

人智第二次大作业报告

姓名：姜卓

班级：自 74

学号：2017010550

Public 0.888 101 wonderful_jz_

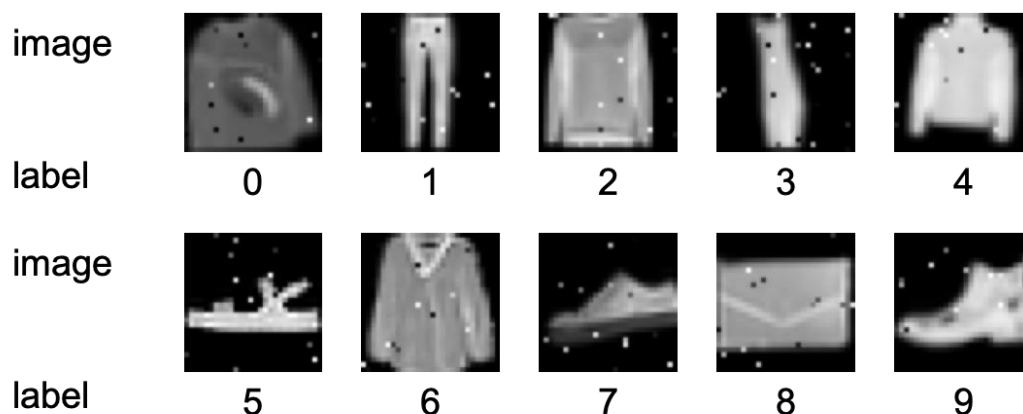
Private 0.893 94 wonderful_jz_

目录

- 人智第二次大作业报告1
 - 一、任务描述.....2
 - 二、任务实现.....2
 - 1. 数据导入和预处理..... 2
 - (1) 自定义 train_dataset 类和 test_dataset 类..... 2
 - (2) 图像的预处理..... 3
 - 2.选择网络模型.....3
 - (1) resnet 网络模型的优点.....3
 - (2) 超参的设置..... 5
 - 三、训练效果评估.....5
 - 1.评估方式.....5
 - 2.调参分析.....6
 - 四、参考资料.....7

一、 任务描述

本次任务旨在利用深度学习网络训练图像分类模型，完成 10 类图像的分类问题。训练集包含图片信息和标签信息，图片信息存储在 train.npy 文件，标签信息存储在 train.csv 文件中。图片示例如下图。



每个 npy 文件包含一个 $N \times 784$ 的矩阵， N 为图片数量。矩阵每行对应一张 28×28 的图片，train.csv 文件包含训练集的标签，含 image_id 和 label 两列，共 30000 行，image_id 对应矩阵中的行下标，label 为该图片的类别标签。

在预测环节，需要利用训练好的模型对测试集中的 5000 张图片进行分类，预测结果应生成 submit.csv 文件，同样包含 image_id 和 label 两列，共 5000 行，每行对应一张图片的结果。

二、 任务实现

1. 数据导入和预处理

(1) 自定义 train_dataset 类和 test_dataset 类

以 train_dataset 为例，train_dataset 类包含 __init__、__getitem__ 和 __len__ 三个方法。

__init__ 是构造函数，用 numpy.load() 实现了 npy 文件的导入。同时导入 csv 文件的标签信息。npz 文件中的图片信息导入后要将第二维张量（784）张成三维 ($28 \times 28 \times 1$)，以便后续图像的处理。__init__ 方法还利用 torchvision.transforms 定义了对图像的预处理。

__getitem__ 和 __len__ 方法分别实现了图像、标签信息的索引，和返回图像的长度。

(2) 图像的预处理

为了消除图像的共性特征，凸显出图像的个性特征，以便深度学习网络更好地捕捉和学习图像特征，我们利用 `transforms.Normalize(mean,std)`，减去图像的均值，再除以标准差，将图像标准化处理。

为了让深度学习网络充分学习图像的各种状态下特征，使网络模型具有一定的鲁棒性，减小过拟合现象，我们将图像信息先利用 `transforms.ToPILImage`，将 `numpy` 类型转成 `image`，再进行随机水平旋转 (`transforms.RandomHorizontalFlip()`) 和随机旋转 (`transforms.RandomRotation()`)

