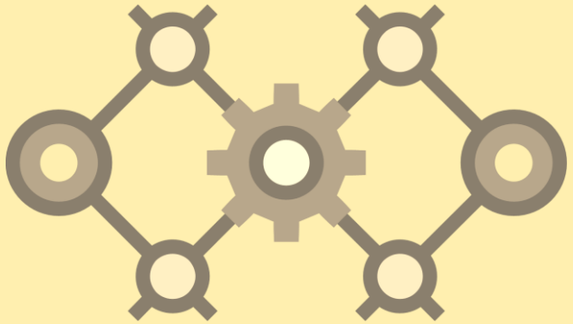
Python을 활용한 이미지 분석

2. 심층 신경망 모델 구현

기상청

기상청



심층 신경망 모델 구현

01 텐서플로와 케라스

02 활성화 함수

03 모델 만들기

1

텐서플로와 케라스

1. 텐서플로와 케라스

기상청

정의

케라스 (Keras)	파이썬으로 작성된 오픈 소스 신경망 라이브러리

개발배경

- 케라스는 구글의 엔지니어 프랑소아 숄레(Francois Chollet)의 주도하에 만들어짐
- 2017년, 텐서플로의 코어 라이브러리에 케라스를 지원하기로 결정함

특징

- 높은 수준의 직관적인 추상화 집합을 표현함으로써 신경망을 구성하기 쉽게 만들어 줌

TensorFlow

기상청

케라스 API

케라스 시퀀셜 API를 통한 모델 정의와 학습

```
model = keras.Sequential([])
model.compile()
model.fit()
model.evaluate()
```

기상청

실습

예제

문제 상황	실습 코드
tensorflow의 keras 로드	import tensorflow as tf from tensorflow import keras

예시 화면

```
# tensorflow와 tf.keras를 임포트합니다
import tensorflow as tf
from tensorflow import keras
```

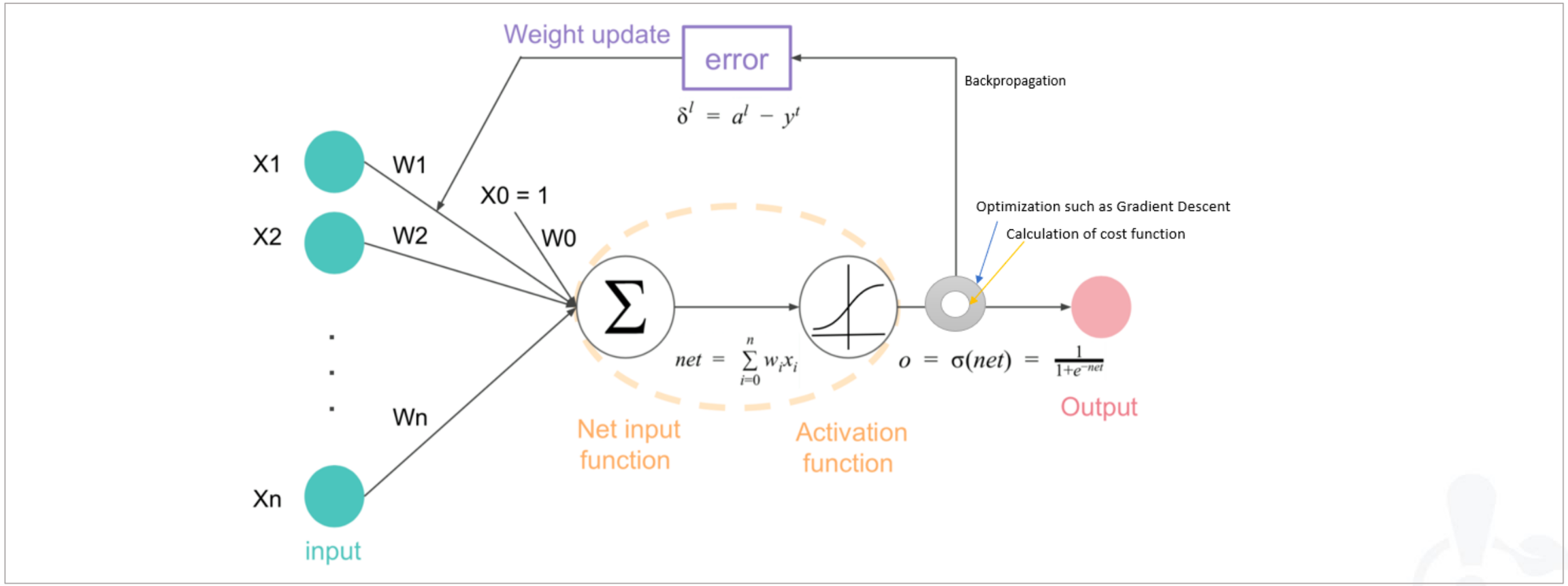
2

활성화 함수

2. 활성화 함수

오차 역전파법(Backpropagation)

