



# Day58; 20221128

날짜	@2022년 11월 28일
유형	@2022년 11월 28일
태그	

**GitHub - u8yes/AI**

You can't perform that action at this time. You signed in with another tab or window. You signed out in another tab or window. Reload to refresh your session. Reload to refresh your session.

<https://github.com/u8yes/AI>

**u8yes/AI**

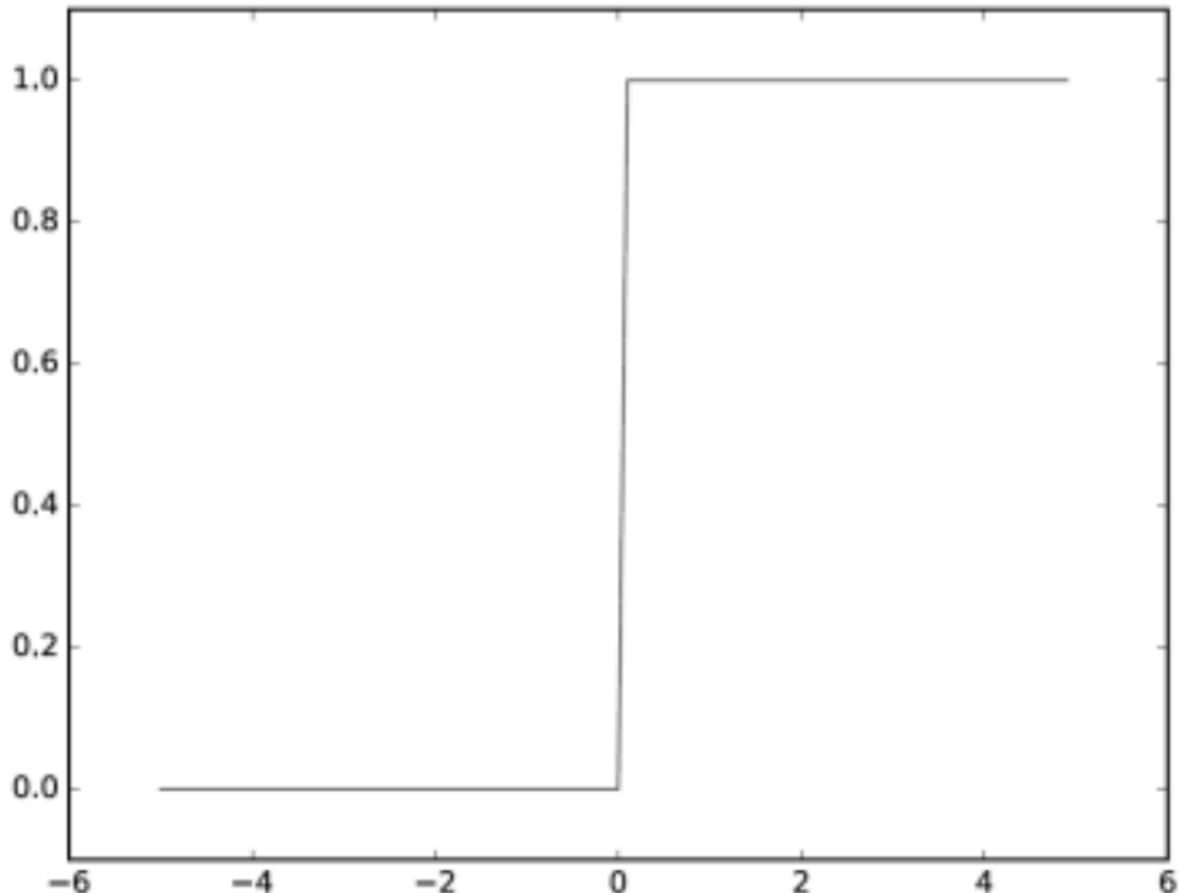


Ax 1 Contributor    0 Issues    0 Stars    0 Forks

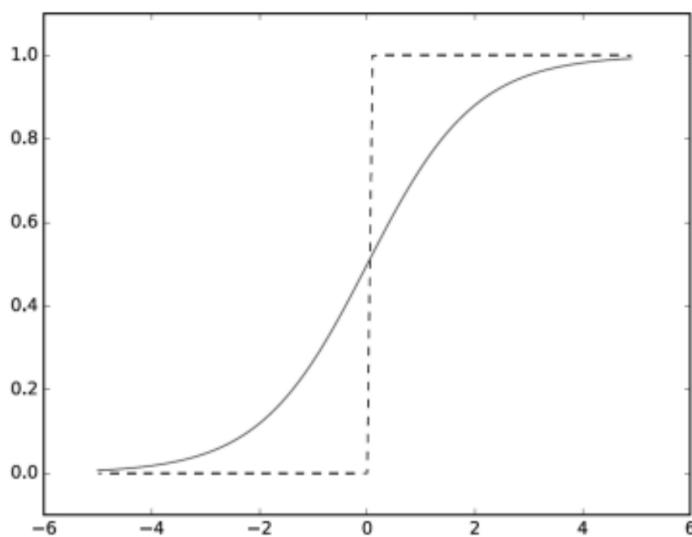
[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/d7733398-72c6-4e87-8eb9-9e00c5bda688/06\\_%EC%8B%A0%EA%B2%BD%EB%A7%9D\(%EB%94%A5%EB%9F%AC%EB%8B%9D\).pdf](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/d7733398-72c6-4e87-8eb9-9e00c5bda688/06_%EC%8B%A0%EA%B2%BD%EB%A7%9D(%EB%94%A5%EB%9F%AC%EB%8B%9D).pdf)

