|  |  |
| --- | --- |
| 교육 제목 |  |
| 교육 일시 | 2021. 10. 28 |
| 교육 장소 | 영우글로벌러닝 |
| **교육 내용** | |
| 오전 | import tensorflow as tf  import matplotlib.pyplot as plt  import numpy as np  (x\_train, y\_train), (x\_test, y\_test) = tf.keras.datasets.mnist.load\_data()  test\_image = x\_test[0]  test\_image\_reshape = test\_image.reshape(1, 784).astype('float64')  # 28 \* 28이미지를 784개로 reshape하고, 1--> 데이터가 1개다. batchsize  model = tf.keras.models.load\_model('./my\_NN\_Test.h5')  y\_pred = model.predict(test\_image\_reshape)  print(y\_pred)  index = np.argmax(y\_pred)  print(index)  index = np.argmax(y\_pred)  value = y\_pred[:, index]  plt.imshow(test\_image, cmap='Greys')  plt.xlabel(str(index)+" "+str(value))  plt.show()  ----------------------------------------------------------------------------------  import tensorflow as tf  import matplotlib.pyplot as plt  import numpy as np  import cv2  (x\_train, y\_train), (x\_test, y\_test) = tf.keras.datasets.mnist.load\_data()  # 이미지 회전 변환 매트릭스 구하기  M = cv2.getRotationMatrix2D((28/2, 28/2), 0, 1) # Matrix생성  # 이미지 이동 변환 매트릭스 구하기  M[0, 2] = M[0, 2] +3  M[1, 2] = M[1, 2] +3  # 이미지 변환 매트릭스 적용  test\_image = cv2.warpAffine(x\_train[5], M, (28, 28)) #image에 matrix곱  plt.imshow(test\_image, cmap="Greys")  test\_image\_reshape = test\_image.reshape(1,784).astype('float64')  model = tf.keras.models.load\_model('./my\_NN\_Test.h5')  y\_pred = model.predict(test\_image\_reshape)  index = np.argmax(y\_pred)  value = y\_pred[:, index]  plt.imshow(test\_image, cmap='Greys')  plt.xlabel(str(index)+" "+str(value))  plt.show()  print(value)  plt.imshow(x\_train[5], cmap='Greys')  -----------------------------------------------------------------------------  import matplotlib.pyplot as plt  import os  import tensorflow as tf  import numpy as np  (x\_train, y\_train), (x\_test, y\_test) = tf.keras.datasets.mnist.load\_data()  # 데이터 (배치사이즈 x 28 x 28 x 1)로 이미지로 변환 --> 그레이스케일 채널은 1  x\_train = x\_train.reshape(x\_train.shape[0], 28, 28, 1).astype('float32')  x\_test = x\_test.reshape(x\_test.shape[0], 28, 28, 1).astype('float32')  y\_train = tf.keras.utils.to\_categorical(y\_train)  y\_test = tf.keras.utils.to\_categorical(y\_test)  # CNN 모델 설계  input\_layer = tf.keras.layers.Input(shape=(28, 28, 1))  x = tf.keras.layers.Conv2D(32, (3, 3), strides =1, activation='relu', padding='same')(input\_layer)  x = tf.keras.layers.MaxPool2D((2, 2))(x)  x = tf.keras.layers.Conv2D(64, (3, 3), strides=1, activation='relu')(x)  x = tf.keras.layers.MaxPool2D((2, 2))(x)  x = tf.keras.layers.Flatten()(x)  output\_layer = tf.keras.layers.Dense(10, activation='softmax')(x)  model = tf.keras.Model(inputs=[input\_layer], outputs=[output\_layer])  model.summary()  loss = tf.keras.losses.categorical\_crossentropy  optimizer = tf.keras.optimizers.RMSprop(learning\_rate=0.0001)  metrics = tf.keras.metrics.categorical\_accuracy  model.compile(loss=loss,  optimizer = optimizer,  metrics= [metrics])  model\_dir = './cnn\_mnist\_model/'  if not os.path.exists(model\_dir):  os.mkdir(model\_dir)  model\_path='./cnn\_mnist\_model/{epoch:02d}-{val\_loss:.4f}.hdf5'  callback\_list = [tf.keras.callbacks.ModelCheckpoint(filepath=model\_path, monitor='val\_loss', verbose=1, save\_best\_only=True),  tf.keras.callbacks.EarlyStopping(monitor='val\_loss', patience=10)]    history = model.fit(x\_train, y\_train, validation\_split=0.2, epochs=5, batch\_size=200, verbose=1, callbacks=callback\_list) |