가중치와 편향 수정



[출처 : StackExchange. https://datascience.stackexchange.com]

2. 활성화 함수

기상청

기울기 소실(Gradient Vanishing)

문제

01 오차 역전파를 통해 가중치를 최적화하는 과정에서 입력층에 가까워질수록 가중치가 잘 학습되지 않는 문제가 발생함

02 오차 역전파의 계산 과정에서 층마다 활성화 함수의 기울기를 곱하는데 작은 값을 곱하게 되면 0에 가까워짐

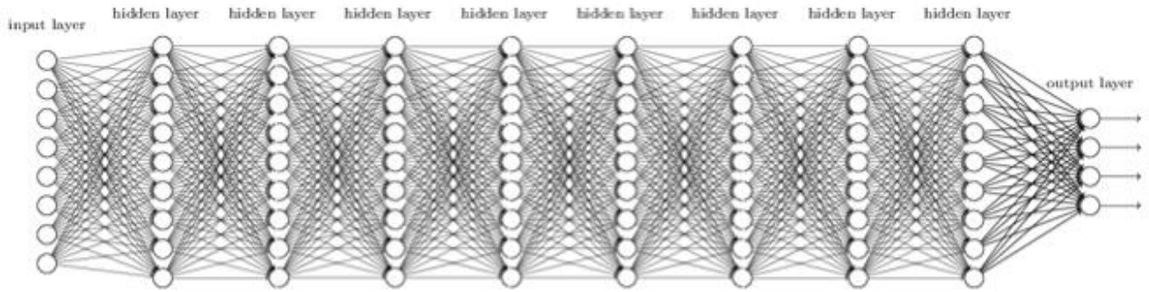
➡ 기울기가 0에 가까우면 기울기의 크기에 학습률을 곱한 결과가 0에 가까워져서 제대로 학습이 되지 않음

2. 활성화 함수

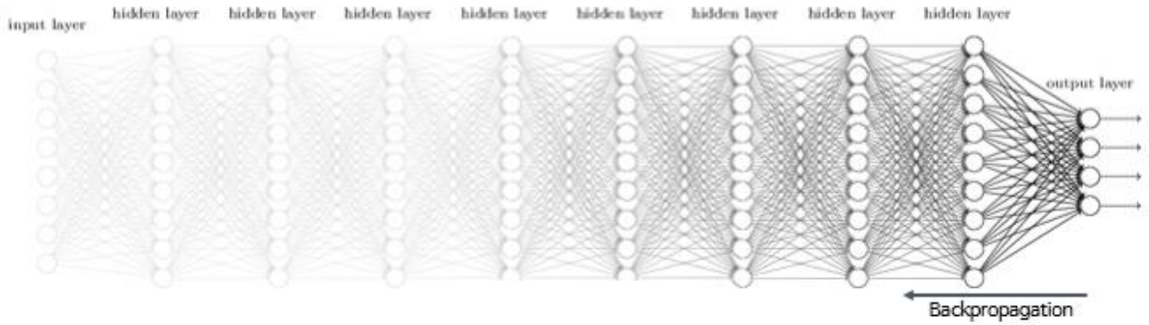
기울기 소실(Gradient Vanishing)

문제

Deep Neural Network



Vanishing Gradient

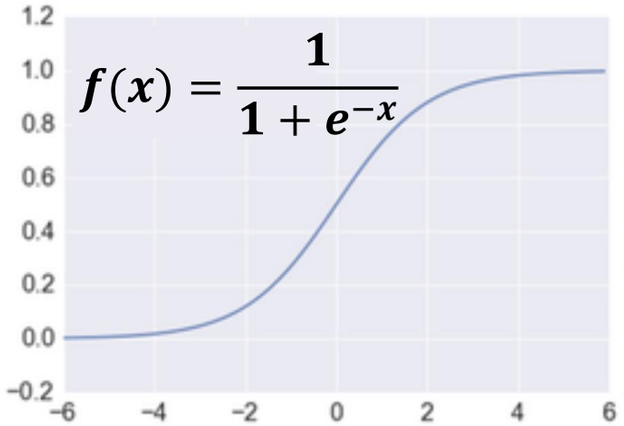


2. 활성화 함수

기울기 소실(Gradient Vanishing)

