



EE369 课程大作业

# 基于3D神经网络的医学图像分类

董吉浩 517030910259

2020年1月3日



上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

# 项目完成情况



- 最终Leaderboard上的分数： 0.66217
- Leaderboard上显示的名字： 517030910259\_董吉浩
- 总共提交次数： 31
- 是否使用小号刷分： 是
- 方法简述： 基于DenseNet模型3D神经网络的医学图像处理
- 主要使用的代码框架： TensorFlow、Keras
- 模型大小（MB）： 1.77MB
- 亮点： 对数据进行裁剪和抠取结节，减小数据大小；对数据进行旋转、平移等操作增加数据量；对数据进行mixup处理。
- 代码链接： <https://github.com/dongjihao123/EE369-M3DV>

# 问题描述

- 本项目是一个肺部结节的分类项目，使用train\_val数据集训练并评估模型，然后使用test数据集进行测试。
- 数据集是一系列的3D肺部结节的CT扫描。每个样本由三部分组成：一个大小为 $100 \times 100 \times 100$ 、已经经过预处理的3D体素，其中心为肺部结节；由医生给出的结节分割mask；0/1标签。



3d visualization

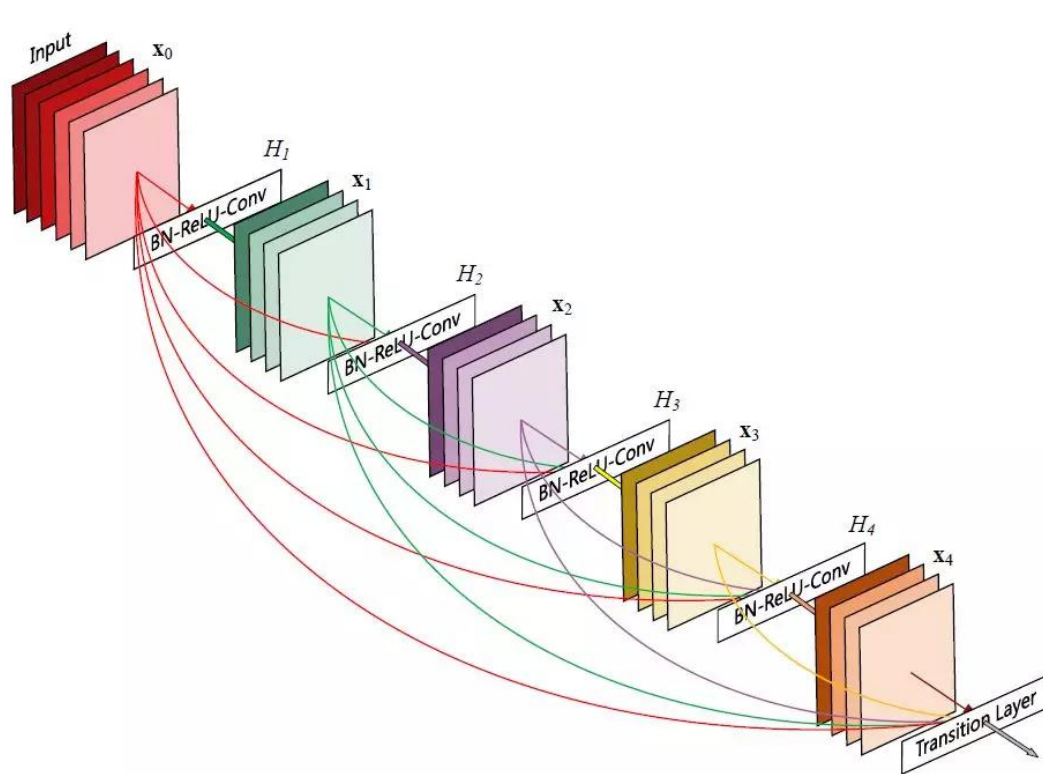


[Export to plot.ly »](#)

# 模型设计

本项目使用的训练模型为DenseNet模型，它与ResNet模型的基本思路一致，但它前面层与后面层的连接更加密集，并且通过特征在channel上的连接来实现特征重用。

DenseNet的网络结构主要由DenseBlock和Transition组成。其中，在DenseBlock中，各个层的特征大小一致，可以在channel维度上连接；Transitio层主要用于连接两个DenseBlock层，并降低特征大小。模型用到的卷积核大小均为 $3 \times 3 \times 3$ 。

