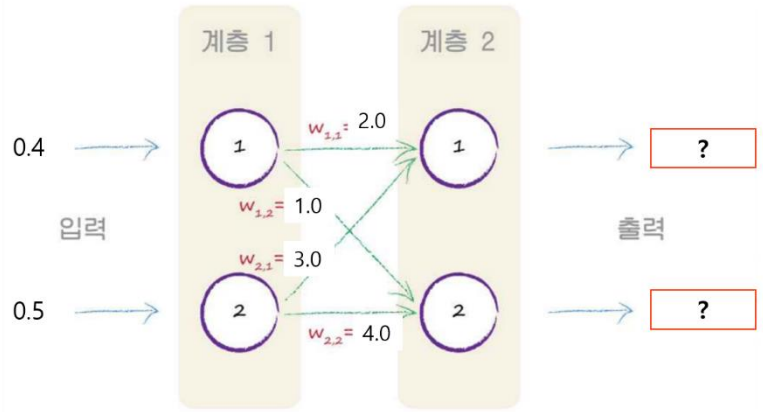
III. 아래 그림은 간단한 역전파의 예입니다. 변수들 사이의 관계는  $f(x, y, z) = \max(x \times y) + z$  입니다. 순방향 패스는 입력에서 출력으로 값을 계산합니다(초록색 표시), 역방향 패스는 역전파를 수행하여 출력에서 입력에 이르기까지 그래디언트(빨간색 표시)를 계산합니다. 전파해야 할 오차 값은 1이라고 가정하고, 역전파에 반영하는 가중치 비율은 정규화(normalizing) 하지 않고 적용합니다. [각10점]

(Hint) <https://cs231n.github.io/optimization-2/> 참고하세요.

1. 물음표의 값을 계산해서 채우세요.
2. 이러한 회로를 구현하는 코드를 작성하고 실행한 결과가 1번의 답과 같음을 보이세요.



IV. 아래 그림과 같은 2계층, 계층별 2개 뉴런을 가진 단순한 신경망의 입력 노드의 값은 각각 0.4와 0.5이고, 가중치가  $w_{1,1} = 2.0$ ,  $w_{1,2} = 1.0$ ,  $w_{2,1} = 3.0$ ,  $w_{2,2} = 4.0$ 로 초기화 되었다고 가정하고 각 물음에 답하세요. 모든 결과 값은 반올림하지 않고 출력합니다. [각10점]



1. 활성화 함수로 수식  $y = \frac{1}{1+e^{-x}}$  로 표현되는 sigmoid 함수를 사용할 때 각 출력 노드의 출력 값은 각각 얼마인지 계산하는 코드를 작성하고 그 실행결과를 보이세요.

(hint)

```
def sigmoid(x):
    return 1 / (1 + numpy.exp(-x))

input
[[0.4]
 [0.5]]
W_input_output
[[2. 3.]
 [1. 4.]]
```

$$y = \frac{1}{1 + e^{-x}}$$

2. 위 예제에서 목표값(target)이 각각 target1 = 1.70887704와 target2 = 1.4168273 이었을 때, 입력 계층으로 역전파되는 오차 값은 각각 얼마인지 계산하는 코드를 작성하고 그 실행결과를 보이세요. 오차(error)는 L1(절대값) 손실(loss), 즉  $\text{abs}(\text{target} - \text{output})$ 을 사용하고, 역전파에 반영하는 가중치 비율은 정규화(normalizing) 하여 적용합니다.

```
target
[[1.70887704]
 [1.4168273 ]]
```

3. 학습률이 0.1일 때, 위 예제의 1번과 2번 문제처럼 1회 순전파 및 1회 역전파를 수행한 후 업데이트된 가중치  $W_{\text{input\_hidden}}$  값들은 각각 얼마로 업데이트 되는지 계산하는 코드를 작성하고 그 실행결과를 보이세요.

V. 아래는 신경망 학습에서 가중치를 업데이트하는 핵심 알고리즘들을 표현합니다. 이들 알고리즘을 코딩한 Python 코드와 알고리즘을 구성하는 각 변수들을 해당하는 것끼리 그림으로 또는  $W_{jk} = \text{Who}$  형식으로 서로 연결하여 표시해 보세요. 단, j는 은닉계층을, k는 출력계층을 표현하며, who는 은닉계층과 출력계층 사이의 가중치 값을 표현합니다. [10점]

$$\text{new } w_{jk} = \text{old } w_{jk} - \alpha \cdot \frac{\partial E}{\partial w_{jk}}$$

$$\Delta W_{jk} = \alpha * E_k * \text{sigmoid}(O_k) * (1 - \text{sigmoid}(O_k)) \cdot O_j^T$$

```
#은닉 계층과 출력 계층 간의 가중치 업데이트
self.who = self.lr * numpy.dot((output_errors * final_outputs * (1.0 - final_outputs)), numpy.transpose(hidden_outputs))
```

< 수고 많으셨습니다! >