

**课 程 实 验 报 告**

**课程名称： 物联网中间件**

**专业班级： 物联网工程1601**

**学 号： U2016148989**

**姓 名： 潘翔**

**指导教师： 顾琳**

**报告日期： 2019.4.8**

**计算机科学与技术学院**

目 录

[1 基于 TensorFlow平台的手写数字识别系统 3](#_Toc212780910)

[1.1 实验环境 3](#_Toc1142807388)

[1.2 实验内容与要求 3](#_Toc1865126060)

[1.3 实验过程与结果 3](#_Toc805358075)

[1.4 核心源码说明 7](#_Toc1617264233)

[1.5 思考题 8](#_Toc1590295633)

[1.6 实验体会与总结 8](#_Toc879893916)

[参考文献 9](#_Toc552606407)

# 基于 TensorFlow平台的手写数字识别系统

## 实验环境

### 系统环境

OS: Manjaro 18.0.4 Illyria

Kernel: x86\_64 Linux 5.0.7-1-MANJARO

CPU: Intel Core i7-6700HQ @ 8x 3.5GHz

GPU: GeForce GTX 965M

RAM: 7865MiB

### 平台环境

CUDA: V10.1.105

Tensorflow: 1.13.1

## 实验内容与要求

1. 熟悉TensorFlow和Keras平台和提供的库。
2. 利用平台和提供的库，实现以下的功能：
   1. 用Q-learning 的方法实现一个小例子在世界寻找宝藏。
   2. 在MNIST数据集上训练一个简单的深度神经网络，改变神经元的个数和迭代的次数，考察训练的神经网络的准确度的变化。
   3. 保存应用训练好的的深度神经网络，识别新的手写数字。