| 오후 | import matplotlib.pyplot as plt  import os  import tensorflow as tf  import numpy as np  # MNIST 데이터 불러오기  (x\_train, y\_train), (x\_test, y\_test)=tf.keras.datasets.fashion\_mnist.load\_data()  # (배치사이즈\*28\*28\*1)로 이미지 변환 --> 그레이스케일은 채널1  x\_train=x\_train.reshape(x\_train.shape[0], 28, 28, 1).astype('float32')  x\_test=x\_test.reshape(x\_test.shape[0], 28, 28, 1).astype('float32')  y\_train=tf.keras.utils.to\_categorical(y\_train)  y\_test=tf.keras.utils.to\_categorical(y\_test)  # CNN 모델 설계  input\_layer = tf.keras.layers.Input(shape=(28, 28, 1))  x = tf.keras.layers.Conv2D(32, (3,3), strides=1, activation='relu', padding='same')(input\_layer)  x = tf.keras.layers.MaxPool2D((2, 2))(x)  x = tf.keras.layers.Conv2D(64, (3, 3), strides=1, activation='relu', padding='same')(x)  x = tf.keras.layers.MaxPool2D((2, 2))(x)  x = tf.keras.layers.Flatten()(x)  x = tf.keras.layers.Dense(512, activation='relu')(x)  output\_layer = tf.keras.layers.Dense(10, activation='softmax')(x)  model = tf.keras.Model(inputs=[input\_layer], outputs=[output\_layer])  model.summary()  loss = tf.keras.losses.categorical\_crossentropy  optimize = tf.keras.optimizers.RMSprop(learning\_rate=0.0001)  metric = tf.keras.metrics.categorical\_accuracy  model.compile(loss=loss,  optimizer = optimize,  metrics=[metric])  model\_dir = './cnn\_fasion\_mnist\_model/'  if not os.path.exists(model\_dir):  os.mkdir(model\_dir)  model\_path='./cnn\_fasion\_mnist\_model/{epoch:02d}-{val\_loss:.4f}.hdf5'  callback\_list = [tf.keras.callbacks.ModelCheckpoint(filepath=model\_path, monitor='val\_loss', verbose=1, save\_best\_only=True),  tf.keras.callbacks.EarlyStopping(monitor='val\_loss', patience=10)]  history = model.fit(x\_train, y\_train, validation\_split=0.2, epochs=5, batch\_size=200, verbose=1, callbacks=callback\_list)  #테스트 정확도 출력  print("\n Test Accuracy: %.4f" % (model.evaluate(x\_test, y\_test)[1]))  # 테스트 셋의 오차  y\_vloss = history.history['val\_loss']  # 학습셋의 오차  y\_loss = history.history['loss']  # 그래프로 표현  x\_len = np.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.grid()  plt.xlabel('epoch')  plt.ylabel('loss')  plt.show()  --------------------------------------------------------------------------------  import numpy as np  import tensorflow as tf  from tensorflow import keras  import matplotlib.pyplot as plt  from tensorflow.keras.datasets import cifar10  # 데이터 불러오기  (x\_train, y\_train), (x\_test, y\_test) = cifar10.load\_data()  # 데이터 shape 확인하기  x\_train=x\_train.reshape(x\_train.shape[0], x\_train.shape[1], x\_train.shape[2], 3)  x\_test=x\_test.reshape(x\_test.shape[0], x\_test.shape[1], x\_test.shape[2], 3)  print(x\_train.shape, x\_test.shape)  # 데이터 시각화  plt.figure(figsize=(10, 10))  for i, img in enumerate(x\_train[:8]):  plt.subplot(2,4,i+1)  plt.imshow(x\_train[i])  plt.show()  # y라벨 one hot encoding 수행  y\_train=tf.keras.utils.to\_categorical(y\_train)  y\_test2=tf.keras.utils.to\_categorical(y\_test)  print(y\_train.shape, y\_test2.shape)  # 정규화  print(x\_train.max(), x\_test.max())  x\_train=x\_train/x\_train.max()  x\_test=x\_test/x\_test.max()  print(x\_train.max(), x\_test.max())  # model1 ( kernel+initializer, callback,Dropout)  input\_layer = tf.keras.Input(shape = (32,32,3))  x = tf.keras.layers.Conv2D(32,3,activation = 'relu',padding = 'same',kernel\_initializer= tf.keras.initializers.he\_normal())(input\_layer) # 32 channel kernel\_size =32 conv2d relu  x = tf.keras.layers.MaxPool2D(2,padding = 'same')(x)  x = tf.keras.layers.Dropout(0.2)(x)  x = tf.keras.layers.Conv2D(64,3,activation = 'relu',padding = 'same',kernel\_initializer= tf.keras.initializers.he\_normal())(x)  x = tf.keras.layers.MaxPool2D(2,padding = 'same')(x)  x = tf.keras.layers.Dropout(0.2)(x)  x = tf.keras.layers.Conv2D(128,3,activation = 'relu',padding = 'same',kernel\_initializer= tf.keras.initializers.he\_normal())(x)  x = tf.keras.layers.MaxPool2D(2,padding = 'same')(x)  x = tf.keras.layers.Dropout(0.2)(x)  x = tf.keras.layers.Flatten()(x)  x = tf.keras.layers.Dense(512, activation = 'relu',kernel\_initializer= tf.keras.initializers.he\_normal())(x)  x = tf.keras.layers.Dropout(0.5)(x)  output\_layer = tf.keras.layers.Dense(10,activation = 'softmax',kernel\_initializer= tf.keras.initializers.he\_normal())(x)  model = tf.keras.Model(inputs = [input\_layer], outputs = [output\_layer])  model.summary() |