

# 2018 TAIL CAMP AI实战训练营



## 图像项目

学习材料指引

2018-01-20

### 总体目标

掌握深度学习的基本概念与大数据处理的技巧，熟悉深度学习框架的使用，基于卷积神经网络（CNN）搭建和运行网络模型，解决教学场景中常见的图像分类问题。

### 涉及内容

- Caffe等深度学习框架
- CNN相关基础理论
- 手写文字识别

- 人脸表情识别
- 数据处理与格式转换

P.S. 对于深度学习不太熟悉的小伙伴们，在本文档里我们主要以Caffe为例进行了一系列基础学习资料的推荐。有一定的DL经验或者有自己的熟悉使用的框架的同学可以根据自身情况跳过开篇的学习指引部分直奔任务环节！

## Week 1：手写文字识别

### 总体目标

- 熟悉并搭建深度学习框架
- 利用HWDB数据集，实现基于CNN的手写文字识别

### 学习指引

#### Day 1：Caffe搭建和简单测试

Caffe是一个可读性高、快速清晰的深度学习框架。对DL还不太熟悉的同学们可以先从了解Caffe来打开新世界的大门！

##### 1. Caffe官网

<http://caffe.berkeleyvision.org>

##### 2. 安装教程

<http://caffe.berkeleyvision.org/installation.html>

##### 3. Caffe官方教程中译本

<http://pan.baidu.com/s/1miwlUta>

##### 4. Caffe自带MNIST手写数字识别上手小测试

<http://blog.csdn.net/yingyujianmo/article/details/44984485>

<http://blog.csdn.net/lkj345/article/details/51280369>

P.S. 大部分的Caffe安装教程针对的是全新环境安装，包括较为繁琐的显卡环境配置，同学们可以先检查系统是否已配置好CUDA环境，正常的话跳过依赖项直接从github上下载Caffe进行编译就可以了~

#### Day 2：CNN的概念和基础学习

CNN也就是深度学习的核心——卷积神经网络，在接下来的这一段时间里我们都会围绕着CNN来展开学习与实践，如果你对CNN是什么和TA的工作原理还不太了解的话，赶紧阅读下面的资料吧！

1. CNN斯坦福教程（入门必备！CNN的讲解在Lecture 5，CNN训练方式在Lecture 6&7，其他的课程大家可以根据需要来学习~）

<http://cs231n.stanford.edu/syllabus.html>

中文版看这里

<http://www.52ml.net/17723.html>

网易云课堂视频

<http://study.163.com/course/introduction.htm?courseId=1003223001#/courseDetail>

2. 关于CNN学习的博客

<http://blog.csdn.net/kanghe2000/article/details/70940491>

[http://blog.csdn.net/v\\_july\\_v/article/details/51812459](http://blog.csdn.net/v_july_v/article/details/51812459)

3. CNN自带实例的实现（包含Day 1的MNIST，不过今天需要更深入地了解训练的过程）

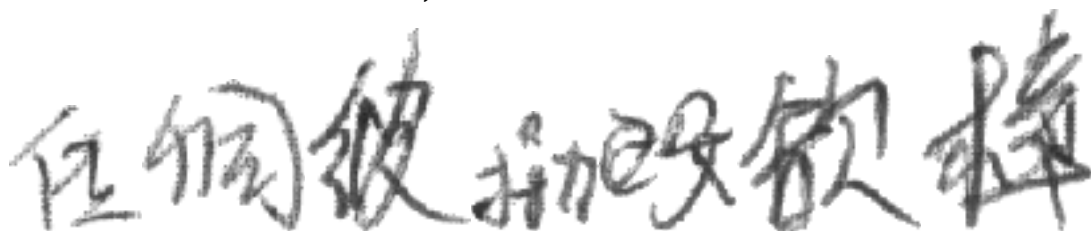
<https://www.cnblogs.com/denny402/p/5075490.html>

4. CNN网络可视化页面（复制train\_val.prototxt，deploy.txt里的内容到网页里，来更直观地了解CNN网络结构吧！）

<http://ethereon.github.io/netscope/#/editor>

### Day 3：实战！手写文字的识别之数据篇！

对手写文字的识别是教学场景乃至更多涉及到书写动作的场景里经常遇到和急需解决的问题，手写文字图片一般长成下面这样：



大家都能看出来上面的都是什么字吗？不能？但你可以让机器在你的训练下变得可以！磨刀不误砍柴工，今天的主要任务是要对数据进行预处理和封装。

1. 数据分析

我们提供给大家的数据库里包含着训练集train和测试集test，其中train文件夹里包含1877个文件夹，代表1877个汉字，每个文件夹的命名其实就是每个字的label，而test文件夹里则是打乱顺序的测试集。认真了解数据集结构，为后续工作打好基础！

