



基于深度学习的医学影像分类识别

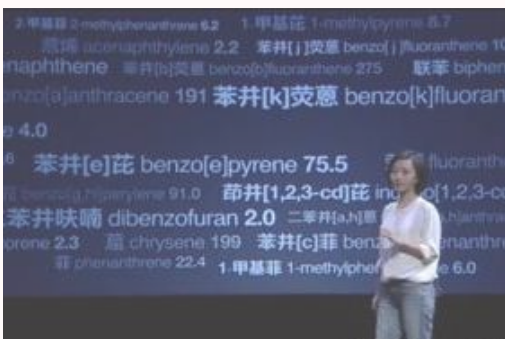
作者 周纵苇
专业 计算机科学与技术
导师 王洪凯



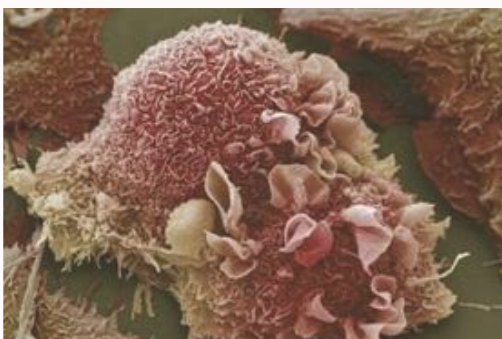
研究背景及意义

关键词：非小细胞肺癌；计算机辅助诊断

绪论综述



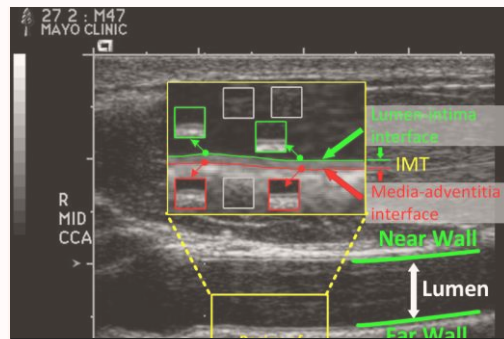
男女、城乡恶性肿瘤死亡首位均为肺癌，其中非小细胞肺癌则占肺癌总数的80%，是最为常见的肺癌。



