# Tutorial on How to Build Simple Image Classification Network

Author ChenglinLu

目录

[Tutorial on How to Build Simple Image Classification Network 1](#_Toc525040700)

[1. 配置TensorFlow 3](#_Toc525040701)

[1.1安装方法 3](#_Toc525040702)

[1.2安装过程遇到的坑 3](#_Toc525040703)

[1.3安装常用CV库（Python） 4](#_Toc525040704)

[2.准备数据集 4](#_Toc525040705)

[2.1下载CIFAR-10 (163M): 4](#_Toc525040706)

[2.2关于CIFAR-10 5](#_Toc525040707)

[3.图像分类任务 5](#_Toc525040708)

[3.1数据预处理 5](#_Toc525040709)

[3.2构建网络 5](#_Toc525040710)

[3.3训练 6](#_Toc525040711)

[3.4推理 7](#_Toc525040712)

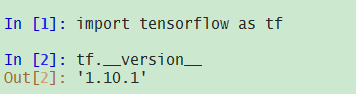
# 配置TensorFlow

## 1.1安装方法

建议根据官网安装指南进行安装：

<https://www.tensorflow.org/install/>

安装后若显示以下信息，则表明安装成功：



## 1.2安装过程遇到的坑

* 官方总结：

<https://www.tensorflow.org/install/install_linux#common_installation_problems>

* 在VMware10.0, Ubuntu18.04环境下，建议安装TensorFlow1.5版本，高于此版本运行时会报错，双系统下无此问题；
* 安装时提示Read time out，则表明网速太慢或网络中断；
* 安装GPU版本的TensorFlow，要求CUDA计算能力3.5以上，详见GPUs Supported:

<https://en.wikipedia.org/wiki/CUDA#GPUs_supported>

* 安装GPU版本时，特别注意CUDA和cuDNN版本对应问题；可用下面命令检查CUDA和cuDNN版本：

cat /usr/local/cuda/version.txt

cat /usr/local/cuda/include/cudnn.h | grep CUDNN\_MAJOR -A 2

CUDA cuDNN官方安装教程：

* 官网推荐的CUDA和Tensorflow版本:



Figure1. Tensorflow, CUDA and cuDNN configurations

Windows, MAC详见：

<https://www.tensorflow.org/install/install_sources#tested_source_configurations>

* ……

## 1.3安装常用CV库（Python）

详见requirements.txt

Install:

pip install –r requirements.txt

# 2.准备数据集

## 2.1下载CIFAR-10 (163M):

<https://www.cs.toronto.edu/~kriz/cifar.html>

## 2.2关于CIFAR-10

CIFAR-10 是由Hinton 的两个大弟子 Alex Krizhevsky（提出了AlexNet），Ilya Sutskever 收集的一个用于普适物体识别的数据集；数据集包含60000张 32x32 彩色图像，共有10类, 每个类别共计6000张图像. 数据集包含50000个训练集和10000个测试集。

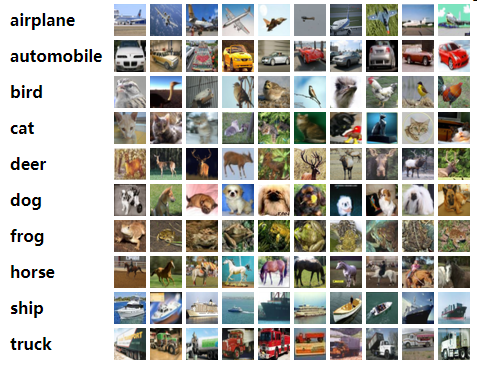


Figure2. CIFAR-10 Dataset

# 3.图像分类任务

注：由于公司电脑没有可用的NVIDIA显卡，以下测试均在Ubuntu14.04， TensorFlow-CPU下进行。

## 3.1数据预处理

主要包括数据集的载入，归一化处理，one-hot编码转换等，详见data.py代码部分。

## 3.2构建网络

为了便于理解，本教程采用AlexNet，其网络结构如下：

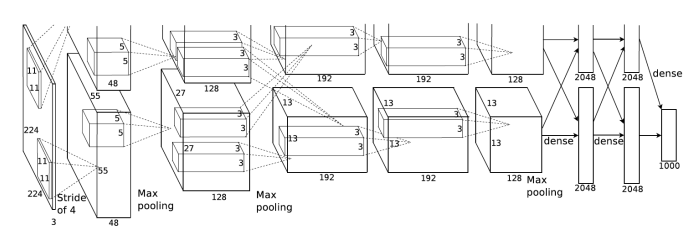


Figure3. AlexNet architecture

详细介绍可以参考论文《Imagenet Classification with Deep Convolutional Neural Networks》

<https://mohitjainweb.files.wordpress.com/2018/06/imagenet-classification-with-deep-convolutional-neural-networks.pdf>