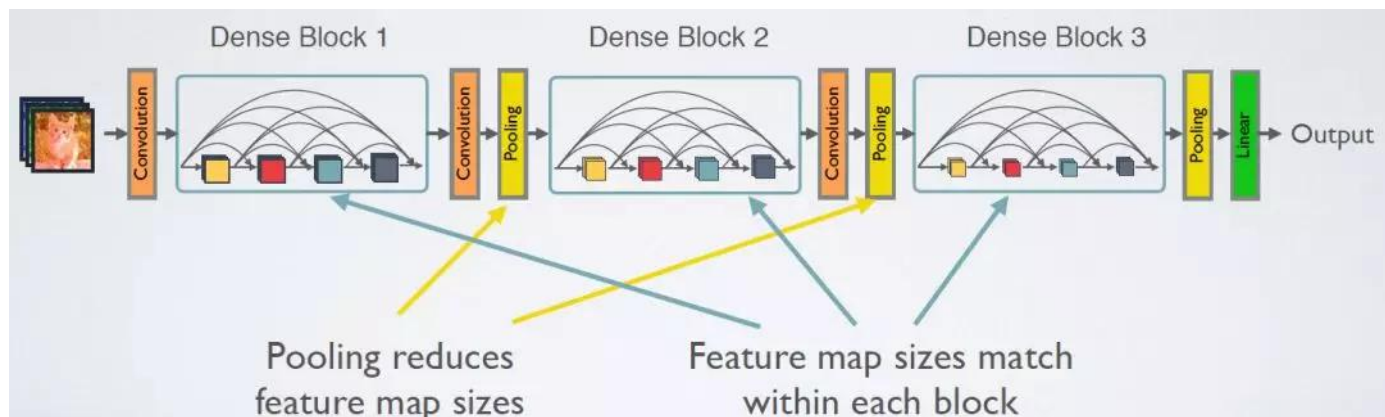




# 模型设计



- 综合来看，DenseNet的优势主要体现在以下几个方面：
  - 由于密集连接方式，DenseNet提升了梯度的反向传播，使得网络更容易训练。由于每层可以直达最后的误差信号，实现了隐式的“deep supervision”；
  - 参数更小且计算更高效，这有点违反直觉，由于DenseNet是通过concat特征来实现短路连接，实现了特征重用，并且采用较小的growth rate，每个层所独有的特征图是比较小的；
  - 由于特征复用，最后的分类器使用了低级特征。



# 性能分析



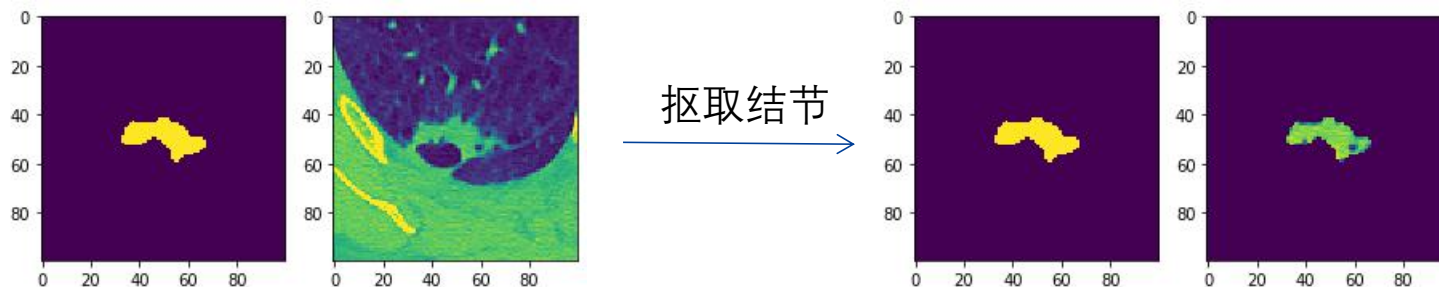
- 实验结果
  - 在batch\_size取30， epoch取26的情况下得到的结果分数为0.66217。
- 稳定性
  - 预测结果分数基本在0.66左右，较为稳定。
- 运行时间
  - 环境配置： Nvidia Tesla P4
  - 每个epoch大约需要57s左右
- 模型大小： 1.77MB

# 技巧设计



## ■ 数据的裁剪与抠取

- 实验提供的数据集每个样本的大小为 $100 \times 100 \times 100$ ，使用原始数据经常会发生内存不够的情况，因此我们将数据大小裁剪成 $32 \times 32 \times 32$ ，并利用医生给出的结节分割mask将结节抠取出来，这大大减小了数据大小。