肺癌的生物学特性十分复杂，恶性程度高，80%的肺癌患者在确诊时已属晚期。



医生肉眼观察PET/CT影像的恶性肿瘤的识别率仅为66%，特异性为93%，很容易出现误判。

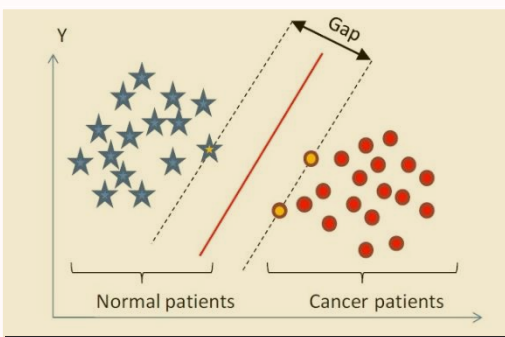


计算机辅助诊断系统可以在医疗诊断中有效地协助医生，近年来涌现出不少基于PET/CT影像的研究。

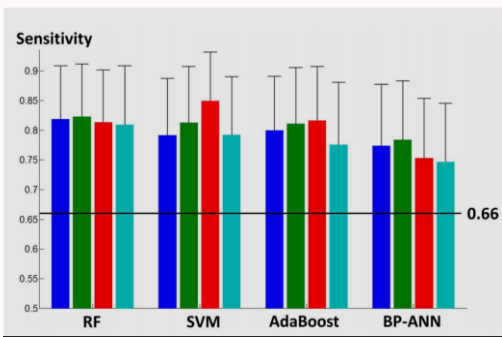
研究方法

完成情况

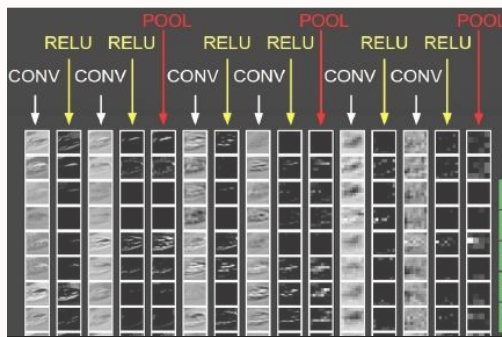
成果展示



CAD中常用的传统的分类器包括AdaBoost，随机森林，SVM，反向传播人工神经网络等。



对18F-FDG PET/CT影像进行肿瘤良恶性的分类，主流分类器能达到的敏感度为81%~84%之间。



CAD用深度学习来做，网络可以通过自主训练学习出一套较为有效的抽象特征来做分类。

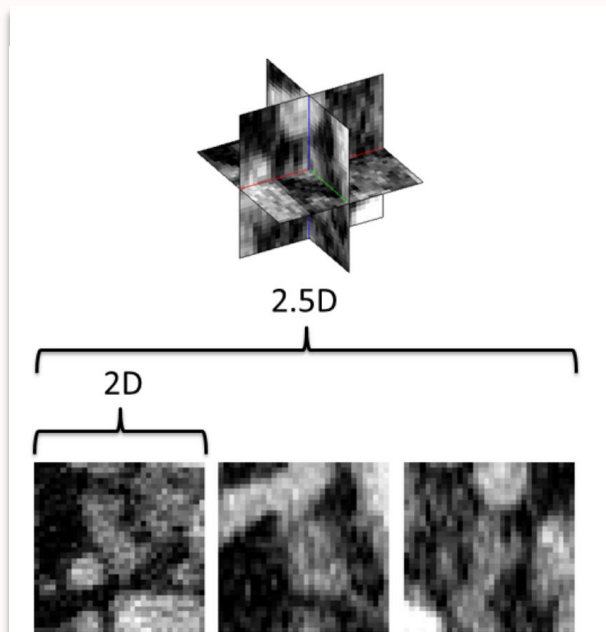


本文将测试卷积神经网络分类器，希望深度学习可以更好地用在医疗事业中，人们的生活能更好。



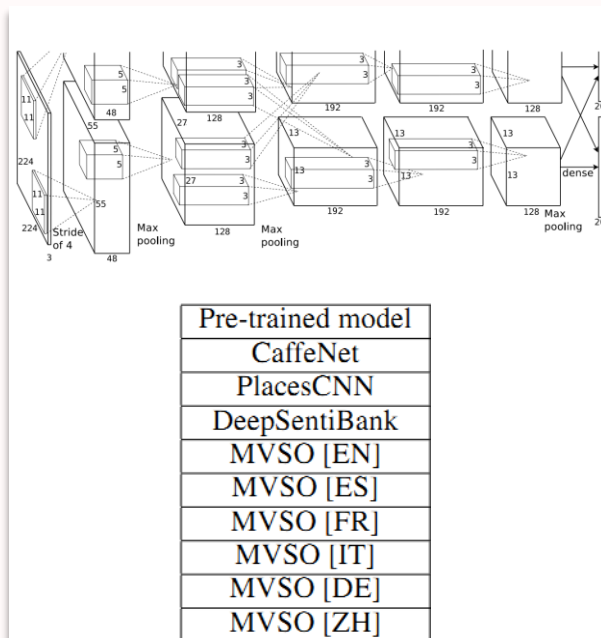
国内外研究现状

[R. Roth, et al., 2015]



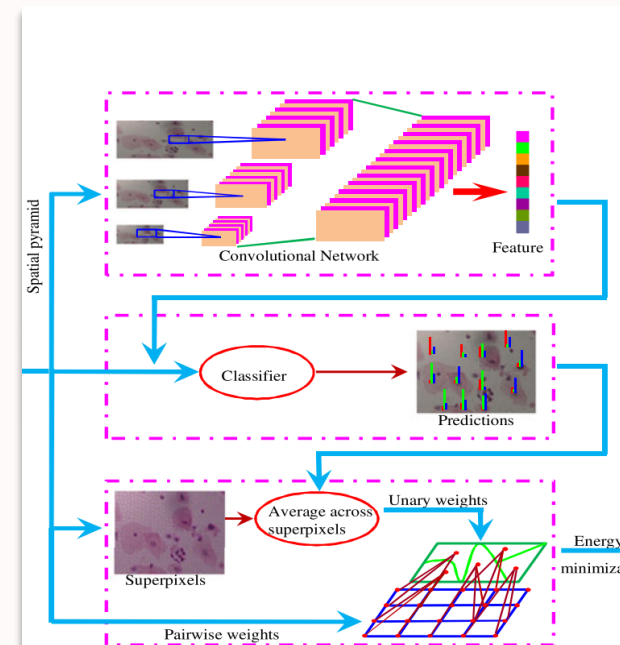
用横截面、冠状面和矢状面来代替三维信息，分别作为网络的三通道（RGB）输入训练，极大地降低了网络处理的数据量，并且准确度比二维图像更高。

[V. Campos, et al., 2016]



网络结构是AlexNet模型，分别用预先训练好的八个模型的参数作为网络参数的初始值，最后的几个全连接层参数用的是传统高斯分布的随机数初始。

[Song, et al., 2015]