同时我还用了将图片进行 `Resize`，把图片扩大到边长为 224，据验证，这种处理对提升网络模型的分类准确率有一定提升，可能是提高了数据量的缘故。

2.选择网络模型

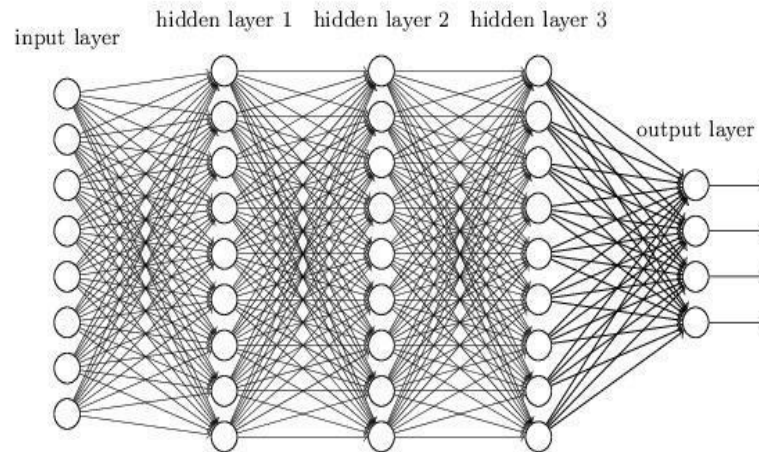
(1) resnet 网络模型的优点

网络深度对图像分类的效果有很大的影响。一般的认识上，我们觉得网络设计的越深，所能达到的识别和分类效果就会越好。然而根据搜集资料，我发现事实并非如此。常规的网络堆叠在网络层数很深的时候反而效果越来越差。其中的原因之一就是网络越深，梯度消失现象就越来越明显，限制了网络的训练效果。而 `resnet` 模型，恰恰能对这一问题有很好的解决。

梯度消失：在神经网络中，当前面隐藏层的学习速率低于后面隐藏层的学习速率，即随着隐藏层数目的增加，分类准确率反而下降了。这种现象叫做消失的梯度问题。

这里我们来模拟一下梯度消失的形成过程：

以一个含有三层隐藏层的简单神经网络来说，当梯度消失发生时，接近于输出层的隐藏层由于其梯度相对正常，所以权值更新时也就相对正常，但是当越靠近输入层时，由于梯度消失现象，会导致靠近输入层的隐藏层权值更新缓慢或者更新停滞。这就导致在训练时，只等价于后面几层的浅层网络的学习。【1】