2. 数据预处理

大部分的CNN网络要求统一的图片尺寸，但是我们的数据库里每一张图片的尺寸都不尽相同，请首先按照你对网络设计的预估来resize图片吧！（懒人推荐：96x96）

### 3. 数据集划分和封装

请注意！由于没有提供label，所以test文件夹仅用来做最后的验证，在训练过程中是无法使用的。因此，需要大家自己来从train数据中重新划分training set和validation set。另一方面，Caffe支持多种数据读入方式，LMDB是其中较为常用和快捷的一种，参考以下文档划分好自己的训练集和验证集，并打包数据~在进行数据集划分和封装的时候，一定要注意train和val的比例，label的生成（0-1876），以及图像均值的计算。

<http://blog.csdn.net/xiaoyang19910623/article/details/53420433>

<https://www.cnblogs.com/maohai/p/6453424.html>

P.S. 感谢中国科学院自动化研究所（CASIA）和国家模式识别实验室（NLPR）授权本次活动使用HWDBv1.1手写文字数据库，此次训练营活动中的数据任何学员不得私自下载和传播，如有感兴趣的小伙伴需要该数据集进行后续的学术研究，请参考以下地址向CASIA和NLPR提出免费使用的书面申请。

[http://www.nlpr.ia.ac.cn/databases/handwriting/Application\\_form.html](http://www.nlpr.ia.ac.cn/databases/handwriting/Application_form.html)

## Day 4&5：实战！手写文字的识别之训练篇！

成功完成对数据集的处理后，接下来就是最重要的网络设计以及训练环节。文字识别本质上是一个分类问题，那么首先我们需要了解分类问题中几种经典的网络结构，然后再根据数据集的规模和目标问题的复杂度来选择其中的一种或者自行构建一个网络进行训练。

1. 了解经典CNN分类网络模型框架（每一种网络在网上都有大量的中文介绍，可自行搜寻）

LeNet: <http://yann.lecun.com/exdb/lenet/>

AlexNet: <https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>

VGGNet: [http://www.robots.ox.ac.uk/~vgg/research/very\\_deep/](http://www.robots.ox.ac.uk/~vgg/research/very_deep/)

GoogleNet: [http://www.cv-foundation.org/openaccess/content\\_cvpr\\_2015/papers/Szegedy\\_Going\\_Deeper\\_With\\_2015\\_CVPR\\_paper.pdf](http://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Szegedy_Going_Deeper_With_2015_CVPR_paper.pdf)

ResNet: <https://arxiv.org/pdf/1512.03385v1.pdf>

2. 选择网络或者根据自己的理解搭建网络。网络结构越复杂，所需的训练时间越长，而且更容易导致过拟合，注意合理设计网络的宽度和深度。
3. 基于确定好的数据集和网络进行训练。训练过程中关注training loss和validation loss的变化趋势，对网络结构进行适当的调整以获取最佳训练效果。

参考: <http://blog.csdn.net/thystar/article/details/51619771>

### Day 6: 实战! 手写文字的识别之验证篇!

当手中的网络模型已经被训练到最佳状态的时候(结合train loss和validation loss来考虑), 最后一步就是对我们的test集进行验证。在这里, 只需要把test的每一张图片输入训练好的网络, 输出预测的label值, 并将图片文件名和label写入指定格式的json文件(参考我们提供的submission\_sample.json文件), 上传提交到平台上等待结果就可以啦!

其实, 相对数字化的结果输出, 更重要的是整个学习和研究的过程。因此, 除了最终的submission, 也请同学们静下心来总结第一周的心得体会: 从数据处理到网络设计, 从模型训练到超参数调整, 可以是train和val的loss曲线, 也可以是你在训练中遇到的给你带来最大困扰的bug, 把所有的亮点和难点都记录成一份简单的技术文档吧。

最后需要提交的文件包括:

- code (必选)
- 技术文档 (必选)
- submission\_sample.json