

# Python与深度学习基础

**致谢：**通过将近一学期的学习，使得本小组成员对深度学习特别是卷积神经网络的训练有了一些粗浅的认识，在此特别感谢各位授课的年青教师的仔细讲解于两位助教的认真、负责的工作。小组成员谨认为该课程开辟了我校公选课的新维度，为我校提升本科生教育质量做出了不可忽视的贡献。感谢一位15级学姐的热心帮助。感谢CSDN博客，感谢百度百科，谷歌。

## 1 任务说明

小组成员在前期探索本次大作业的方向的时候，首先明确了该作业的核心就是要训练出一个深度学习的模型，体会训练模型的过程。在小组组建之初的前瞻性讨论中，提出了“智能仿写白落梅文章”、“生成指定人的手写体文字”、“生成动作模型”等课题。

然而，鉴于小组成员均为初次接触到深度学习这个概念，对于相关框架流程未能全面掌握。经过考察相关资料和与高年级学姐学长的全面、深入地交流探讨，小组最后决定选定课题“基于深度学习的PC端防疲劳程序”，该题目来源于生活实际，在日常使用电脑等电子设备的时候，人们会不知不觉地减少眨眼的次数，时间长了就会造成视觉疲劳，长此以往对视力会造成较为严重的伤害。于是自然想到能否在人们“忘记眨眼”的时候通过程序进行及时提醒呢？

我们的程序大体思路就是通过调用笔记本摄像头取帧并通过Opencv库函数自动框取脸部图像再进行判断是否处于睁眼状态，然后设定时间决定是否报警提示。

该课题的难点在于如让计算机知道人是不是“疲劳”了，也就是说要能够判断屏幕前的用户是不是在眨眼，由此我们想到了在本课程的手讲授中作为例子出现写体数字二分类识别模型，希望借此构建我们的深度学习框架，训练出核心模型。

## 2 数据处理

既然是深度学习就得有相当量级的数据去处理，同样要面对的是数据分类的问题。我们

采取了使用Opencv库函数调用摄像头对不同的志愿者、在不同背景下进行等间隔取帧来获得原始数据库。

对于分类问题，我们主要先后采取了三种方法。第一是原始手动分类，在数据库里面分别找睁眼或闭眼的图片然后将其复制到相应目录下，该方法效率相对极低，只在前期采用；第二就是在采集原始数据的时候使用技巧，志愿者在面对摄像头的时候在较长时间保持同一个状态的基础上微动，然后就会得到成批的同类数据，便于分类；第三种是基于自然状态下的分类方式，让志愿者处于自然状态，后期通过程序一键分类到不同目录下，同时对坏数据进行清理。小组成员主要采取了后两种方式。

对于数据标注，基于上述的分类方法，小组成员开发出了一个非常实用的C++程序，能够对目录下的同种图片以文件重新命名的方式进行标注，并生成相应的TXT索引文件。

特别地，还注意到了控制两种数据的量级相当，确保训练较为准确。

### 3 算法原理

本程序核心使用了CNN网络架构对图片进行处理，Loss函数选用普适的交叉熵损失函数计算公式，Relu激活函数。卷积神经网络可以大大简化学习所需的参数的数量级，使得学习变得相对容易。

## 4 实验细节

### 4.1 模型训练

在保证训练与测试数据集不变的前提下，经过调整网络结构发现同样的循环数、学习率与Batch数下，增加神经网络层数可以提高训练效率，达到较高的准确率。增加循环数有利于出现更多的网络参数组，便于择优存储。尝试了更换激活函数从RELU到SIGMOD，发现几乎无法达到训练目的。更换NLLL负对数损失函数，同样无法增加准确率。特别地，在探索Batch\_size对训练结果的影响时结果如图：

```

本次训练相关参数: EPOCH : 10 BATCH_SIZE : 5 LR : 0.0001
Accuracy of the network on the 10000? test images: 48 %
Accuracy of the network on the 10000? test images: 83 %
Accuracy of the network on the 10000? test images: 97 %
Accuracy of the network on the 10000? test images: 96 %
Accuracy of the network on the 10000? test images: 96 %
Accuracy of the network on the 10000? test images: 94 %
Accuracy of the network on the 10000? test images: 97 %
Accuracy of the network on the 10000? test images: 87 %
Accuracy of the network on the 10000? test images: 90 %
Accuracy of the network on the 10000? test images: 87 %