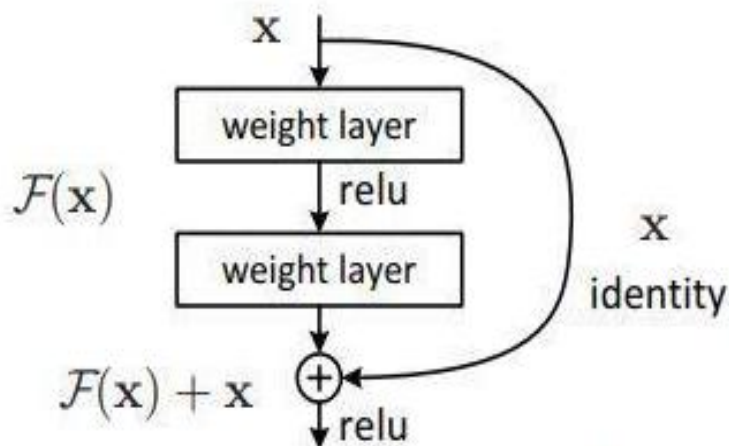


在进行参数更新时，我们利用损失函数对参数求导，假设激活函数选择 Sigmoid 函数，其值在 $0 \sim 1$ 范围，对它求导得到：

$$S'(x) = S(x) (1 - S(x))$$

取值范围在 $0 \sim 0.25$ 之间，通常初始化的网络权值小于 1，因此在层数增多时，小于 1 的值不断相乘，最后就导致梯度消失的情况出现。

Resnet，即深度残差网络，它的一个基本残差单元如下图：



相比较于以前网络的直来直去结构，残差中有很多这样的跨层连接结构，这样的结构在反向传播中具有很大的好处，见下式：

$$\frac{\partial \mathcal{E}}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L} \frac{\partial \mathbf{x}_L}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L} \left(1 + \frac{\partial}{\partial \mathbf{x}_l} \sum_{i=l}^{L-1} F(\mathbf{x}_i, \mathbf{W}_i) \right)$$

式子的第一个因子 $\partial \mathcal{E} / \partial \mathbf{x}_l$ 表示的损失函数到达 L 的梯度，小括号中的 1 表明短路机制可以无损地传播梯度，而另外一项残差梯度则需要经过带有 weights 的层，梯度不是直接传递过来的。残差梯度不会那么巧全为 -1，而且就算其比较小，有 1 的存在也不会导致梯度消失。所以残差学习会更容易。

因此我在本次任务中选择了这一网络。

（2）超参的设置

a.学习率:

学习率指在优化算法中更新网络权重的幅度大小，如果学习率设置过大，损失 loss 可能会上下震荡，模型不收敛，过小则模型收敛缓慢，所需训练周期更长。学习率可以是恒定不变的，也可以逐渐下降，或者自适应的，我的学习率取法是初始值设为 0.003，每 10 个 epoch 衰减为原来的十分之一，经验证得到还可以的训练准确度（89.4%）。

b.批次大小 batchsize

这个参数代表每次训练神经网络送入模型的样本数，通常在算力可以保证的情况下，大批次可以使网络更快收敛，但由于自己计算机内存有限，设置过大容易使程序内核崩溃，顾设为 32。

c.优化器 optimizer

```
设置如下: optimizer = optim.SGD(net.parameters(), lr=LR, momentum=0.9, weight_decay=5e-4)
```

根据本人的查阅资料，使用随机梯度下降（SGD）虽然收敛缓慢，但是加入动量 Momentum 可以加快收敛，而且带动量的随机梯度下降算法有更好的最优解，即模型收敛后会有更高的准确性。若追求训练速度可以改用 Adam 优化器，该训练器收敛速度很快，由于本人后期幸运的得到了 cuda 资源，故采用前者的优化方式，参数设置如上。

d.迭代次数 epoch

迭代次数是指整个训练集输入到神经网络进行训练的次数，当测试错误率和训练错误率相差较小时，可认为当前迭代次数合适；当测试错误率先变小后变大时则说明迭代次数过大了，需要减小迭代次数，否则容易出现过拟合，根据训练时的效果进行不断调整，最终取 epoch=40.【2】