## Step function

# 계단 함수(step function)



## 시그모이드 함수와 계단 함수 비교



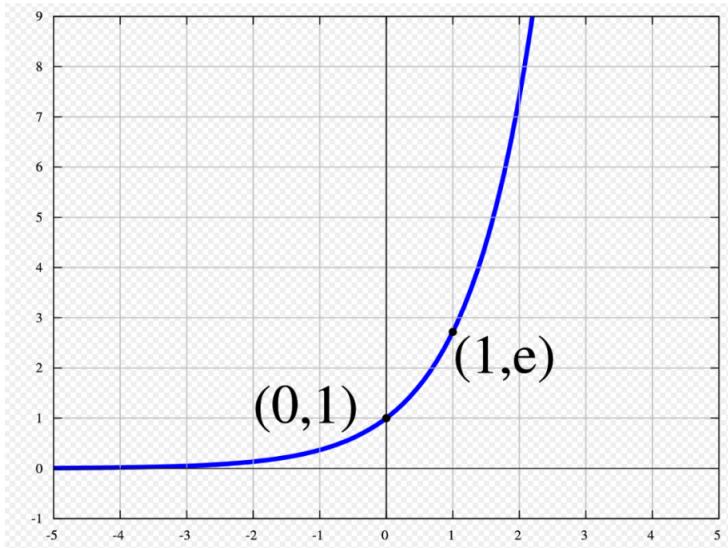
\* 퍼셉트론과 신경망의 주된 차이는 활성화 함수의 차이

- 계단 함수가 0과 1 중 하나의 값만 돌려주는 반면 시그모이드 함수는 실수를 돌려준다.
- 즉, 퍼셉트론에서는 뉴런 사이에 0 혹은 1이 흘렀다면, 신경망에서는 연속적인 실수가 흐른다.
- 두 함수 모두 비선형 함수.
- 신경망에서는 활성화 함수로 비선형 함수를 사용해야 함.

---

## 출력층 설계(3) – 소프트맥스 함수

---



자연 지수 함수  $y = e^x$

발산하는 문제를 막기 위해 max값 조정.

---

## 소프트맥스 함수 구현 시 주의점

---

➤ 오버플로 문제

- 소프트맥스의 지수 함수를 계산할 때 어떤 중수를 더하거나 빼도 결과는 바뀌지 않는다는 것을 이용.
- 오버플로를 막을 목적으로는 입력 신호 중 최댓값을 이용하는 것이 일반적.

$$\begin{aligned}y_k &= \frac{\exp(a_k)}{\sum_{i=1}^n \exp(a_i)} = \frac{C \exp(a_k)}{C \sum_{i=1}^n \exp(a_i)} \\&= \frac{\exp(a_k + \log C)}{\sum_{i=1}^n \exp(a_i + \log C)} \\&= \frac{\exp(a_k + C')}{\sum_{i=1}^n \exp(a_i + C')}\end{aligned}$$

## 소프트맥스 함수의 특징

---

- 소프트맥스 함수의 출력은 0에서 1사이의 실수.
- 출력의 총합은 1.
- 출력을 확률로 해석할 수 있음.
- 신경망을 학습시킬 때는 출력층에서 소프트맥스 함수를 사용.
- 추론 단계에서는 출력층의 소프트맥스 함수를 생략하는 것이 일반적.
  - 각 원소의 대소 관계는 변하지 않음.
  - 지수 함수  $y = e^x$  가 단조 증가 함수이기 때문.

---

## MNIST 데이터셋

---

- Modified National Institute of Standards and Technology.
- 손으로 직접 쓴 숫자(필기체 숫자)들로 이루어진 데이터셋.
- 0 ~ 9까지의 숫자 이미지로 구성되며, 60,000개의 트레이닝 데이터와 10,000개의 테스트 데이터로 이루어져 있음.
- 28 x 28 size의 흑백(1ch) 이미지 데이터.

The screenshot shows a search result for 'Modified National Institute of Standards and Technology'. The page title is 'Modified National Institute of Standards and Technology' with a subtitle '마더파이드 내셔널 인스티튜트 어브 스텠더즈 언드 텍날러지'. Below the title is a snippet of text: '수정된 국립 표준 및 기술 연구소' followed by '55 / 5000'. At the bottom are several icons: a speaker icon, a document icon, a star icon, and a link icon. A green button labeled '번역하기' (Translate) is also visible.

▶ 길이 | 넓이 | 무게 | 부피 | 온도 | 압력 | 속도 | 연비 | 데이터양 | 시간

인치 (in) → 센티미터 (cm)

28 in = 71.12 cm

711.2 밀리미터(mm)	71.12 센티미터(cm)	0.7112 미터(m)
0.000711 킬로미터(km)	28 인치(in)	2.333333 피트(ft)
0.777778 야드(yd)	0.000442 마일(mile)	2.34696 자(尺)
0.39116 간(間)	0.006519 정(町)	0.001811 리(里)
0.000384 해리(海里)		

### Fashion Mnist Challenge

Design and learn machine learning models to classify Fashion Mnist images

↳ <https://www.kaggle.com/c/insar-fashion-mnist-challenge>



Google search results for "mnist fashion dataset".

Search terms: mnist fashion dataset

Results:

- Machine Learning Mastery Deep Learning CNN for Fashion-MNIST
- TensorFlow fashion\_mnist | TensorFlow
- Codelab 9. Fashion MNIST 이미지...
- Datasets - Activeloop Fashion MNIST Dataset | Machine Learning D...
- Medium How To Import and Plot The Fashion MNIST...
- Towards Data Science Multi-Label Classification and Class Activ...
- ResearchGate Sample Images from Fashion-MNIST dataset | Down...
- wikicca.net 강의 00 패션 MNIST 웹 데이터셋 소...
- DEV Community Machine Learning - Fashion MNIST dataset - Prologue
- Katie Codex Exploring image classification with Fashion-MN...
- Label Description Examples
- Medium Classifying Fashion\_Mnist...
- John Bosco FashionMNIST classificatio...

