EE369 课程大作业

基于Densenet与数据增强的M3DV项目报告

姓名 赵伟基　学号 517021910883

2019 年12月27日

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

项目完成情况

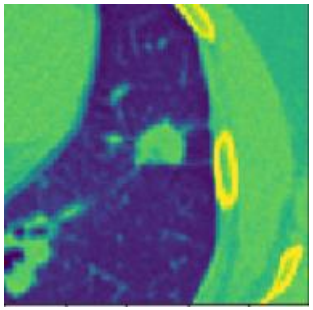
- 最终Leaderboard上的分数：0.72023
- Leaderboard上显示的名字：517021910883_赵伟基
- 总共提交次数：16+14=30
- 是否使用小号刷分：是
- 方法简述：使用3D-DenseNet的基本框架；通过对输入数据进行占空比来进行数据增强，达到比较好的划分效果。
- 主要使用的代码框架：Keras
- 模型大小（MB）：55.8MB
- 亮点：进行了诸多数据增强例如翻转、平移、反射等测试。加入更多dropout层用来减缓过拟合，同时减少模型层数使训练速度较快。
- 代码链接： https://github.com/vikizhao156/sjtu-M3VD-master

问题描述

- **M3VD（Medical 3D Voxel Classification）**:一个肺部结节分类的问题
- 通过由CT扫描与医生给出的结节Mask，由我们训练出一个可供预测结节病灶情况二分类的一个模型。
- 参考方法：1. 3D卷积神经网络:Densenet、Resnet