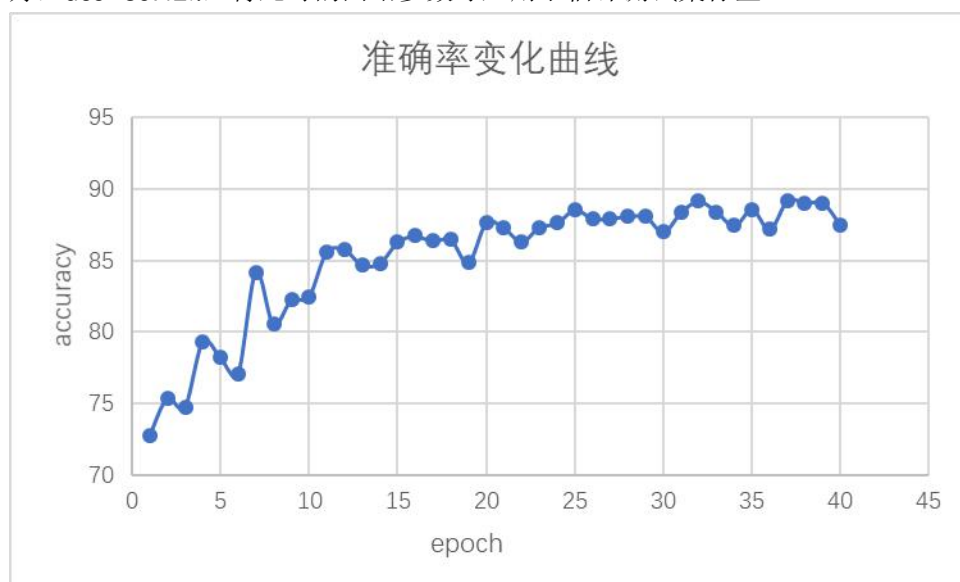
三、 训练效果评估

1.评估方式

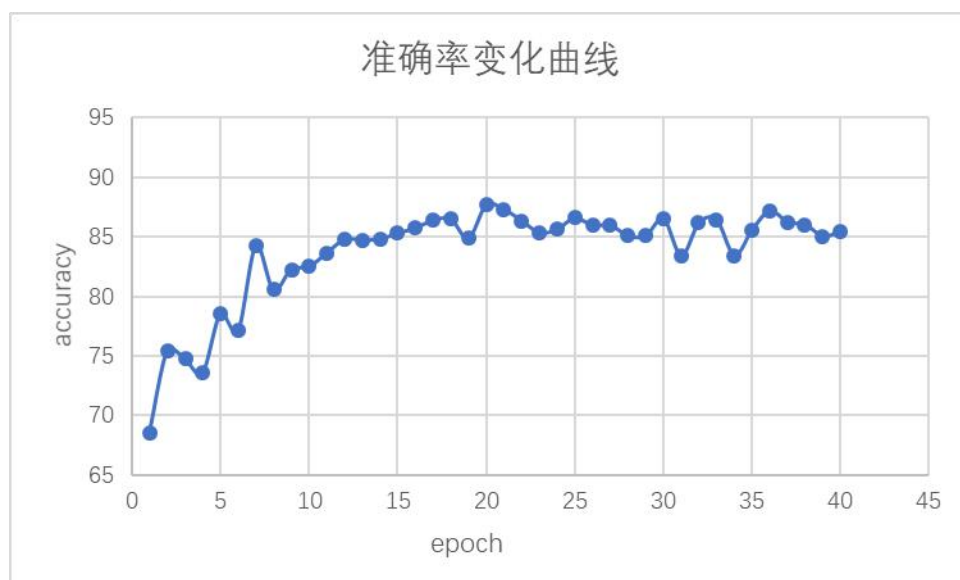
将 train.npy 中的图片按 5: 1 分为训练集和验证集，用训练集中的 25000 张图片训练 Resnet 网络，再将验证集 image 输入网络得到预测的标签信息，再与实际标签信息进行比较，得到对 Resnet 网络分类准确率的评估。

2.调参分析

将分类网络的准确率记录并绘制曲线，得到 acc-epoch 曲线，下图为提交的最终网络的表现情况，可以观察出在 epoch=40 的时候，准确率将趋于收敛，在 epoch=32 时效果最好，acc=89.42%，将此时的网络参数导入用于估计测试集标签。



上图的调参结果时本人不断探索得到的，其中很多参数对网络的表现都有着微妙的影响，我并不能将其详细地探究出来，只能用最基础的控制变量法尝试着摸索。比如上图的学习率为 0.003，0.0003，0.00003 梯度下降，当我把学习率对应调成 0.005，0.0005，0.00005 时，表现就会变差，如下图所示。



类似对分类效果影响比较大的参数还有 epoch 的设置，transforms.Resize()中参数的设置，数据增强的强度等等。不一一画出展示。当然我很清楚的明白我这种调参方法是充满着盲目性的，因为在对准确率的影响上，各个参数之间应该会存在耦合，即各个参数之间不是

独立的，因此深度学习网络性能的进一步提升应该是可以实现的，这也将是我后续研究和学习的目标。

四、参考资料

【1】cdsn——梯度消失和梯度爆炸问题详解。<https://www.jianshu.com/p/3f35e555d5ba>

【2】深度学习超参数调整总结。<https://www.cnblogs.com/andre-ma/p/8676220.html>