## 实验过程与结果

### Q-learning Test



图1.1 Q-Learning Test Run

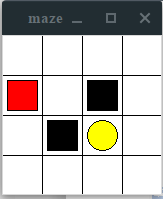


图1.2 Q-Learning Test运行过程

### MNIST Test

1. 模型训练

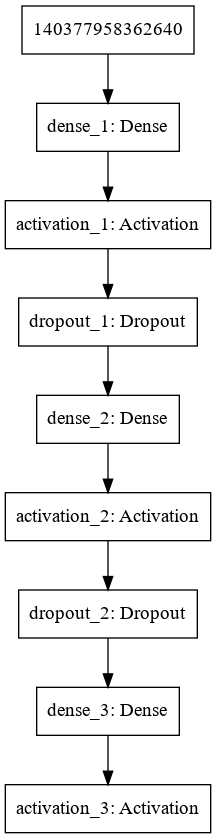


图1.3 CNN网络可视化

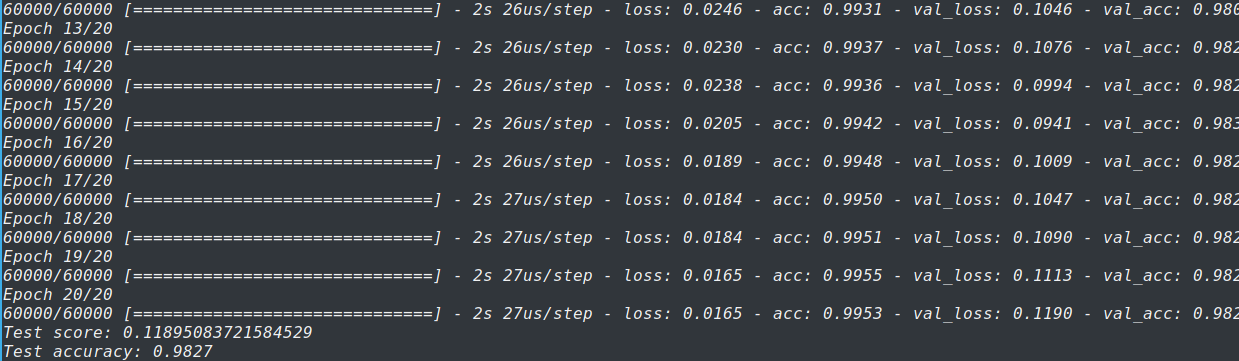


图1.4 MNIST Test运行过程

可以看到在19/20左右Loss不再下降，说明模型已经接近收敛，增大epoch不会有较大提升



图1.5 MNIST Test Tensorboard训练过程可视化

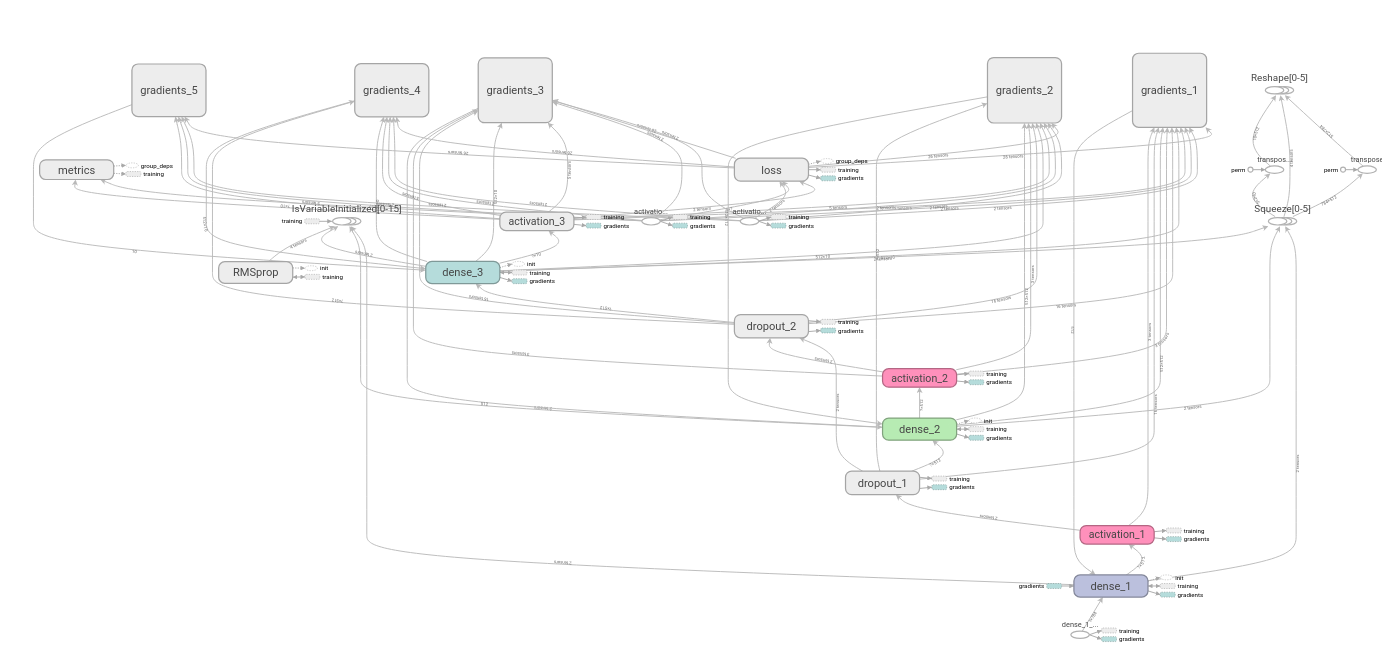


图1.6 MNIST Test Tensorboard计算图可视化

1. 保存模型

**from keras.models import** load\_model

model.save('mnish.h5')

1. 模型测试
2. 模型更改

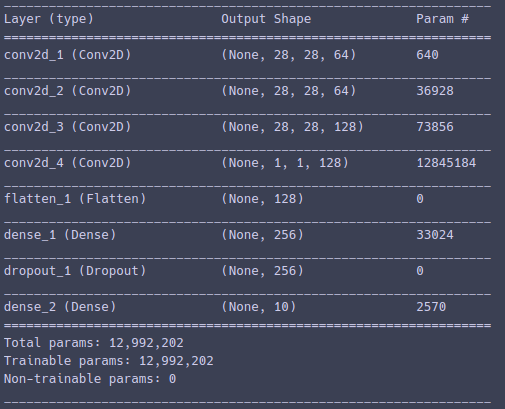


图1.7 Conv2D网络Summary

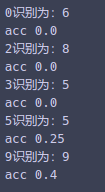


图1.8 Conv2D网络测试结果

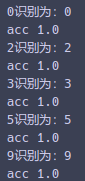


图1.9 二值化后取反测试结果

## 核心源码说明

### Keras-CNN

model = Sequential()callbacks = [keras.callbacks.ModelCheckpoint('minist.h5', monitor='val\_acc', verbose=1, save\_best\_only=**True**,

mode='auto')]model.add(Conv2D(64, kernel\_size=(3, 3),

activation='relu',padding='same',

input\_shape=(28,28,1)))model.add(Conv2D(64, (3, 3),padding='same', activation='relu'))model.add(Conv2D(128, (3, 3),padding='same', activation='relu'))model.add(Conv2D(128, (28, 28),activation='relu'))model.add(Flatten())model.add(Dense(256, activation='relu'))model.add(Dropout(0.5))model.add(Dense(10, activation='softmax'))model.summary()sgd = SGD(lr=0.01, momentum=0.9)model.compile(loss='categorical\_crossentropy', optimizer=sgd, metrics=['accuracy'])model.fit(x\_train, y\_train,

batch\_size=128,

epochs=50,

verbose=1,

validation\_data=(x\_test, y\_test),callbacks=callbacks)

## 实验体会与总结

实验基于Keras实现MINIST,因为对于Keras较为熟悉，实验较为简单，其中有一些细节：

1. 对于输入图片需要注意二值化，同时需要注意，原始训练集为黑底白字，此时可以对于图片中的像素进行统计，判断为白底黑字还是黑底白字，对其适应二值化
2. 图片边缘存在边框可能会有一定影响，需注意裁剪
3. 使用MLP和Conv2D在其上表现类似

# 参考文献