具体实现见 alexnet.py

## 3.3训练

训练中用到的优化方法为RMSPropOptimizer，详见：

<http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf>

训练中的Hyper parameters:

learning\_rate=1e-3,

\_BATCH\_SIZE = 128, # 32,64,128,256, and so on

\_ITERATION = 2000, # here, equal to epochs

关于Hyper parameters可参考：

<https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a-deep-neural-network-d0604917584a>

训练过程详见train.py

结果如下：

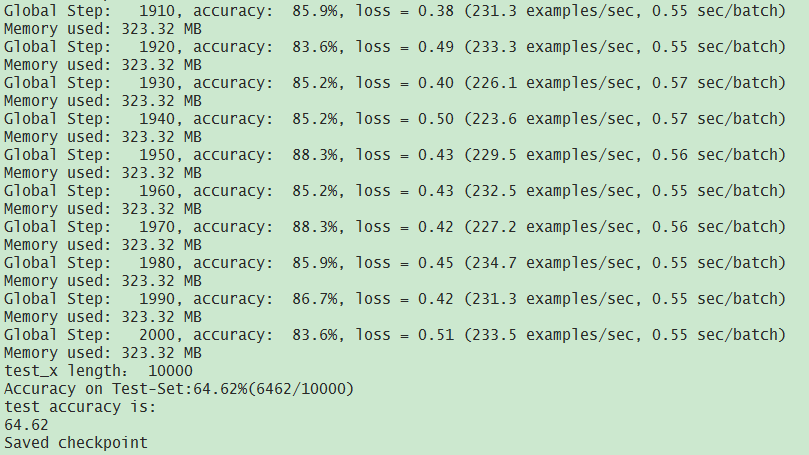


Figure4. training result

## 3.4推理

根据最后一次training保存的模型参数文件：

* \*. data-00000-of-00001
* \*.meta
* \*.index

进行inference，并对其性能进行评估，包括耗时，内存占用等，详见inference.py。

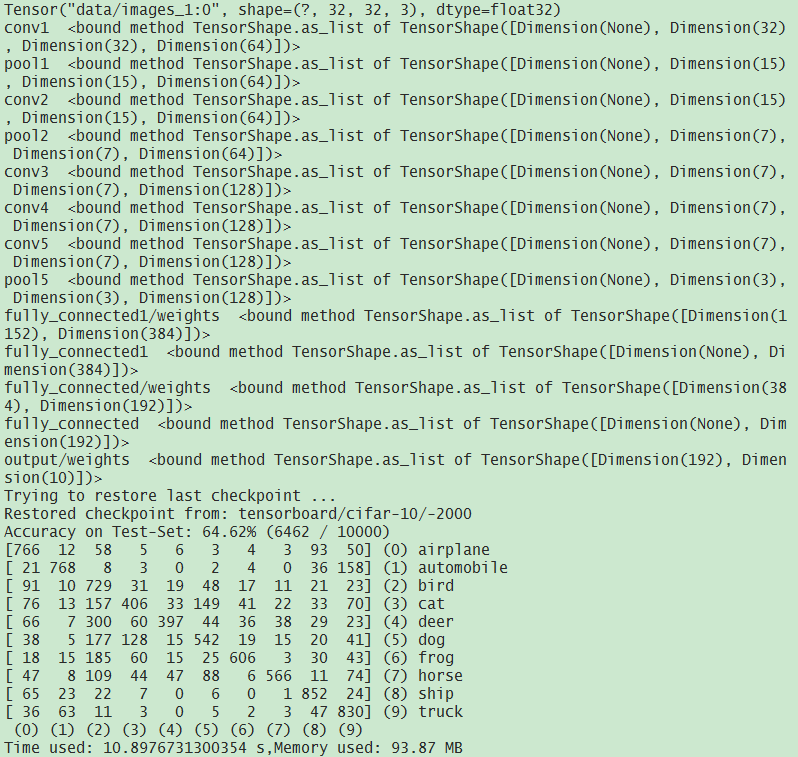


Figure5. Inference result

Ref:

[1] (How to Build a Simple Image Recognition System with TensorFlow)

<https://www.wolfib.com/Image-Recognition-Intro-Part-1/>

[2] (PAPER EXPLANATION: IMAGENET CLASSIFICATION WITH DEEP CONVOLUTIONAL NEURAL NETWORKS (ALEXNET))

<https://mohitjain.me/2018/06/06/alexnet/>

[3](CIFAR-10 Image Classification inTensorFlow)

<https://towardsdatascience.com/cifar-10-image-classification-in-tensorflow-5b501f7dc77c>

[4] (Python 性能分析入门指南) <https://segmentfault.com/a/1190000000616798>

[5] (TensorFlow Python-API) <https://www.tensorflow.org/api_docs/python/tf>

[6] (CNN经典模型整理Lenet, Alexnet, GoogLeNet, VGG,Deep Residual Learning, SqueezeNet)

<https://blog.csdn.net/m0_37264397/article/details/75174484>

[7] (TensorFlow卷积网络教程)<https://blog.csdn.net/hustqb/article/details/80228627>

[8] (What is the class of this image?)

<http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html>