Tensorflow-模型的保存

导入包

```
import os
import tensorflow
```

下载数据

```
(train_images, train_labels), (test_images, test_labels)
= tensorflow.keras.datasets.mnist.load_data()
```

处理数据

取出1000个数据

```
train_labels = train_labels[:1000]
test_labels = test_labels[:1000]
```

像数值缩小

```
train_images = train_images[:1000].reshape(-1, 28 * 28)
/ 255.0
test_images = test_images[:1000].reshape(-1, 28 * 28) /
255.0
```

一个简单的序列模型

```
def create_model():
    model = tensorflow.keras.Sequential()
    model.add(
        tensorflow.keras.layers.Dense(
            units=512, activation='relu', input_shape=
(784, )
        ),
    ),
    model.add(
        tensorflow.keras.layers.Dropout(0.2),
    )
```

```
    model.add(
        tensorflow.keras.layers.Dense(
            units=10, activation='softmax', ),
    )
    model.compile(
        optimizer='adam',
        loss='sparse_categorical_crossentropy',
        metrics=['accuracy'],
    )
    return model
```

实例话图像处理对象

```
model = create_model()
```

显示模型结构

```
model.summary()
```

获取路径

> *模型数据的保存地址*

```
checkpoint_path = "training_1/cp.ckpt"
# 返回路径 ： dirname
checkpoint_dir = os.path.dirname(checkpoint_path)
```

创建回调

> *保存模型权重*

```
cp_callback =
tensorflow.keras.callbacks.ModelCheckpoint(
    filepath=checkpoint_path,
    save_weights_only=True,
    verbose=1,
)
```

训练模型

```
model.fit(train_images,
          train_labels,
          epochs=10,
          validation_data=(test_images, test_labels),
          callbacks=[cp_callback], # 通过回调训练
          )
```

评估模型

```
loss, acc = model.evaluate(test_images,  test_labels,
verbose=2)

print("Untrained model, accuracy:
{:5.2f}%".format(100*acc))
```