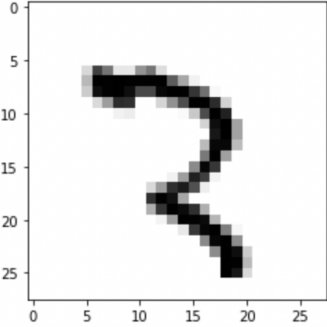


교육 제목	데이터 기반 인공지능 시스템 엔지니어 양성 과정
교육 일시	2021년 10월 27일
교육 장소	YGL C-6 학과장 & 자택(디스코드 이용한 온라인)
교육 내용	
오전	<p>딥러닝 사전 import 주요 라이브러리</p> <pre># 필요한 라이브러리를 불러옵니다. import pandas as pd import tensorflow as tf from sklearn.model_selection import train_test_split import matplotlib.pyplot as plt</pre>
	<p>2진 분류 딥러닝 모델 설정 시 옵션 설정</p> <pre>loss = tf.keras.losses.binary_crossentropy optimizer = tf.keras.optimizers.SGD(learning_rate=0.4) metrics = tf.keras.metrics.binary_accuracy model.compile(loss=loss, optimizer=optimizer, metrics=[metrics])</pre>
	<p>딥러닝 모델 구축 절차</p> <pre># 1. 데이터 불러오기 # 2. 데이터 확인하기 # 3. x,y 나누기 # 4. 전체 데이터에서 학습데이터와 테스트데이터(0.2) 구분 # 5. 딥러닝 설계 ---> kernel_initializer = tf.ke # 6. model.compile # 7. model.fit # 8. 그래프 결과 확인 # 9. test 정확도 확인</pre>
	<p>NN_MNIST</p> <p>1. imshow()</p> <pre>plt.imshow(x_train[500], cmap=plt.cm.binary)</pre> <p><matplotlib.image.AxesImage at 0x7fd92cc00750></p> 

2. 정규화

```
import numpy as np
print('최소값:', np.min(x_train), '최대값:', np.max(x_train))

최소값: 0 최대값: 255

# 정규화
x_train_norm = x_train / 255.0
x_test_norm = x_test / 255.0
print('최소값:', np.min(x_train_norm), '최대값:', np.max(x_test_norm))

최소값: 0.0 최대값: 1.0
```

3. 데이터 shape 변환

```
# reshape 28 * 28 --> 784, 1변환
x_train = x_train_norm.reshape(x_train.shape[0], 784).astype('float32')
x_test = x_test_norm.reshape(x_test.shape[0], 784).astype('float32')

y_train = tf.keras.utils.to_categorical(y_train, 10)
y_test = tf.keras.utils.to_categorical(y_test, 10)
```

4. 모델 구축

```
from tensorflow import keras
model = keras.models.Sequential()
model.add(keras.layers.Dense(16, activation='relu', input_shape=(784,)))
model.add(keras.layers.Dense(32, activation='relu'))
model.add(keras.layers.Dense(10, activation='softmax'))

model.summary()
```

Model: "sequential_1"

Layer (type)	Output Shape	Param #
dense_18 (Dense)	(None, 512)	401920
dense_19 (Dense)	(None, 256)	131328
dense_20 (Dense)	(None, 10)	2570

Total params: 535,818
Trainable params: 535,818
Non-trainable params: 0

5. 모델 컴파일

```
model.compile(optimizer='adam',
              loss = 'categorical_crossentropy',
              metrics = ['accuracy'])

history = model.fit(x_train, y_train,
                    validation_split=0.1, epochs=10, batch_size=200,
                    verbose=1)
```

6. 결과 확인 예측

```
test_loss, test_accuracy = model.evaluate(x_test,
```

	<pre> y_test, verbose=2) print('test loss : {}'.format(test_loss)) print('test_accuracy: {}'.format(test_accuracy)) predicted_result = model.predict(x_test) predicted_labels = np.argmax(predicted_result,axis=1) </pre> <p>7. 그래프</p> <pre> # 테스트 셋의 오차 y_vloss = history.history['val_loss'] # 학습셋의 오차 y_loss = history.history['loss'] # 그래프로 표현 x_len = numpy.arange(len(y_loss)) plt.plot(x_len, y_vloss, marker='.', c="red", label='Testset_loss') plt.plot(x_len, y_loss, marker='.', c="blue", label='Trainset_loss') # 그래프에 그리드를 주고 레이블을 표시 plt.legend(loc='upper right') # plt.axis([0, 20, 0, 0.35]) plt.grid() plt.xlabel('epoch') plt.ylabel('loss') plt.show() </pre>
오후	<p>가위바위보 분류기 만들기 구글 teachablemachine 활용 : https://teachablemachine.withgoogle.com/</p> <ol style="list-style-type: none"> 1. 데이터 준비 2. 딥러닝 네트워크 설계하기 3. 딥러닝 네트워크 학습시키기 4. 테스트 하기 5. 개선 등