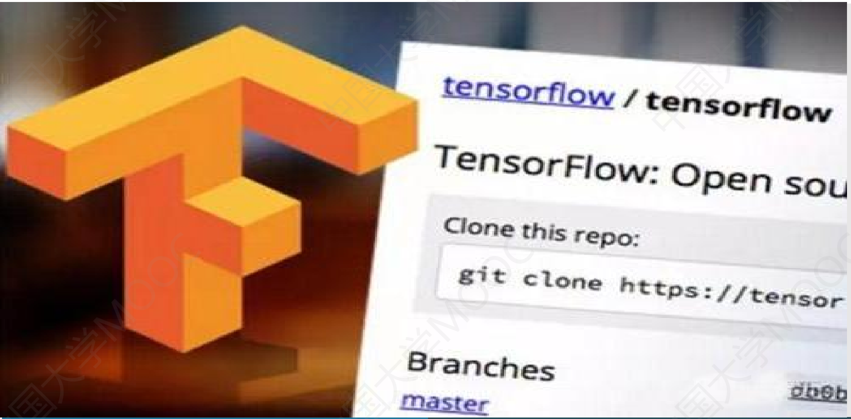

14.5 实例：卷积神经网络实现手写数字识别

中国大学MOOC

导入库

```
In  [1]: import tensorflow as tf
         tf.__version__, tf.keras.__version__

Out[1]: ('2.0.0', '2.2.4-tf')

In  [2]: import numpy as np
         import matplotlib.pyplot as plt

In  [3]: gpus = tf.config.experimental.list_physical_devices('GPU')
         tf.config.experimental.set_memory_growth(gpus[0], True)
```


中国大学MOOC

加载数据集

In [4]:
```
mnist=tf.keras.datasets.mnist
(train_x, train_y),(test_x,test_y) = mnist.load_data()
```

In [5]:
```
print(train_x.shape)
print(train_y.shape)
print(test_x.shape)
print(test_y.shape)
```

```
(60000, 28, 28)
(60000,)
(10000, 28, 28)
(10000,)
```

In [6]:
```
type(train_x), type(train_y)
```

Out[6]:
```
(numpy.ndarray, numpy.ndarray)
```

In [7]:
```
type(test_x), type(test_y)
```

Out[7]:
```
(numpy.ndarray, numpy.ndarray)
```

神经网络&深度学习 TENSORFLOW Google

数据预处理

```
In  [7]: type(test_x), type(test_y)

Out[7]: (numpy.ndarray, numpy.ndarray)

In  [8]: X_train, X_test = tf.cast(train_x, dtype=tf.float32)/255.0, tf.cast(test_x, dtype=tf.float32)/255.0
         y_train, y_test = tf.cast(train_y, dtype=tf.int32), tf.cast(test_y, dtype=tf.int32)

In  [9]: X_train = train_x.reshape(60000, 28,28,1)
         X_test = test_x.reshape(10000, 28,28,1)
```

维度变换

```
In  [9]: X_train = tf.expand_dims(train_x,3)
         X_test = tf.expand_dims(test_x,3)
```

```
In  [10]: print(X_train.shape)
          print(X_test.shape)

(60000, 28, 28, 1)
(10000, 28, 28, 1)
```


西安科技大学 计算机科学与技术学院

建立模型

In [11]:
```
model=tf.keras.Sequential([

    # unit 1
    tf.keras.layers.Conv2D(16, kernel_size=(3, 3), padding="same", activation=tf.nn.relu, input_shape=(28,28,1)),
    tf.keras.layers.MaxPool2D(pool_size=(2, 2)),

    # unit 2
    tf.keras.layers.Conv2D(32, kernel_size=(3, 3), padding="same", activation=tf.nn.relu),
    tf.keras.layers.MaxPool2D(pool_size=(2, 2)),

    # unit 3
    tf.keras.layers.Flatten(),

    # unit 4
    tf.keras.layers.Dense(128, activation="relu"),
    tf.keras.layers.Dense(10, activation="softmax")

])
```


中国大学MOOC

神经网络&深度学习 Google TENSORFLOW

查看摘要

```
In  [12]: model.summary()

Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 28, 28, 16)        160
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 14, 14, 16)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 14, 14, 32)        4640
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 7, 7, 32)          0
_________________________________________________________________
flatten (Flatten)            (None, 1568)              0
_________________________________________________________________
dense (Dense)                (None, 128)               200832
_________________________________________________________________
dense_1 (Dense)              (None, 10)                1290
=================================================================
Total params: 206,922
Trainable params: 206,922
Non-trainable params: 0
```

西安科技大学
计算机科学与技术学院

配置训练方法

```
In  [13]:  model.compile(optimizer='adam',
                         loss='sparse_categorical_crossentropy',
                         metrics=['sparse_categorical_accuracy'])
```


中国大学MOOC

训练模型

```
In  [14]: model.fit(X_train, y_train, batch_size=64, epochs=5, validation_split=0.2)

Train on 48000 samples, validate on 12000 samples
Epoch 1/5
48000/48000 [==============================] - 5s 100us/sample - loss: 0.4406 - sparse_categorical_accuracy: 0.9316 - val_loss: 0.0844 - val
_sparse_categorical_accuracy: 0.9742
Epoch 2/5
48000/48000 [==============================] - 3s 55us/sample - loss: 0.0639 - sparse_categorical_accuracy: 0.9803 - val_loss: 0.0911 - val_
sparse_categorical_accuracy: 0.9733
Epoch 3/5
48000/48000 [==============================] - 2s 51us/sample - loss: 0.0416 - sparse_categorical_accuracy: 0.9868 - val_loss: 0.0998 - val_
sparse_categorical_accuracy: 0.9718
Epoch 4/5
48000/48000 [==============================] - 2s 50us/sample - loss: 0.0356 - sparse_categorical_accuracy: 0.9884 - val_loss: 0.0671 - val_
sparse_categorical_accuracy: 0.9822
Epoch 5/5
48000/48000 [==============================] - 3s 53us/sample - loss: 0.0266 - sparse_categorical_accuracy: 0.9915 - val_loss: 0.1178 - val_
sparse_categorical_accuracy: 0.9727

Out[14]: <tensorflow.python.keras.callbacks.History at 0x2814c2eac88>
```


中国大学MOOC

评估模型

```
In  [15]: model.evaluate(X_test, y_test, verbose=2)

          10000/1 - 1s - loss: 0.0390 - sparse_categorical_accuracy: 0.9802

Out[15]: [0.07610834636164945, 0.9802]
```


中国大学MOOC