TF2.0 시작하기

- 손글씨 데이터 셋(MNIST)을 이용한 신경망 만들기

사전 작업

- tf2.0 설치 후, 재시작(설치 적용을 위해)
- 런타임 - 런타임 유형 변경 - GPU 설정
- google colab의 경우 기본 2.x버전 선택됨.

학습내용

- tensorflow2.x를 이용한 신경망을 구현해본다.
- Dropout에 대해 이해하고 실습해본다.

In [3]:

```
import tensorflow as tf
```

In [4]:

```
print(tf.__version__)
```

```
2.4.0
```

만약 설치시 아래와 같은 명령으로 설치 가능

```
!pip install -q tensorflow-gpu==2.0.0-rc1
import tensorflow as tf
```

MNIST 데이터 셋을 이용한 신경망 구성

In [5]:

```
# 데이터 가져오기
mnist = tf.keras.datasets.mnist
(x_train, y_train), (x_test, y_test) = mnist.load_data()

# 데이터 정규화
x_train, x_test = x_train / 255.0, x_test / 255.0
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mn
ist.npz (https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz)
11493376/11490434 [==============================] - 0s 0us/step
```

In [6]:

```
print("학습용 데이터 : x: {}, y:{}".format(x_train.shape, y_train.shape) )
print("테스트 데이터 : x: {}, y:{}".format(x_test.shape, y_test.shape) )
```

```
학습용 데이터 : x: (60000, 28, 28), y:(60000,)
테스트 데이터 : x: (10000, 28, 28), y:(10000,)
```

신경망 구성

- tf.keras.Sequential를 이용한 모델 구성

In [7]:

```
model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(input_shape=(28, 28)),    # 2D -> 1D
  tf.keras.layers.Dense(128, activation='relu'),    # 활성화 함수 - relu
  tf.keras.layers.Dropout(0.2),                     # Dropout적용
  tf.keras.layers.Dense(10, activation='softmax')   # 활성화 함수 - softmax
])
```

구성

- sparse_categorical_crossentropy : 다중 분류 손실함수 (정수값 기준)
- categorical_crossentropy : 다중 분류 손실함수 (one-hot-encoding 기준 (예측과 실제 결과값))

In [8]:

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

Out[8]:

```
<tensorflow.python.keras.engine.sequential.Sequential at 0x7fb25a98a9b0>
```

모델 훈련 및 평가

In [9]:

```
%%time

model.fit(x_train, y_train, epochs=15)
model.evaluate(x_test,  y_test, verbose=2)
```

```
Epoch 1/15
1875/1875 [==============================] - 5s 2ms/step - loss: 0.4789 - accuracy:
0.8591
Epoch 2/15
1875/1875 [==============================] - 3s 2ms/step - loss: 0.1522 - accuracy:
0.9545
Epoch 3/15
1875/1875 [==============================] - 3s 2ms/step - loss: 0.1068 - accuracy:
0.9674
Epoch 4/15
1875/1875 [==============================] - 3s 2ms/step - loss: 0.0848 - accuracy:
0.9735
Epoch 5/15
1875/1875 [==============================] - 3s 2ms/step - loss: 0.0740 - accuracy:
0.9768
Epoch 6/15
1875/1875 [==============================] - 3s 2ms/step - loss: 0.0631 - accuracy:
0.9795
Epoch 7/15
1875/1875 [==============================] - 3s 2ms/step - loss: 0.0569 - accuracy:
0.9810
Epoch 8/15
1875/1875 [==============================] - 3s 2ms/step - loss: 0.0515 - accuracy:
0.9833
Epoch 9/15
1875/1875 [==============================] - 3s 2ms/step - loss: 0.0458 - accuracy:
0.9847
Epoch 10/15
1875/1875 [==============================] - 3s 2ms/step - loss: 0.0452 - accuracy:
0.9854
Epoch 11/15
1875/1875 [==============================] - 3s 2ms/step - loss: 0.0387 - accuracy:
0.9882
Epoch 12/15
1875/1875 [==============================] - 3s 2ms/step - loss: 0.0378 - accuracy:
0.9875
Epoch 13/15
1875/1875 [==============================] - 3s 2ms/step - loss: 0.0345 - accuracy:
0.9881
Epoch 14/15
1875/1875 [==============================] - 3s 2ms/step - loss: 0.0327 - accuracy:
0.9890
Epoch 15/15
1875/1875 [==============================] - 3s 2ms/step - loss: 0.0316 - accuracy:
0.9892
313/313 - 0s - loss: 0.0689 - accuracy: 0.9800
CPU times: user 52.4 s, sys: 5.36 s, total: 57.8 s
Wall time: 49.5 s
```

REF

- TF2.0 Tutorial : https://www.tensorflow.org/tutorials/quickstart/beginner (https://www.tensorflow.org/tutorials/quickstart/beginner)
- tf.keras.Sequential : https://www.tensorflow.org/api_docs/python/tf/keras/Sequential (https://www.tensorflow.org/api_docs/python/tf/keras/Sequential)

History

- 2020/12/28 tf 2.x (ver 1.1)