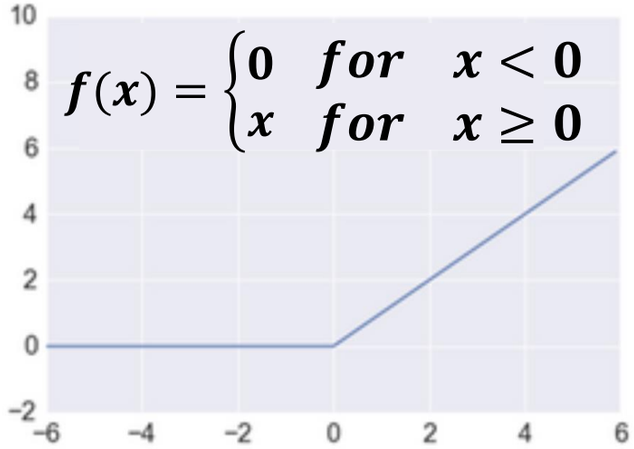
한계와 개선

Sigmoid



- 은닉층을 지날 때마다 작은 값을 미분하면 입력층에 가까워졌을 때는 기울기가 점점 0에 가까워짐

ReLU



- 값이 0을 넘을 때는 미분값이 항상 1이기 때문에 많은 은닉층이 있어도 기울기가 소실되지 않고 잘 전달됨

2. 활성화 함수

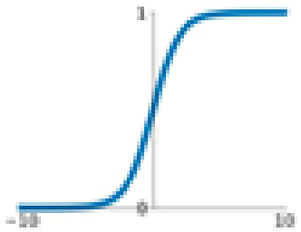
기상청

다양한 활성화 함수

Activation Functions

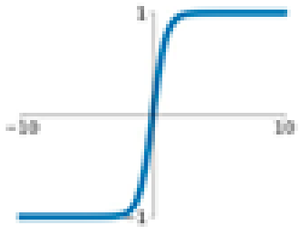
Sigmoid

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



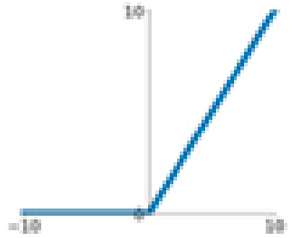
tanh

$$tanh(x)$$



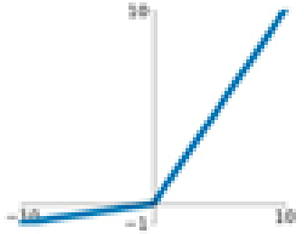
ReLU

$$max(0, x)$$



Leaky ReLU

$$max(0.1x, x)$$

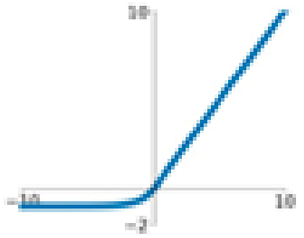


Maxout

$$max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

$$\begin{cases} x & x \geq 0 \\ a(e^x - 1) & x < 0 \end{cases}$$



2. 활성화 함수

기상청

실습

예제

문제 상황	실습 코드
텐서플로에 내장된 활성화 함수 찾기	dir(tf.keras.activations)

예시화면

```
['elu', 'exponential', 'get', 'hard_sigmoid',
'linear', 'relu', 'selu', 'serialize', 'sigmoid',
'softmax', 'softplus', 'softsign', 'swish', 'tanh']
```