  2. 防过拟合方法：早停法，强数据增强，增加dropout层

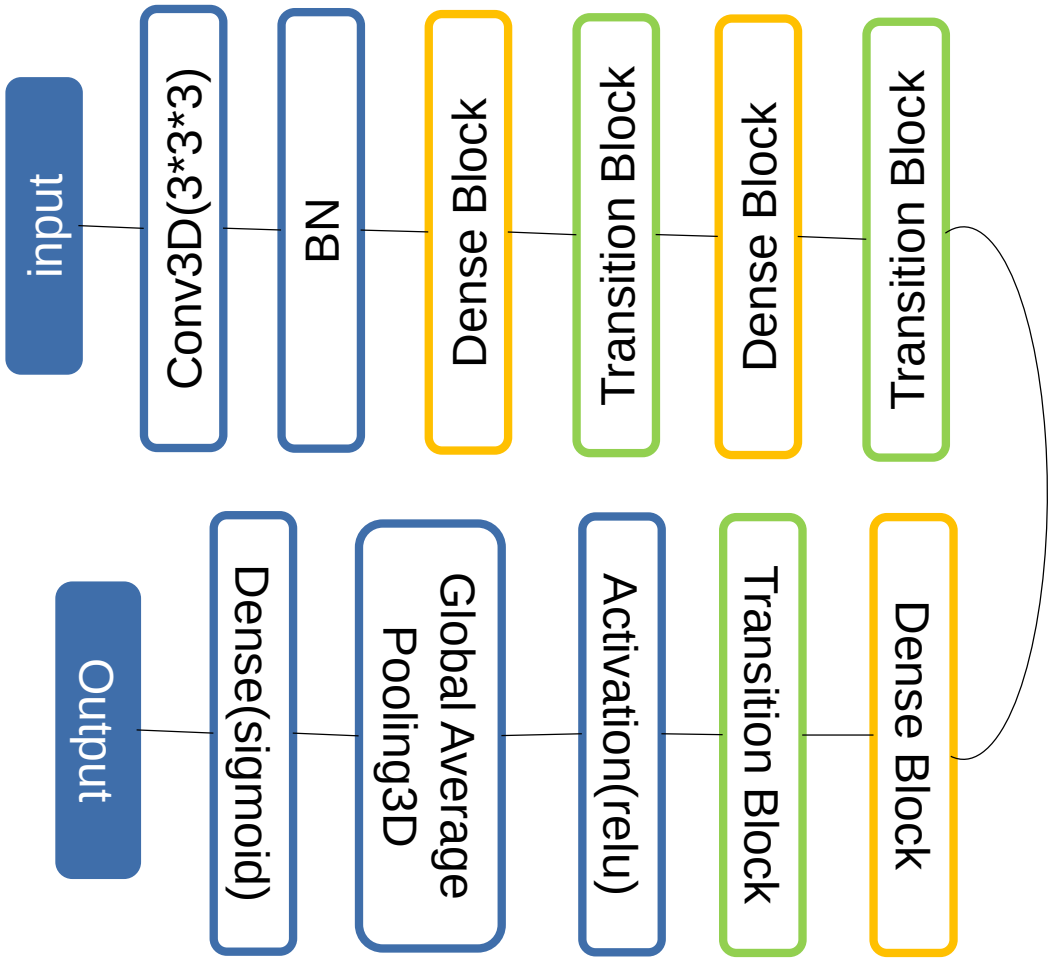
结节Mask

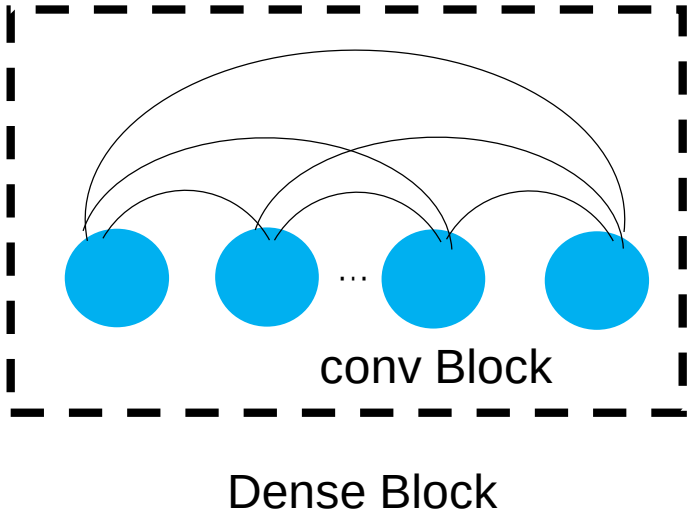


CD图

模型设计

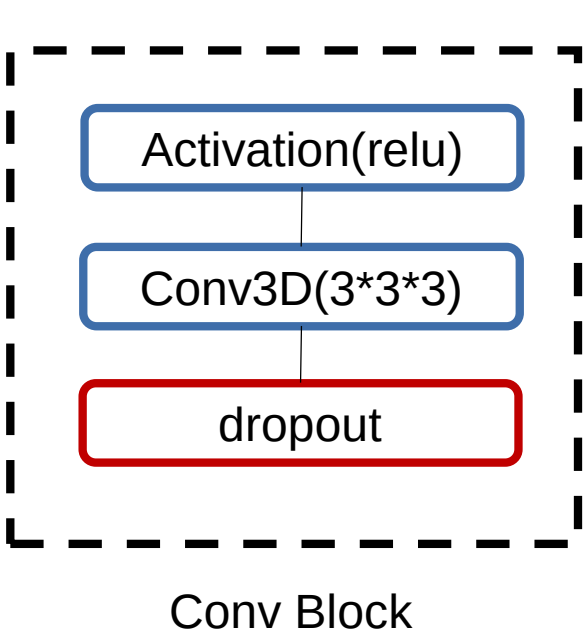
- 本模型是基于2D卷积神经网络Dense Net的基础上修改得到



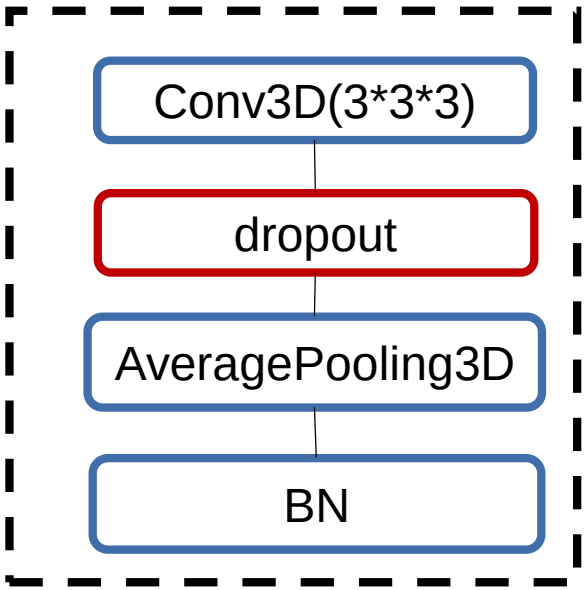


模型设计

- 本模型是基于2D卷积神经网络Dense Net的基础上修改得到



Conv Block



Transition Block

Densenet 3D的全连接设计缓解了梯度消失问题，加强了特征传播，鼓励特征重用

性能分析

- **训练稳定性**：指定参数下多次训练能均能在本地验证集达到73%左右的精度，在最终test上可以达到72.058的AUC值
- **训练时间**：在GTX1080 Ti上，1h左右即可完成训练过程
- **训练资源占用:**由于使用交大云平台，其GPU训练采取递交任务的方式，无法通过类似nvidia-smi等指令观察服务器资源占用，但可同时执行两项训练任务，其计算资源占用较小
- **模型大小**：55.8KB
- **改进方向:**
- 受限于时间没有对多个模型都进行调参处理，可以多训练几个模型来进行集成学习。
- 对数据集进行更多统计判断，对部分特征较强的数据进行cutoff操作。让网络把关注点更多的放在全局，关注隐性特征。