# 01\_fashionMNIST\_LogisticRegression

## Multi-Label Classification and Class Activation Map on Fashion-MNIST

Fashion-MNIST is a fashion product image dataset for benchmarking machine learning algorithms for computer vision. This dataset comprises 60,000 28x28 training images and 10,000 28x28 test images, including 10 categories of

<https://towardsdatascience.com/multi-label-classification-and-class-activation-map-on-fashion-mnist-1454f09f5925>



```
: import tensorflow as tf
(train_input, train_target), (test_input, test_target) = tf.keras.datasets.fashion_mnist.load_data()

Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-labels-idx1-ubyte.gz
29515/29515 [=====] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-images-idx3-ubyte.gz
26421880/26421880 [=====] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-labels-idx1-ubyte.gz
5148/5148 [=====] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-images-idx3-ubyte.gz
4422102/4422102 [=====] - 0s 0us/step
```

5년차 딥러닝 경력은 거짓말이다. 딥마인드가 아닌 이상은 국내에 절대 없다.

IT는 평생 공부하는 직종.

혼·하·파이썬)

## 반복문을 사용한 리스트 생성 [소스 코드 for\\_list01.py](#)

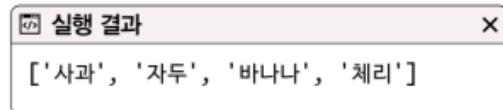
```
01 # 변수를 선언합니다.
02 array = []
03
04 # 반복문을 적용합니다.
05 for i in range(0, 20, 2):
06     array.append(i * i)
07 # 출력합니다.
08 print(array)
```

실행 결과  
[0, 4, 16, 36, 64, 100, 144, 196, 256, 324]

## 조건을 활용한 리스트 내포

소스 코드 array\_comprehensions.py

```
01 # 리스트를 선언합니다.  
02 array = ["사과", "자두", "초콜릿", "바나나", "체리"]  
03 output = [fruit for fruit in array if fruit != "초콜릿"]  
04  
05 # 출력합니다.  
06 print(output)
```



### ▶ 질문있어요!



Q 코드가 너무 길어서 복잡해 보이는데 줄바꿈하면 안 되나요?

A 한 줄로 입력했을 때 코드가 너무 길어서 코드 가독성이 떨어지면 다음과 같이 3줄로 작성할 수 있습니다.

```
리스트 이름 = [표현식  
    for 반복자 in 반복할 수 있는 것  
    if 조건문]
```

```
]: print(train_input.shape, train_target.shape)  
# tuple은 1차원일 때는 (60000,) -> 이런 형태로 보여준다.  
# 아니면 출력할 때 ()모양이 정수형인지 아닌지 모르기 때문에 (,)로 보여줌.  
(60000, 28, 28) (60000,)
```

파이썬에는 tuple, list, dictionary만 있다.

파이썬에서는 배열은 없기 때문에, 배열 사용은 numpy라는 패키지를 사용해야한다.

```
: print(test_input.shape, test_target.shape) # test  
# (10000, 28, 28) 데이터는 10000개, 이미지 사이즈는 28px, 28px이라는 뜻.  
(10000, 28, 28) (10000,)
```

```
: import matplotlib.pyplot as plt  
  
fig, axs = plt.subplots(1,10, figsize=(10,10)) # figsize 인자 단위로 출력  
  
for i in range(10):  
    axs[i].imshow(train_input[i], cmap='gray_r')  
    axs[i].axis('off')  
  
plt.show()
```



```
: print([train_target[i] for i in range(10)]) # 9번 신발, 0번 티셔츠, 7번 운동화, 5번 구두  
[9, 0, 0, 3, 0, 2, 7, 2, 5, 5]
```

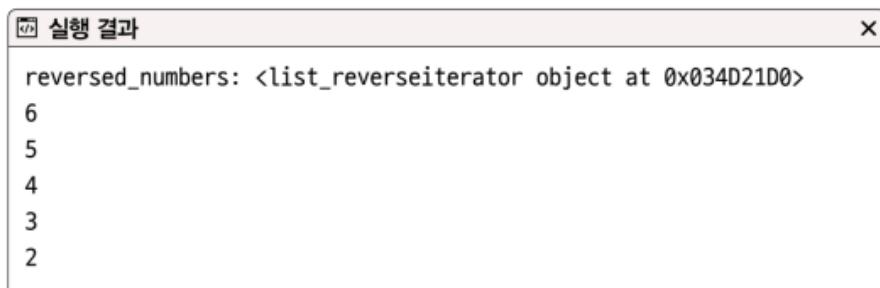
r,g,b 3차원 색, 흑백은 1차원

### if 조건문과 긴 문자열 [소스 코드 if\\_string02.py](#)

```
01 # 변수를 선언합니다.  
02 number = int(input("정수 입력> "))  
03  
04 # if 조건문으로 짤수 짝수를 구분합니다.  
05 if number % 2 == 0:  
06     print("입력한 문자열은 {}입니다.\n{}는(은) 짝수입니다.".format(number, number))  
07 else:  
08     print("입력한 문자열은 {}입니다.\n{}는(은) 홀수입니다.".format(number, number))
```

## reversed() 함수와 이터레이터 [소스 코드 iterator01.py](#)

```
01 # 변수를 선언합니다.  
02 numbers = [1, 2, 3, 4, 5, 6]  
03 r_num = reversed(numbers)  
04  
05 # reversed_numbers를 출력합니다.  
06 print("reversed_numbers :", r_num)  
07 print(next(r_num))  
08 print(next(r_num))  
09 print(next(r_num))  
10 print(next(r_num))  
11 print(next(r_num))
```



min-max - 0 ~ 1 안에서 프레임을 만들어줌.