```

图 1: Batch=5

```

本次训练相关参数: EPOCH : 10 BATCH_SIZE : 10 LR : 0.0001
Accuracy of the network on the 10000? test images: 50 %
Accuracy of the network on the 10000? test images: 81 %
Accuracy of the network on the 10000? test images: 96 %
Accuracy of the network on the 10000? test images: 80 %
Accuracy of the network on the 10000? test images: 93 %
Accuracy of the network on the 10000? test images: 91 %
Accuracy of the network on the 10000? test images: 83 %
Accuracy of the network on the 10000? test images: 94 %
Accuracy of the network on the 10000? test images: 85 %
Accuracy of the network on the 10000? test images: 88 %

```

图 2: Batch=10

```

本次训练相关参数: EPOCH : 10 BATCH_SIZE : 20 LR : 0.0001
Accuracy of the network on the 10000? test images: 52 %
Accuracy of the network on the 10000? test images: 52 %
Accuracy of the network on the 10000? test images: 94 %
Accuracy of the network on the 10000? test images: 89 %
Accuracy of the network on the 10000? test images: 96 %
Accuracy of the network on the 10000? test images: 97 %
Accuracy of the network on the 10000? test images: 93 %
Accuracy of the network on the 10000? test images: 91 %
Accuracy of the network on the 10000? test images: 89 %

```

图 3: Batch=15

对卷积核大小的探索，发现增大卷积核并未有效提升训练效率。

## 4.2 关于GUI

- 1 整个GUI设计的核心理念为：使用方便简单，界面美观大方，功能友好，用户体验至上。
- 2 AGUI.MainRun 为主运行文件；AGUI.mainform为主窗口文件（整合全部窗口）；剩下均为界面设计和界面函数功能的代码文件（其中不带后缀的为界面设计文件），例如带后

缀‘1’的文件为逻辑函数功能文件，例如AGUI.father与AGUI.father1。

- 3 整个GUI文件的逻辑链：先设计好两个主要窗口界面father和child，再定义一个类，用于实现二者的函数功能，分别装在AGUI.father1和AGUI.child1文件里面；在AGUI.mainform里，定义一个新类，但只继承AGUI.father1（此窗口功能十分简单，仅为‘友好交互’设计）。在此新类中，另外设计一个系统托盘，完善功能。第一个窗口father中定义有槽函数，直接链接到我们最重要的第二个窗口child，在child中实现最终功能。
- 4 GUI特色：4.1 有加载界面4.2设计有系统托盘，有用户说明等4.3 第二个界面可拖动，所有界面均不存在长宽比问题4.4 有轻度美化

## 5 心得体会

我主要负责的是GUI功能的设计，将训练完成的CNN网络应用与GUI视频界面上对眨眼进行判断和统计，同时也参与了CNN网络的搭建以及数据集的采集。

我负责设计的GUI功能主要是：从摄像头中采集图像、在图像中标注出人脸、把获取的人脸丢入训练好的CNN模型，获取睁眼或闭眼的判断、根据实时更新的判断，对于人的眨眼进行统计和分析、根据分析数据，显示在GUI界面上，并给出适当的提醒。