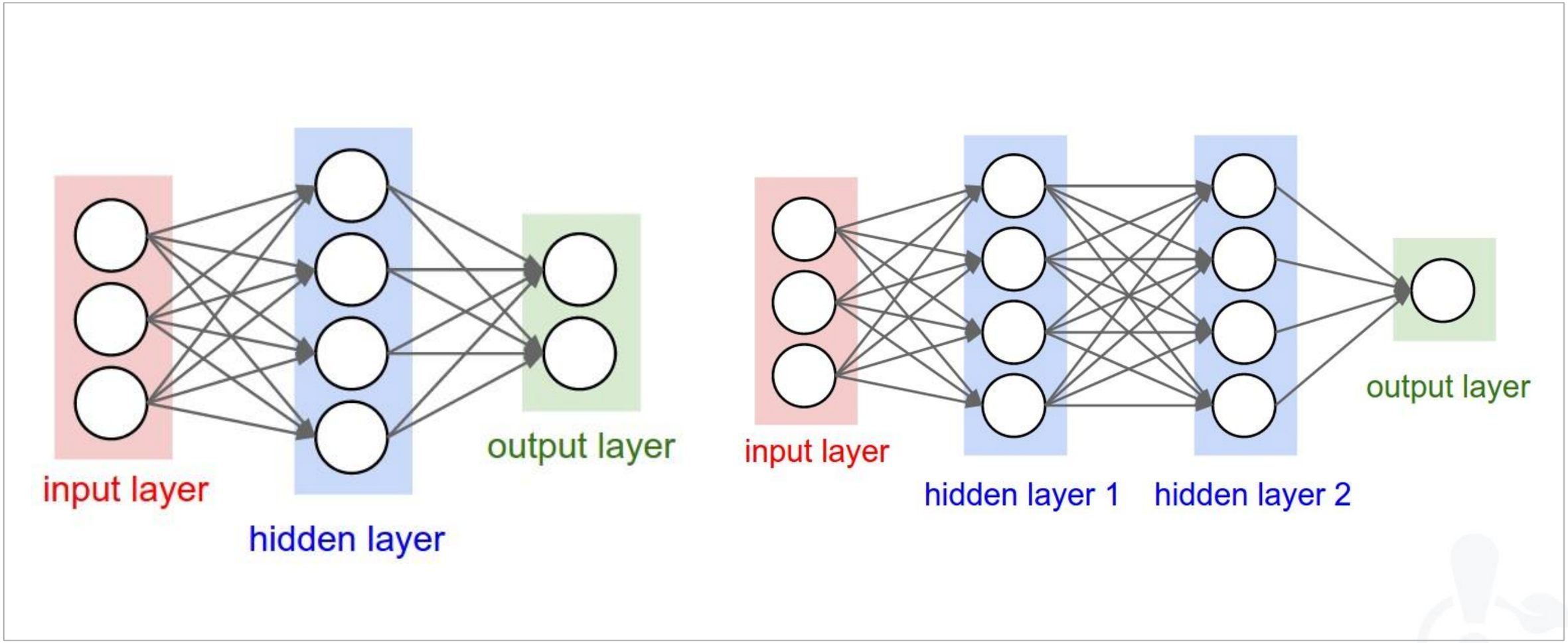
3

모델 만들기

3. 모델 만들기

기상청

심층 신경망 구조



[출처 : Stanford University CS231n, https://cs231n.github.io]

3. 모델 만들기

기상청

Sequential 모델

Sequential 모델을 통한 네트워크 구성

```
model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(input_shape=(28, 28)),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dense(10, activation='softmax')
])
```

3. 모델 만들기

기상청

출력층

분류의 예측 값이 n개일 때

```
tf.keras.layers.Dense(n, activation='softmax')
```

- n개의 확률을 반환하고 반환된 값의 전체 합은 1
- 각 노드는 현재 이미지가 n개 클래스 중 하나에 속할 확률을 출력함

분류의 예측 값이 둘 중 하나일 때

```
tf.keras.layers.Dense(1, activation='sigmoid')
```

- 둘 중 하나를 예측할 때 1개의 출력값을 출력함
- 확률을 받아 임계 값 기준으로 True, False로 나눔

3. 모델 만들기

기상청

출력층

회귀일 때

```
tf.keras.layers.Dense(1)
```

- 특정 값을 출력함

3. 모델 만들기

기상청

네트워크 모델 요약

model.summary()

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
flatten (Flatten)            (None, 784)               0
_________________________________________________________________
dense (Dense)                (None, 128)               100480
_________________________________________________________________
dense_1 (Dense)              (None, 10)                1290
=================================================================
Total params: 101,770
Trainable params: 101,770
Non-trainable params: 0
_________________________________________________________________
```

3. 모델 만들기

기상청

실습

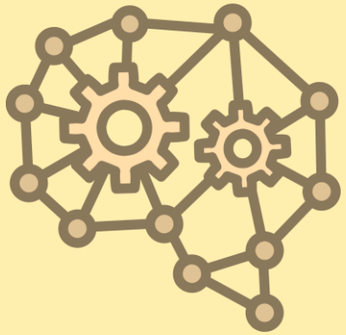
예제

문제 상황	실습 코드
Sequential 모델 레이어 구성	tf.keras.models.Sequential()

예시화면

```
model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(input_shape=(28, 28)),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dense(10, activation='softmax')
])
```

기상청



KEY POINT

▶ 텐서플로와 케라스

- 텐서플로의 코어 라이브러리에 케라스를 지원함
- 케라스는 높은 수준의 추상화로 신경망을 구성하기 쉽게 만들어 줌

▶ 활성화 함수

- 기울기 소실(Gradient Vanishing) 문제
- Sigmoid 사용 시 가중치가 잘 학습되지 않는 문제
- Relu를 사용하여 기울기 소실 문제를 해결

▶ 모델 만들기

- Sequential 모델을 통해 입력층 - 은닉층 - 출력층을 구성함
- model.summary()를 통해 네트워크 모델을 요약함

python™

Python을 활용한 이미지 분석

2. 심층 신경망 모델 구현

"이번 시간을 모두 마치셨습니다.
수고하셨습니다."

THANK YOU

기상청