표준화 - 평균을 0으로, 표준편차를 1로 맞춰서 해주는 것

## 이미지 데이터 처리 방법

### 로지스틱 회귀로 패션 아이템 분류하기

```
: train_scaled = train_input / 255.0 # 색상의 최대값을 255로 맞춰줌.  
train_scaled = train_scaled.reshape(-1, 28 * 28) # 28*28=784 # 열에 맞춰서 -1행은 알아서 늘려가라 명령하는 것.  
# 컬러일 경우엔 28*28*3을 또 해줘야 함. # 이미지는 784개로 열을 생성해줄.  
train_scaled.shape  
(60000, 784)
```

0일 경우에는 사라져버리기 때문에 1-y로

# Cost function

$$cost(W) = \frac{1}{m} \sum c(H(x), y)$$

$$C(H(x), y) = \begin{cases} -\log(H(x)) & : y = 1 \\ -\log(1 - H(x)) & : y = 0 \end{cases}$$

$$C(H(x), y) = -y \log(H(x)) - (1 - y) \log(1 - H(x))$$

---

## inconsistent

미국·영국 [ɪn'kən'sɪstənt] ↗ 영국식 ↗

(형용사)

1 (진술 등이) 내용이 다른[모순되는]

The report is inconsistent with the financial statements. ↗  
그 보고서는 재무제표와 내용이 안 맞는다.

2 (규범 등에) 부합하지 않는

Her behaviour was clearly inconsistent with her beliefs. ↗  
그녀의 행동은 분명히 그녀의 신념에 부합하지 않았다.

3 너무 자주 변하는, 일관성 없는 (↔consistent)

inconsistent results ↗  
일관성 없는 결과들

영어사전 다른 뜻 1

```
: from sklearn.model_selection import cross_val_score
from sklearn.linear_model import SGDClassifier
import numpy as np

sc = SGDClassifier(loss='log', max_iter=10, random_state=42) # random_state는 dataset을 random하게 쓸겠다는 것.

scores = cross_val_score(sc, train_scaled, train_target, n_jobs=-1)
# cross_val_score는 교차검증으로 (모든 데이터를 1번씩) 돌리는 것
# n_jobs 코어를 몇 개를 돌릴지.. -1이면 모든 코어를 사용하는 것을 말함.
print(np.mean(scores['test_score'])) # 10개값의 평균을 구해서 출력으로 확인해봄.
```

0.8313833333333334

## 02\_fashionMNIST\_ANN

```
[1]: import tensorflow as tf
from sklearn.model_selection import train_test_split

(train_input, train_target), (test_input, test_target) = tf.keras.datasets.fashion_mnist.load_data()

train_scaled = train_input / 255.0
train_scaled = train_scaled.reshape(-1, 28 * 28)
```

```
[2]: train_scaled, val_scaled, train_target, val_target = train_test_split(train_scaled, train_target, test_size=0.2, random_state=42)
```

```
[3]: print(train_scaled.shape, train_target.shape)
print(val_scaled.shape, val_target.shape)

(48000, 784) (48000,)
(12000, 784) (12000,)
```

```
[4]: model = tf.keras.Sequential()
model.add(tf.keras.layers.Dense(units=10, input_shape=(28*28,), activation='softmax'))
# input_dim이 아닌 input_shape으로 하니깐 가능해짐. 둘이 같음.
```

### sparse

미국식[spa:rs] ↗ 영국식[spa:s] ↗

(형용사)

(흔히 넓은 지역에 분포된 정도가) 드문, (밀도가) 희박한

the sparse population of the islands ↗

그 섬들의 희박한 인구 밀도

[영어사전 결과 더보기](#)

```
: model.compile(loss='sparse_categorical_crossentropy', metrics='accuracy')
# sparse_categorical_crossentropy 원핫으로 처리할 때 알아서 one-hot 인코딩으로 바꿔줌.
```

```
[16]: model.fit(train_scaled, train_target, epochs=10)

Epoch 1/10
1500/1500 [=====] - 2s 938us/step - loss: 0.6053 - accuracy: 0.7956
Epoch 2/10
1500/1500 [=====] - 1s 947us/step - loss: 0.4784 - accuracy: 0.8406
Epoch 3/10
1500/1500 [=====] - 1s 922us/step - loss: 0.4563 - accuracy: 0.8487
Epoch 4/10
1500/1500 [=====] - 1s 932us/step - loss: 0.4443 - accuracy: 0.8514
Epoch 5/10
1500/1500 [=====] - 2s 1ms/step - loss: 0.4358 - accuracy: 0.8554
Epoch 6/10
1500/1500 [=====] - 1s 972us/step - loss: 0.4329 - accuracy: 0.8565
Epoch 7/10
1500/1500 [=====] - 1s 941us/step - loss: 0.4271 - accuracy: 0.8597
Epoch 8/10
1500/1500 [=====] - 1s 976us/step - loss: 0.4243 - accuracy: 0.8612
Epoch 9/10
1500/1500 [=====] - 2s 1ms/step - loss: 0.4219 - accuracy: 0.8622
Epoch 10/10
1500/1500 [=====] - 2s 1ms/step - loss: 0.4201 - accuracy: 0.8617
```

```
[16]: <keras.callbacks.History at 0x1d6c89a4848>
```

```
def print_n_times(value, n): → 매개변수를 2개 지정했는데
    for i in range(n):
        print(value)

# 함수를 호출합니다.
print_n_times("안녕하세요") → 하나만 넣었습니다.
```

코드를 실행하면 다음과 같이 오류를 출력합니다. 'print\_n\_times()'라는 함수의 매개변수 n이 없다'라는 오류입니다. 처음 프로그래밍을 할 때 자주 발생하는 오류이므로 눈도장을 꼭 찍어 두세요.