# 技巧设计



- 数据做mixup处理
  - 对数据做mixup处理来增加训练性能。大型深度神经网络是非常强大的，但其损耗巨大的内存以及对对抗样本的敏感性一直不太理想。mixup是一个简单地减缓两种问题的方案。本质上，mixup在成对样本及其标签的凸组合上训练神经网络。这样做，mixup规范神经网络增强了训练样本之间的线性表达。研究表明，mixup可以改进当前最先进的神经网络架构的泛化能力。除此之外，mixup能够减少对错误标签的记忆，增加对抗样本的鲁棒性，并能够稳定对生成对抗网络的训练过程。

$$\begin{aligned}\tilde{x} &= \lambda x_i + (1 - \lambda)x_j, \\ \tilde{y} &= \lambda y_i + (1 - \lambda)y_j,\end{aligned}$$



# 讨论



- 数据的预处理

- 本次项目提供的数据集数量较少，每个样本又比较大，因此要想减少内存的使用并达到较好的训练结果，就必须对数据进行预处理，比如切割，旋转，平移，mixup等。

- 相关参数的调整

- 在模型基本确定的情况下，要想提高结果，需要对相关参数进行调整，主要调整的参数有batch\_size, epoch, learning\_rate等。

- 预测结果处理

- 在调整相关参数分数提升不大的情况下，我们可以使用不同模型多次运行，将得到的结果进行加权平均，得到最后结果。

# 讨论

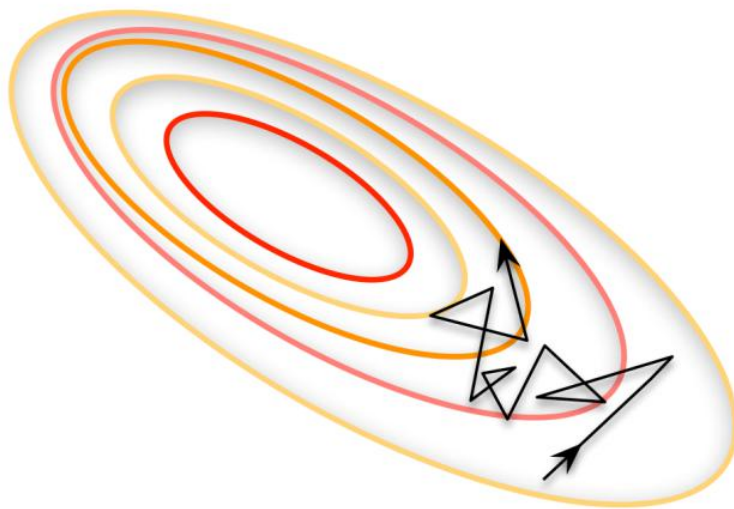


- 网络的大小和深度
  - 在实验中发现，网络并不是越大、越深越好，在实验初期使用只有3层卷积的网络也得到过0.64左右的分数，后来使用DenseNet网络，分数提升并不是很大。
- train loss 和 test loss 相关变化的分析：
  - train loss 不断下降，test loss不断下降，说明网络仍在学习；
  - train loss 不断下降，test loss趋于不变，说明网络过拟合；
  - train loss 趋于不变，test loss不断下降，说明数据集有问题；
  - train loss 趋于不变，test loss趋于不变，说明学习遇到瓶颈，需要减小学习率或批量数目；
  - train loss 不断上升，test loss不断上升，说明网络结构设计不当，训练超参数设置不当，数据集经过清洗等问题。

# 讨论



- batch\_size对网络的影响
  - Batch 的选择，首先决定的是下降的方向。如果数据集比较小，完全可以采用全数据集的形式，这样做至少有 2 个好处：其一，由全数据集确定的方向能够更好地代表样本总体，从而更准确地朝向极值所在的方向。其二，由于不同权重的梯度值差别巨大，因此选取一个全局的学习率很困难。。
  - 对于更大的数据集，以上 2 个好处又变成了 2 个坏处：其一，随着数据集的海量增长和内存限制，一次性载入所有的数据进来变得越来越不可行。其二，由于各个 Batch 之间的采样差异性，各次梯度修正值相互抵消，无法修正。
  - 因此，我们可以通过调整batch\_size来提升训练效果。



# Thank You



上海交通大学

SHANGHAI JIAO TONG UNIVERSITY