**멀티캠퍼스 융복합과정 AI**

**이동규**

**Fashion MNIST**

1. **Fashion MNIST with DNN**
2. **Fashion Mnist with CNN**

**Fashion MNIST with DNN**

import pandas as pd

import numpy as np

import tensorflow as tf

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dropout, Dense, Flatten

from tensorflow.keras.optimizers import Adam

import matplotlib.pyplot as plt

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import MinMaxScaler

from sklearn.metrics import classification\_report

# Raw Data Loading

df = pd.read\_csv('/content/drive/My Drive/Multi Cam/Machine Learning/Fasion\_Mnist/fashion-mnist\_test.csv')

df.head(2)

# 결측치와 이상치는 없어요!!

# label 개수 확인

print(df.label.unique())

# shape 확인

print(np.sqrt(df.shape[1]))

# Data Split(Train data와 Test data 분리)

x\_data\_train, x\_data\_test, t\_data\_train, t\_data\_test = \

train\_test\_split(df.drop('label', axis=1, inplace=False),

df['label'],

test\_size=0.3,

random\_state=0)

# Min-Max Normalization

scaler = MinMaxScaler()

scaler.fit(x\_data\_train)

x\_data\_train\_norm = scaler.transform(x\_data\_train)

x\_data\_test\_norm = scaler.transform(x\_data\_test)

## Tensorflow 2.x 구현

model = Sequential()

# 1st hidden layer

model.add(Dense(units= 256,input\_shape = (28,28,1),

activation = 'relu'))

model.add(Flatten())

model.add(Dropout(rate = 0.5))

# 2nd hidden layer

model.add(Dense(units= 20,

activation = 'relu'))

# 3rd hidden layer

model.add(Dense(units= 10,

activation = 'relu'))

# 4th hidden layer

model.add(Dense(units= 10,

activation = 'relu'))

# output layer

model.add(Dense(units=10,

activation = 'softmax'))

print(model.summary())

model.compile(optimizer= Adam(learning\_rate=1e-3),

loss = "sparse\_categorical\_crossentropy",

metrics = ['sparse\_categorical\_accuracy'])

history = model.fit(x\_data\_train\_norm.reshape(-1, 28, 28, 1),

t\_data\_train,

epochs = 20,

batch\_size = 100,

verbose = 1, # 출력이 나오는 것을 눈으로 확인

validation\_split = 0.3)

model.evaluate(x\_data\_test\_norm.reshape(-1,28,28,1),

t\_data\_test)

94/94 [==============================] - 0s 4ms/step - loss: 0.6089 - sparse\_categorical\_accuracy: 0.7977

[0.6088966727256775, 0.7976666688919067]

print(history.history.keys())

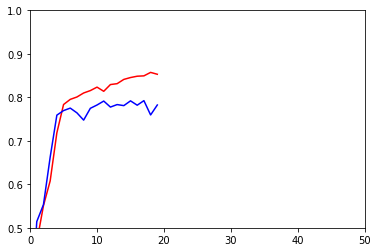
plt.plot(history.history['sparse\_categorical\_accuracy'], color = 'r')

plt.plot(history.history['val\_sparse\_categorical\_accuracy'], color = 'b')

plt.ylim(0.5, 1)

plt.xlim(0,50)

plt.plot



test\_df = pd.read\_csv('/content/drive/My Drive/Multi Cam/Machine Learning/Fasion\_Mnist/fashion-mnist\_test.csv')

test\_df.head(3)

# label 개수 확인

print(test\_df.label.unique())

# shape 확인

print(np.sqrt(test\_df.shape[1]))

x\_test\_data = test\_df.drop('label', axis=1, inplace=False)

t\_test\_data = test\_df['label']

test\_data\_norm = scaler.transform(x\_test\_data)

test\_data\_norm.shape

pred = model.predict(test\_data\_norm.reshape(-1,28,28,1))

y\_pred = [np.argmax(i) for i in pred]

print(classification\_report(t\_test\_data,y\_pred))

precision recall f1-score support

0 0.78 0.86 0.82 1000

1 0.99 0.95 0.97 1000

2 0.62 0.86 0.72 1000

3 0.89 0.90 0.89 1000

4 0.79 0.63 0.70 1000

5 0.89 0.94 0.92 1000

6 0.55 0.42 0.48 1000

7 0.94 0.87 0.90 1000

8 0.97 0.94 0.95 1000

9 0.91 0.95 0.93 1000

accuracy 0.83 10000

macro avg 0.83 0.83 0.83 10000

weighted avg 0.83 0.83 0.83 10000

**Fashion MNIST with CNN**

import pandas as pd

import numpy as np

import tensorflow as tf

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dropout, Dense

# Conv2D: convolution layer만드는

# MaxPooling2D : Max pooling layer 만드는

# Flatten : Flatten layer 만드는

# Dropout : Dropout layer

# Dense : Dense layer

from tensorflow.keras.optimizers import Adam

import matplotlib.pyplot as plt

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import MinMaxScaler

from sklearn.metrics import classification\_report

# Raw Data Loading

df = pd.read\_csv('/content/drive/My Drive/Multi Cam/Machine Learning/Fasion\_Mnist/fashion-mnist\_test.csv')

df.head(2)