### 오류

```
Traceback (most recent call last):
  File "test5_01.py", line 6, in <module>
    print_n_times("안녕하세요")
TypeError: print_n_times() missing 1 required positional argument: 'n'
```

매개 변수 하나가 빠짐

```
print_n_times() missing 1 required positional argument: 'n'
```

```
def print_n_times(value, n): → 매개변수를 2개 지정했는데  
    for i in range(n):  
        print(value)  
  
# 함수를 호출합니다.  
print_n_times("안녕하세요", 10, 20) → 3개를 넣었습니다.
```

코드를 실행하면 다음과 같이 오류를 출력합니다. 'print\_n\_times()' 함수는 2개의 매개변수가 필요한데 3개가 들어왔다'라는 오류입니다. 따라서 함수를 호출할 때는 함수를 선언할 때와 같은 개수의 매개변수를 입력해야 합니다.

#### 오류

```
Traceback (most recent call last):  
  File "test5_02.py", line 6, in <module>  
    print_n_times("안녕하세요", 10, 20)  
TypeError: print_n_times() takes 2 positional arguments but 3 were given
```

## positional

미국·영국 [pə'zɪʃənl] ⓘ 영국식 ⓘ

(형용사)

위치와 관련된, 위치상의

```
]: model.evaluate(val_scaled, val_target) # 실질적인 test 데이터(한번도 사용하지 않은)  
375/375 [=====] - 0s 918us/step - loss: 0.4433 - accuracy: 0.8548  
]: [0.4432770609855652, 0.8548333048820496]
```

## 가변 매개변수 함수 [소스 코드](#) variable\_param.py

```
01 def print_n_times(n, *values):
02     # n번 반복합니다.
03     for i in range(n):
04         # values는 리스트처럼 활용합니다.
05         for value in values:
06             print(value)
07         # 단순한 줄바꿈
08         print()
09
10    # 함수를 호출합니다.
11 print_n_times(3, "안녕하세요", "즐거운", "파이썬 프로그래밍")
```

실행 결과
안녕하세요
즐거운
파이썬 프로그래밍
안녕하세요
즐거운
파이썬 프로그래밍
안녕하세요
즐거운
파이썬 프로그래밍

## 선형 회귀

평균 제곱 오차(mean squared error, MSE)

- 가장 많이 쓰이는 손실 함수

$$MSE = \frac{1}{N} \sum_{i=1}^N (p_i - y_i)^2$$

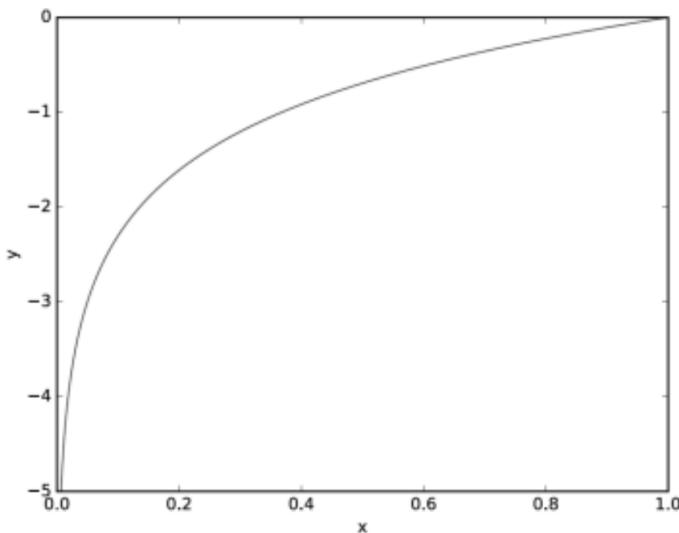
## 로지스틱 회귀

교차 엔트로피 오차(cross entropy error, CEE)

$$CEE = - \sum_{i=1}^n t_i \times \log_e(y_i) , \text{데이터 하나에 대한 손실 함수}$$

## 자연로그 $y = \log x$ 의 그래프

---



- $x$ 가 1일 때  $y$ 는 0이 되고,  $x$ 가 0에 가까워질수록  $y$ 의 값은 점점 작아짐.

일부 데이터를 가지고 와서 빠르게 **Stochastic Gradient Descent**하는 것.

## 미니배치 학습

---

### - 미니배치(mini-batch) 학습

- 모든 훈련 데이터를 대상으로 손실 함수의 합을 구하려면 많은 시간이 걸리고, 현실적이지 않음.
  - 훈련 데이터 중 일부(미니배치)만 골라 학습을 수행.
  - 가령 60,000장의 훈련 데이터 중에서 100장을 무작위로 뽑아 학습(미니배치 학습)하는 것.
- 미니배치의 손실 함수 계측을 통해 전체 훈련 데이터의 '근사치'로 이용.

## 학습 알고리즘 구현하기 – 확률적 경사 하강법(SGD)

---

### 확률적 경사 하강법(Stochastic Gradient Descent, SGD)

- 데이터를 미니배치로 무작위로 선정하여 경사 하강법으로 매개변수를 갱신하는 방법.

### 신경망 학습 절차

- 전제 – 학습
  - 가중치와 편향을 훈련 데이터에 적응하도록 조정하는 과정.
- 1단계 – 미니배치
  - 훈련 데이터 중 일부를 무작위로 가져오는 미니배치를 통해 손실 함수 값을 줄이는 것이 목표.
- 2단계 – 기울기 산출
  - 미니배치의 손실 함수 값을 줄이기 위해 각 가중치 매개변수의 기울기를 구함.
  - 기울기는 손실 함수의 값을 가장 크게 하는 방향을 제시함.
- 3단계 – 매개변수 갱신
  - 가중치 매개변수를 기울기 방향으로 아주 조금 갱신함.
- 4단계 – 반복
  - 1 ~ 3단계를 반복함.