在设计的过程中，也出现过很多问题，虽然训练出来的模型对于睁眼闭眼判断的准确率已经很高，但是，实际应用中出现了很多不适应。比如，使用Opencv的人脸检测模型的时候，经常会检测到两个及以上的人脸（有的甚至检测到非人脸），多个人脸就会对于眨眼的统计造成很大的影响，最后，决定输出面积最大的人脸图像进行检测。还有，对于睁眼闭眼数据的分析，也修改了很多分析方法，增强用户体验，例如从最开始的统计平均眨眼频率，到后面，没十秒进行一次统计数据刷新，计算瞬时的眨眼频率，在细节方面有很多改进。在实际使用的过程中遇到过一些困难，如移动头部、光线、眼镜反光等都会降低其准确率的判断，这方面，我们也调整了训练集的参数，使用更好的训练模型来改善这个问题。

刚开始接触这个任务的时候，确实对于怎么实现任务比较迷茫，有一个大的方向，但是对于具体工作怎么去做还是未知数，开始选题的时候，也是提出了一大堆方案，但是在讨论可执行性的时候，由于我们对于深度学习了解的不足，所以也是放弃了一些很有意思但是很难的课题，最后选择的这个眨眼判断的程序虽然实现起来简单，但是在生活上也确实有很广泛的应用，除了针对使用电脑的抗疲劳之外，还可以应用于检测疲劳驾驶，甚至检测上课是否犯困，这些与眨眼息息相关的生活场景都是我们应用的方向。

虽说是看似简单的任务，但在实际实现的工作中，也确实出现了许许多多的困难，最开始找到一个CNN网络的例子是MINST数据集的测试，它使用的是已经处理好的数据集，但是如何加载我们的数据集呢？本来是以为简单的把jpg的图片放进模型训练就可以，后来发现事

情没有那么简单，我们写了一个标准类来处理数据集，光是说把数据集加载进入训练的模型就花了我们挺长时间去找各种资料，看代码，同时也写了很多的附加程序来统一处理数据集照片的格式和大小等。经过了几个星期的鏖战，终于训练出来第一个能用的模型，便开始多线程作战，训练数据、调整参数、GUI界面……最后的工作相比于开始没有任何头绪来说就相对轻松一些，虽然在我负责GUI的设计方面也遇到了很多困难，但也都找到了对应的解决方法。

整个大作业，就是一个深入学习深度学习的过程，从别人的代码和案例中学习所需要的东西，再添加到自己的代码中，也从一个什么都不会的小白，开始逐渐了解到了实现机器学习的整个过程。深度学习这个事情也确实使我大开眼界，也感受到了人工智能的强大之处，很多我们的工作，使用人工智能来做会方便简单地多，给我打开了新世界的大门。最后也感谢我可爱的队友们，是我们共同的不懈努力，使得任务能够圆满完成。

——From 邱岳峰

对本课程的学习从安装各种软件以及各种包开始。开始的时候认为只要嵌套一个模型就好了，可是后来发现“生搬硬套”并非捷径，还需要自己看懂代码才可以，再次用实践证明了编程需要仔细与耐心，不能急躁。在训练模型的过程中，核心问题在于之前没能完全理解深度学习中关于张量等数据类型之间的关系，造成逻辑上的混乱。