技巧设计

- **1.早停法（Early Stopping）**
- 为验证模型的泛化能力，除了对数据进行正则化操作外，我还引入了早停法。通过监视validation set 的泛化误差，设置一定的patience来进行。
- **2. 数据增强(Mix up)**
- 由于样本数量较少，我通过mix up的方式构建虚拟训练样本，进行数据扩充，可以通过这种方式抬高验证集精度，但最终发现在本地数据集过拟合现象严重。最后通过控制扩充数据集的数量来抑制过拟合。

$$\tilde{x} = \lambda x_i + (1-\lambda)x_j, \quad \text{where } x_i, x_j \text{ are raw input vectors}$$
$$\tilde{y} = \lambda y_i + (1-\lambda)y_j, \quad \text{where } y_i, y_j \text{ are one-hot label encodings}$$

技巧设计

- **3. 过程权重保留**
- 由于该数据集过拟合现象严重，在测试多种网络后我发现val_loss均有明显震荡现象，此时通过选取val_loss较小的过程权重，来使最后的权重有较高的泛化能力。

- **4. 数据变换**
- 尝试多种数据变换方式，包括旋转90°，镜像反射，和随机中心移位，对于数据过拟合的抑制并没有起到良好效果，最终模型将这个强化关闭。

讨论

- **数据读取方面:**

由于训练过程中一共有2*465个三维数据需要读取，因此一次将所有数据读取到缓存中能够很好的提高程序运行效率。同时由于缓存和结节大小的实际限制，读取一个数据的同时进行取核裁剪是最好的办法。

- **模型修改方面:**

由于这个数据集是三维输入，为了考察三个维度间的关系我们需要设计好三维卷积核的大小，最终抛弃了原来二维卷积核[3,3]改使用三维卷积核[3,3,3]，相应的卷积网络池化层也需要改为三维池化。

- **参数调整方面:**

通过调整learning_rate及dropout_rate等一些超参数，成功将模型收敛到一个较合适的解，证明超参数对模型收敛的速度具有关键的作用。

讨论

- **优化器选择:**

这里我对Adam和SGD两种优化器进行比较，实验中最好的解是由SGD给出的。

多次训练结果表明，Adam由于其自适应调整学习率的过程往往最后收敛到一个比较差的解，而且解比较固定。而SGD由于随机选取梯度下降的方向在高维空间更有跳跃性，所以每次训练结果相差很大，而且容易跳到更好的解上。

优化器选择	收敛代数	Training accuracy	Testing accuracy
Adam (default value)	40个epoch之后	96%	56%
SGD (lr=0.001,decay=1e-5)	60个epoch之后	89%	73%

讨论

- **损失函数选择:**

由于对结节分类是个二分类问题，选择了二分类交叉熵函数。这里需要注意与输出层的搭配关系：

sigmoid常与binary cross-entropy搭配用于解决二分类问题，softmax则常与categorical cross-entropy搭配使用。

Loss Function选择	Training accuracy	Testing accuracy
Binary cross-entropy	89%	73%
categorical cross-entropy	86%	73%

此项经过比较并没有发现非常明显的差异，这个数据集的核心还是在过拟合的处理上面。

心得感悟与改进

- **心得感悟:**

1.**网络选择**：本次数据划分是一个很隐性特征的划分问题，数据集数量很少的情况下还及其容易过拟合，在尝试了多个网络resnet、Vgg与Densenet后确定了自身其他部分算法的正确性，确认网络选择不是这个分类问题的核心所在，于是就集中精力对Densenet模型进行调参处理。

2.**数据增强**：对数据进行了旋转，镜面反射等方式，但都没有得到好的效果，后来我意识到可以通过Mix up来扩充的我的训练集大小，他确实提高了我训练过程中验证集的精度。但是导致了本地过拟合现象，通过调整mix up新产生训练集与原训练集的比重可以缓解这个问题。

Mix-up后数据集大小	验证集精度	test score（AUC）
465	0.67	67.228
2456	0.98	53.08

心得感悟与改进

- **心得感悟:**

3.**尽可能多的明白自己的数据集的基本统计信息：**在给定数据集中结节的大小不一，有的结节mask非常小，一开始我是直接提取结节部分作为我的训练集，但过于稀疏的矩阵导致训练精度一直上不去，debug过程中发现并设置结节占空比α来解决训练集太稀疏的问题：

$$x_{train} = x_{voxel} * x_{mask} * \alpha + x_{voxel} * (1 - \alpha)$$

强化结节的同时也尽可能避免过于稀疏的情况出现。这一点在前期没有做到花费了很多时间。

4.**网络深度适中即可**

由于densenet是全层密集链接，其权重参数比较多，一味的加深网络不仅会带来模型难以收敛的问题，更降低训练速度。但网络太浅导致模型偏差太大，没有办法很好地拟合出边界。最后我的densenet层数共为25层

Thank You