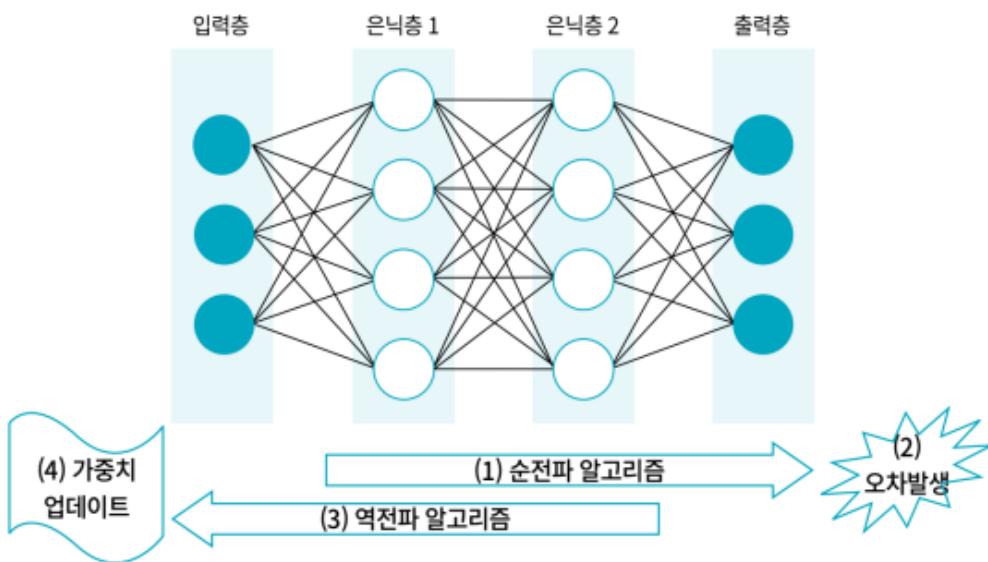
---

### ③ 인공신경망 학습(역전파 알고리즘)

- 인공신경망은 여러 개의 퍼셉트론으로 구성되어 있기 때문에 각 퍼셉트론이 보유한 여러 개의 가중치  $w_i$  값의 결정이 중요하다. 인공신경망은 지도학습의 한 종류로 입력층(독립변수)과 출력층(반응 변수)의 데이터에 따른 이상적인 가중치  $w_i$  값을 결정해야 한다.
- 가중치 값의 결정은 입력층에서 출력층으로 찾아 나가는 순전파 알고리즘을 먼저 활용한다. 이때 발생한 오차들을 줄이고자 출력층에서 입력층 방향으로 거꾸로 찾아 나가는 역전파 알고리즘을 활용하여 가중치 값들을 새롭게 조정한다. 훈련용(train) 데이터의 자료들이 순차적으로 입력될 때마다 가중치가 새롭게 조정되는 것을 인공신경망이 학습한다고 표현한다. 이때 전체 자료들에 의하여 학습이 한 번 되는 것을 1 epoch라 하면 일정 수의 epoch에 도달하거나 혹은 원하는 수준의 정확도를 얻을 때까지 위 작업을 반복한다.

### 【 인공신경망의 학습 과정 】

### 【 인공신경망의 계층 구조 】



```
def print_n_times(n=2, *values):
    # n번 반복합니다.
    for i in range(n):
        # values는 리스트처럼 활용합니다.
        for value in values:
            print(value)
        # 단순한 줄바꿈
        print()

    # 함수를 호출합니다.
print_n_times("안녕하세요", "즐거운", "파이썬 프로그래밍")
```

매개변수가 순서대로 입력되므로 n에는 "안녕하세요"가 들어가고, values에는 ["즐거운", "파이썬 프로그래밍"]이 들어옵니다. 그런데 range() 함수의 매개변수에는 숫자만 들어올 수 있으므로 다음과 같은 오류가 발생합니다.

### 오류

```
Traceback (most recent call last):
  File "test5_03.py", line 11, in <module>
    print_n_times("안녕하세요", "즐거운", "파이썬 프로그래밍")
  File "test.py", line 3, in print_n_times
    for i in range(n):
TypeError: 'str' object cannot be interpreted as an integer
```

따라서 기본 매개변수는 가변 매개변수 앞에 써도 의미가 없다는 것을 기억해 주세요.

```
def print_n_times(*values, n=2):
    # n번 반복합니다.

    for i in range(n):
        # values는 리스트처럼 활용합니다.

        for value in values:
            print(value)

        # 단순한 줄바꿈
        print()

    # 함수를 호출합니다.

print_n_times("안녕하세요", "즐거운", "파이썬 프로그래밍", 3)
```

```
안녕하세요
즐거운
파이썬 프로그래밍
3
```

```
안녕하세요
즐거운
파이썬 프로그래밍
3
```

## 키워드 매개변수 [소스 코드](#) param\_keyword01.py

```
01 def print_n_times(*values, n=2):
02     # n번 반복합니다.
03     for i in range(n):
04         # values는 리스트처럼 활용합니다.
05         for value in values:
06             print(value)
07         # 단순한 줄바꿈
08         print()
09
10    # 함수를 호출합니다.
11    print_n_times("안녕하세요", "즐거운", "파이썬 프로그래밍", n=3)
```

```
실행 결과
안녕하세요
즐거운
파이썬 프로그래밍

안녕하세요
즐거운
파이썬 프로그래밍

안녕하세요
즐거운
파이썬 프로그래밍
```

→ 키워드 매개변수입니다.