# 결측치와 이상치는 없어요!!

# label 개수 확인

print(df.label.unique())

# shape 확인

print(np.sqrt(df.shape[1]))

# Data Split(Train data와 Test data 분리)

x\_data\_train, x\_data\_test, t\_data\_train, t\_data\_test = \

train\_test\_split(df.drop('label', axis=1, inplace=False),

df['label'],

test\_size=0.3,

random\_state=0)

# Min-Max Normalization

scaler = MinMaxScaler()

scaler.fit(x\_data\_train)

x\_data\_train\_norm = scaler.transform(x\_data\_train)

x\_data\_test\_norm = scaler.transform(x\_data\_test)

## Tensorflow 2.x 구현

model = Sequential()

# 첫번째 convolution layer

model.add(Conv2D(filters=32,

kernel\_size=(3,3),

activation='relu',

input\_shape = (28,28,1),

))

# 첫번째 pooling layer

model.add(MaxPooling2D(pool\_size=(2,2)))

# 두번째 convolution layer

model.add(Conv2D(filters=64,

kernel\_size=(3,3),

activation='relu',

))

# 두번째 pooling layer

model.add(MaxPooling2D(pool\_size=(2,2)))

# 세번째 convolution layer

model.add(Conv2D(filters=64,

kernel\_size=(3,3),

activation='relu',

))

# flatten layer

model.add(Flatten())

model.add(Dropout(rate = 0.5))

model.add(Dense(units= 256,

activation = 'relu'))

model.add(Dense(units=10,

activation = 'softmax'))

print(model.summary())

model.compile(optimizer= Adam(learning\_rate=1e-3),

loss = "sparse\_categorical\_crossentropy",

metrics = ['sparse\_categorical\_accuracy'])

history = model.fit(x\_data\_train\_norm.reshape(-1, 28, 28, 1),

t\_data\_train,

epochs = 50,

batch\_size = 700,

verbose = 1,

validation\_split = 0.3

)

model.evaluate(x\_data\_test\_norm.reshape(-1,28,28,1),

t\_data\_test)

94/94 [==============================] - 0s 2ms/step - loss: 0.3797 - sparse\_categorical\_accuracy: 0.8683

[0.379651814699173, 0.8683333396911621]

print(history.history.keys())

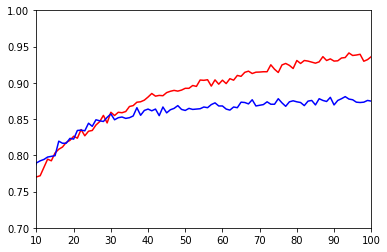
plt.plot(history.history['sparse\_categorical\_accuracy'], color = 'r')

plt.plot(history.history['val\_sparse\_categorical\_accuracy'], color = 'b')

plt.ylim(0.7, 1)

plt.xlim(10,100)

plt.plot



test\_df = pd.read\_csv('/content/drive/My Drive/Multi Cam/Machine Learning/Fasion\_Mnist/fashion-mnist\_test.csv')

test\_df.head(3)

# label 개수 확인

print(test\_df.label.unique())

# shape 확인

print(np.sqrt(test\_df.shape[1]))

x\_test\_data = test\_df.drop('label', axis=1, inplace=False)

t\_test\_data = test\_df['label']

test\_data\_norm = scaler.transform(x\_test\_data)

test\_data\_norm.shape

pred = model.predict(test\_data\_norm.reshape(-1,28,28,1))

y\_pred = [np.argmax(i) for i in pred]

print(classification\_report(t\_test\_data,y\_pred))

precision recall f1-score support

0 0.77 0.93 0.84 1000

1 0.99 0.98 0.98 1000

2 0.78 0.89 0.83 1000

3 0.93 0.91 0.92 1000

4 0.85 0.79 0.82 1000

5 0.99 0.96 0.97 1000

6 0.78 0.58 0.66 1000

7 0.95 0.94 0.94 1000

8 0.97 0.98 0.98 1000

9 0.94 0.98 0.96 1000

accuracy 0.89 10000

macro avg 0.89 0.89 0.89 10000

weighted avg 0.89 0.89 0.89 10000