TF2.0 신경망 만들기

- 손글씨 데이터 셋을 이용한 신경망 만들기

사전 작업

- tf2.0 설치 후, 재시작(설치 적용을 위해)
- 런타임 - 런타임 유형 변경 - GPU 설정

In [0]:

```
import tensorflow as tf
```

In [2]:

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

```
2.0.0-rc1
```

In [0]:

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

In [4]:

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

```
2.0.0-rc1
```

MNIST 데이터 셋을 이용한 신경망 구성

In [5]:

```
mnist = tf.keras.datasets.mnist

(x_train, y_train), (x_test, y_test) = mnist.load_data()
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 [8]:

```
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 [0]:

```
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 [0]:

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

모델 훈련 및 평가

In [13]:

```
model.fit(x_train, y_train, epochs=5)
model.evaluate(x_test,  y_test, verbose=2)
```

```
Train on 60000 samples
Epoch 1/5
60000/60000 [==============================] - 6s 106us/sample - loss: 0.2982 - accu
racy: 0.9138
Epoch 2/5
60000/60000 [==============================] - 5s 75us/sample - loss: 0.1457 - accur
acy: 0.9574
Epoch 3/5
60000/60000 [==============================] - 4s 72us/sample - loss: 0.1076 - accur
acy: 0.9672
Epoch 4/5
60000/60000 [==============================] - 4s 73us/sample - loss: 0.0880 - accur
acy: 0.9722
Epoch 5/5
60000/60000 [==============================] - 5s 75us/sample - loss: 0.0747 - accur
acy: 0.9760
10000/1 - 1s - loss: 0.0380 - accuracy: 0.9753
```

Out[13]:

```
[0.07429377017640051, 0.9753]
```

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)

In [0]:

```
1
```