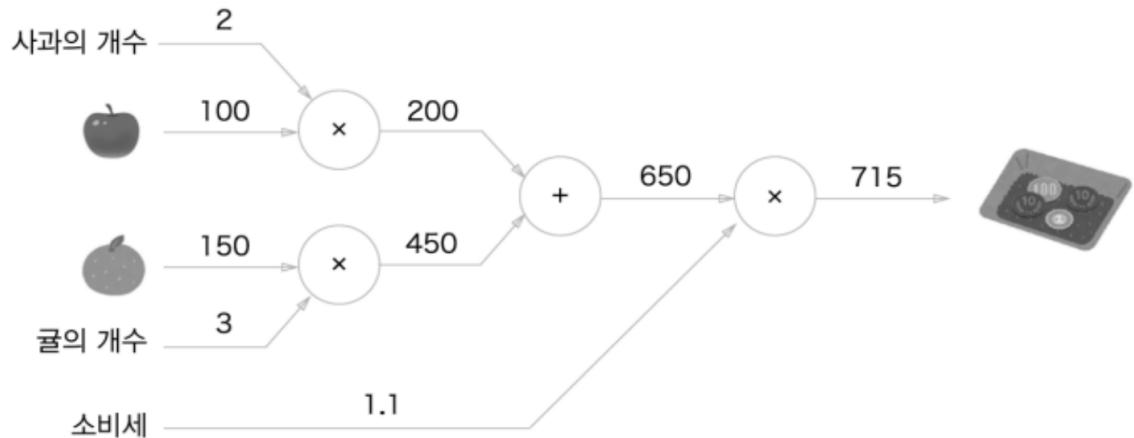
## 여러 함수 호출 형태 [소스 코드](#) param\_examples.py

```
01 def test(a, b=10, c=100):
02     print(a + b + c)
03
04 # 1) 기본 형태
05 test(10, 20, 30)
06 # 2) 키워드 매개변수로 모든 매개변수를 지정한 형태
07 test(a=10, b=100, c=200)
08 # 3) 키워드 매개변수로 모든 매개변수를 마구잡이로 지정한 형태
09 test(c=10, a=100, b=200)
10 # 4) 키워드 매개변수로 일부 매개변수만 지정한 형태
11 test(10, c=200)
```

```
실행 결과
60
310
310
220
```

## 계산 그래프

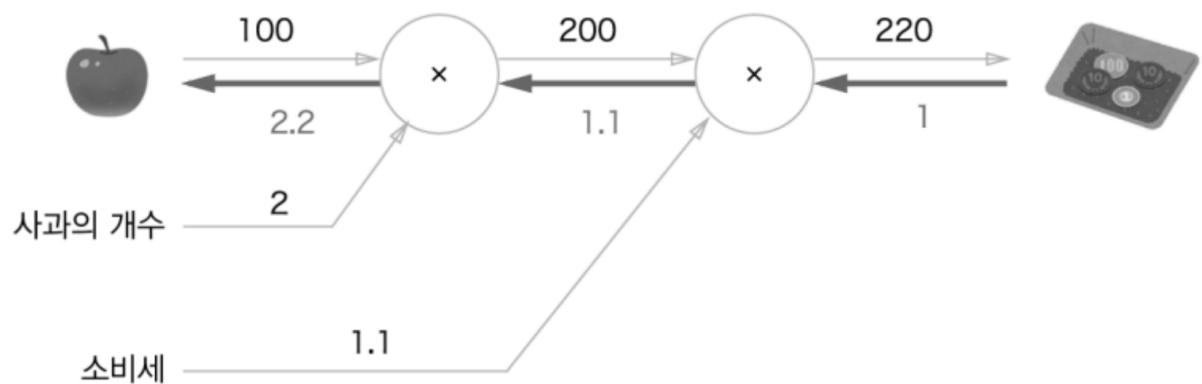
- 문제2 : 현빈군은 슈퍼에서 사과를 2개, 귤을 3개 샀습니다. 사과는 1개에 100원, 귤은 1개 150원입니다. 소비세가 10%일 때 지불 금액을 구하세요.



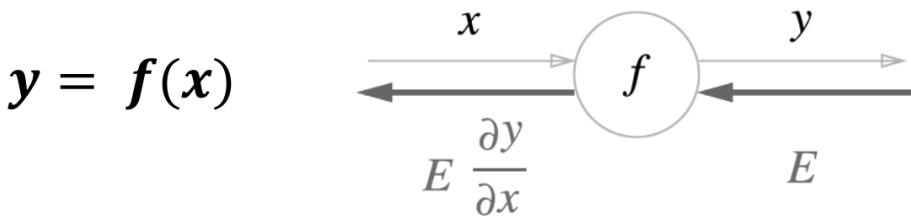
역전파로  $1.1 * 1$

역전파로 2개 \* 1.1

## 역전파에 의한 미분 값의 전달



## 계산 그래프의 역전파



### ● 역전파의 계산 절차

- 신호  $E$ 에, 노드의 국소적 미분( $\frac{\partial y}{\partial x}$ )을 곱한 후 다음 노드로 전달 수행.
- 국소적 미분이란 : 순전파 때의  $y = f(x)$  계산의 미분을 구한다는 것이며, 이는  $x$ 에 대한  $y$ 의 미분( $\frac{\partial y}{\partial x}$ )을 구한다는 뜻.