本课程的大作业不同于之前的数据结构、C语言等课程，用python编写工程性大作业不必考虑一些底层的代码结构，重点在于对各种功能性函数的组合。特别想要指出的是，在研究如何可视化训练过程时，在网上搜了很多方法，但是都未能有效应用，几次到了放弃的边缘，最后还是用TensorboardX实现了预期功能，这是着实令我感到兴奋的。本次大作业本人对Git版本管理的应用很多，的确感觉到了其方便之处现在回过头看历次提交的备注，感觉到了自己的小小进步。

```
$ git log
commit 49f63215c0bc03d1b4bb34d655e2c37da7f85972
Author: BravoEven <572973477@qq.com>
Date: Fri Jun 7 00:04:49 2019 +0800

    20190607 现在就差挑选最准确模型了 <未测试>

commit 3501e4465db2ac6e5c8bca28c06d606a174fa9e9
Author: BravoEven <572973477@qq.com>
Date: Thu Jun 6 16:43:44 2019 +0800

    20190606 基本完成了功能，但还未美化代码。高考学子加油！

commit 5d5f2f0af2ed6861adcd40390fb57d5cf9d91a98
Author: BravoEven <572973477@qq.com>
Date: Mon Jun 3 23:40:27 2019 +0800

    20190603 杂乱的组合，难过

commit 9badc9b2f46673ae622d959710be20accec3f5434
Author: BravoEven <572973477@qq.com>
Date: Sat Jun 1 12:29:55 2019 +0800

    20190531

commit 5c6122fce038fc64d77c1469592f4ea98631a8a8
Author: BravoEven <572973477@qq.com>
Date: Wed May 29 12:07:02 2019 +0800

    20190529 改进了Judge文件使其更加稳定，删除了Normal的一些测试代码

commit 162a74eb576cf527cdfcc1cc17cf73377961c861
Author: BravoEven <572973477@qq.com>
Date: Sun May 26 22:51:56 2019 +0800

    2019056 增加了数据库和网络层数

commit 42e2be509a9f3fd48c86c964de3a27c2fe0627b
Author: BravoEven <572973477@qq.com>
Date: Sat May 25 16:59:40 2019 +0800

    20190525
```

——From 石志远

我在项目中主要负责数据集整理和模型训练。由于数据集规模较大（共有接近9000张图片），我使用python和C++做了几个脚本以提升整理数据集的速度。在训练过程中我发现，对于二分类问题，应尽可能使数据集中两种样本的数量相同或相近，这样可以有效地提高模型的真实准确性。训练过程中学习率对结果的影响较大，学习率越小，模型达到最优的机会就越大，如果学习率过高，模型的准确程度就会在“谷底”两侧跳来跳去，无法达到极值。同时，数据集中的噪音是要尽量避免的，噪音不会提高模型的稳定性，只会降低模型的准确性。训练模型绝对是一个集智慧和人品于一体的工作，好模型需要好参数，更需要好人品。

——From 王铭浩

如同第一次大作业一样，这次的深度学习大作业也几乎是现学现卖，每一个部分都要学好几天才能开始写（CtrlC+CtrlV）代码，特别在听到老师说“我们的作业挺简单的”之后，内心尤为崩溃，但‘在科大的风雨中，这点痛算什么，擦干泪不要怕，至少我们还有数理基础和python。。。’

此次任务中，我主要负责GUI部分的设计，而也借此机会对GUI的设计的PyQt的使用有了更全面的认识。由于第一次作业GUI比较简单，所以参考价值有限，为了做出更完整的GUI（比如包括字体美化、托盘化、进度加载、窗口控件美化设置等），足足看完了大半本《PyQt5快速开发与实战》，并在网上疯狂Google——不过网上的大部分界面设计是C语言版本的或者PyQt4的。在此次GUI的设计中，我考虑了非常多的细节，诸如在托盘处设计的菜单中有一个“还原”选项，那么这个还原到底是要做到什么功能才算是人性化呢？经过多次修改后，由最初的重启app换为窗口再现。当然，更多的细节体现在各个窗口的连接和窗口本身的设计上面。

人生苦短何不学python？——经历了一学期的python学习后，对此话最大的感悟莫非三点。一是python的确是一种高效的语言；二是学习python的同时更是学习如何学习，如何耐住性子debug、如何正确问问题、以及如何整理复杂的思路并实现之；三是与队友伙伴一同合作完成项目，在交流中相互学习，在学习收获友谊——非常感谢本次课程，给我和队友一起打怪升级的机会！

——From 陈一龙

## 6 成员大体分工

邱岳峰 答辩GUI编写与调试、收集数据集、提供加载训练集的代码框架

陈一龙 GUI编写与调试、收集数据集

王铭浩 调试神经网络参数、收集数据集与后期调整、编写数据集整理脚本

石志远 网络结构、训练参数可视化（loss趋势、特征图等、收集数据集、编写分类器、提供训

练网络代码框架、文档、优化训练源代码