## 연쇄법칙이란?

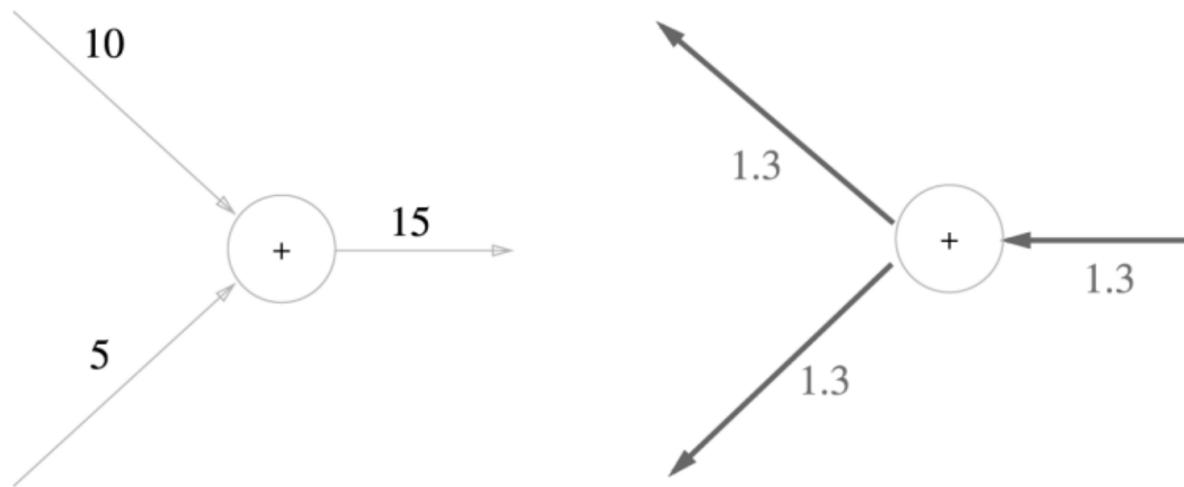
- 합성 함수의 미분은 합성 함수를 구성하는 각 함수의 미분의 곱으로 나타낼 수 있다.
- 합성 함수 : 여러 함수로 구성된 함수.

$$z = (x + y)^2$$

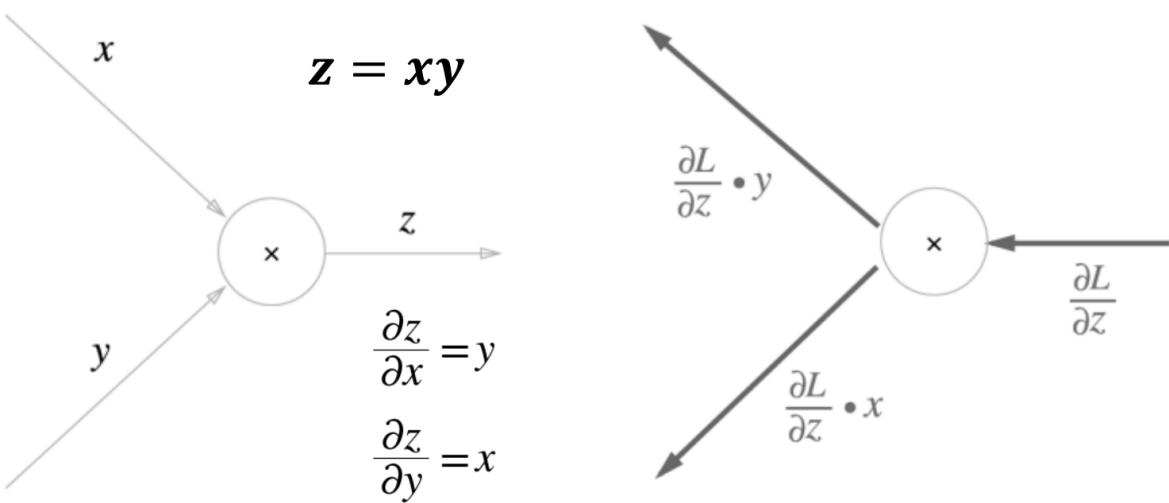
↓

$$\left\{ \begin{array}{l} z = t^2 \\ t = x + y \end{array} \right\}$$
$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial t} \frac{\partial t}{\partial x}$$
$$\frac{\partial z}{\partial t} = 2t$$
$$\frac{\partial t}{\partial x} = 1$$
$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial t} \frac{\partial t}{\partial x} = 2t \cdot 1 = 2(x + y)$$

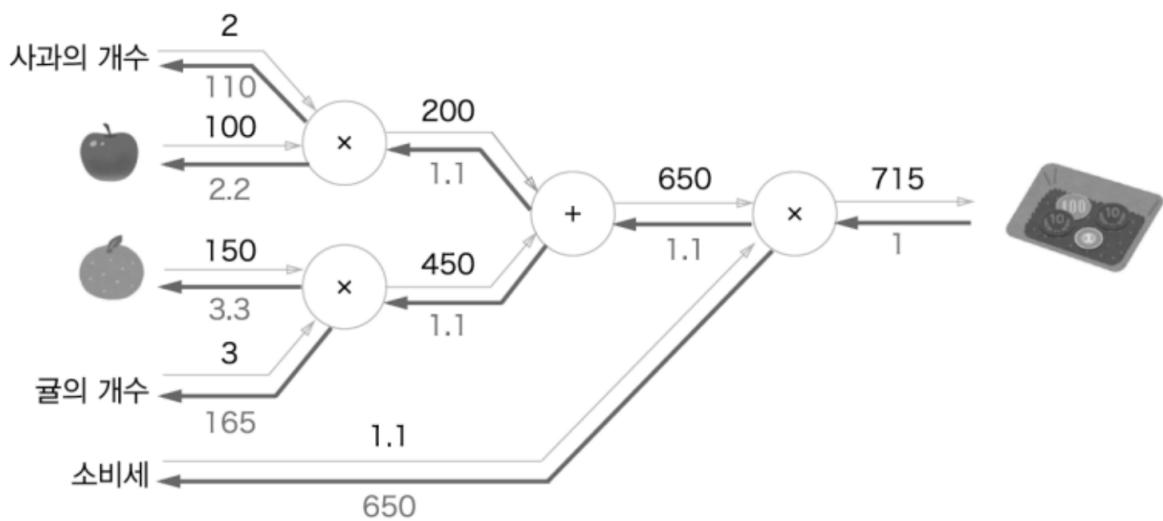
## 덧셈 노드의 역전파의 예



## 곱셈 노드의 역전파



## 덧셈 계층 – 사과 2개와 귤 3개 구입



지수함수는 미분하면 자기 자신이다.