将Softmax层的输入提取出来，是一个一维的长向量，作为图像的特征向量放入传统的分类器（SVM，RF等）中分类，得到结果优于Softmax。

绪论综述

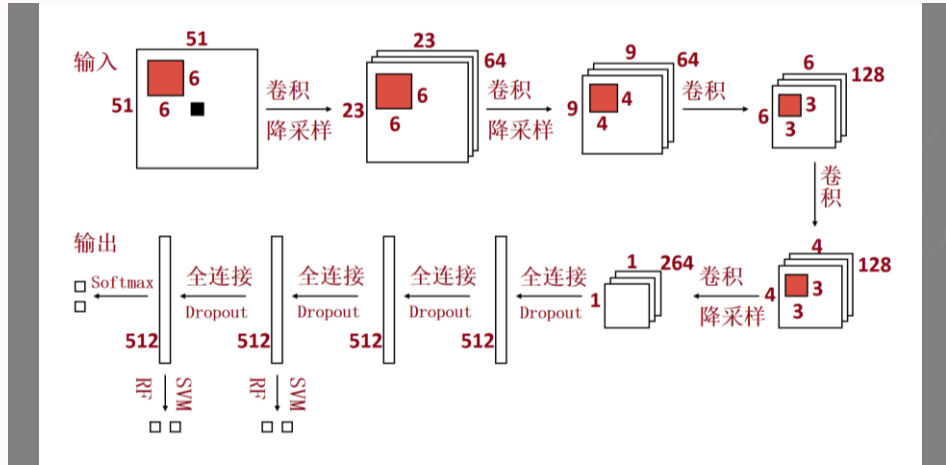
研究方法

完成情况

成果展示



卷积神经网络



□由于PET/CT这种医学图像的灰度级比其他图像形式的要大，在训练网络的时候最好将输入的patch中的值控制在一个特定范围内，本文用了两次处理：

$$X = X - \text{mean}(X), X = X / \text{std}(X)$$

□He等人在文章中给出了一个权值初始化的经验方法，当且仅当激活函数是ReLU时，权重按如下的方法初始化：

$$W = \text{rand}(n) \times \text{sqrt}(2/n)$$

□激活函数使用的是Leaky ReLU，优点是不会饱和，计算速度快，当 $x < 0$ 时有梯度，不会出现梯度消失现象。

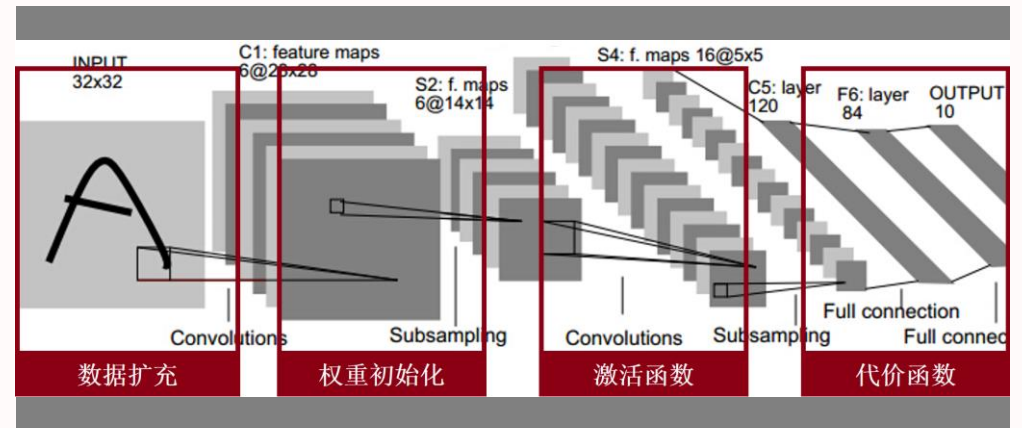
□代价函数使用的是经过L2规范化的交叉熵代价函数：

$$C = -\frac{1}{n} \sum_{x_j} [y_j \ln a_j^l + (1 - y_j) \ln(1 - a_j^l)] + \frac{\lambda}{2n} \sum_w w^2$$

卷积神经网络的特点包括四个方面：

1. 内部连接
2. 共享权重
3. 池化降采样
4. 使用很多层

卷积神经网络与传统的分类器的区别是不需要人为地定义特征，这个特点适用于那些比较复杂的图像识别，也是本文使用CNN的一个很大的原因。缺点是这方面的理论还不完备，整个网络的构建和优化仅仅停留在经验的范畴。



数据扩充

权重初始化

激活函数

代价函数

绪论综述

研究方法

完成情况

成果展示



数据平移旋转扩充

Data Augmentation

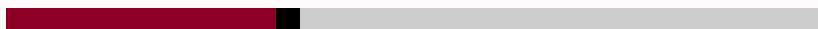
沿X轴正向平移 像素, 绕X轴旋转 度

沿Y轴正向平移 像素, 绕Y轴旋转 度

沿Z轴正向平移 像素, 绕Z轴旋转 度

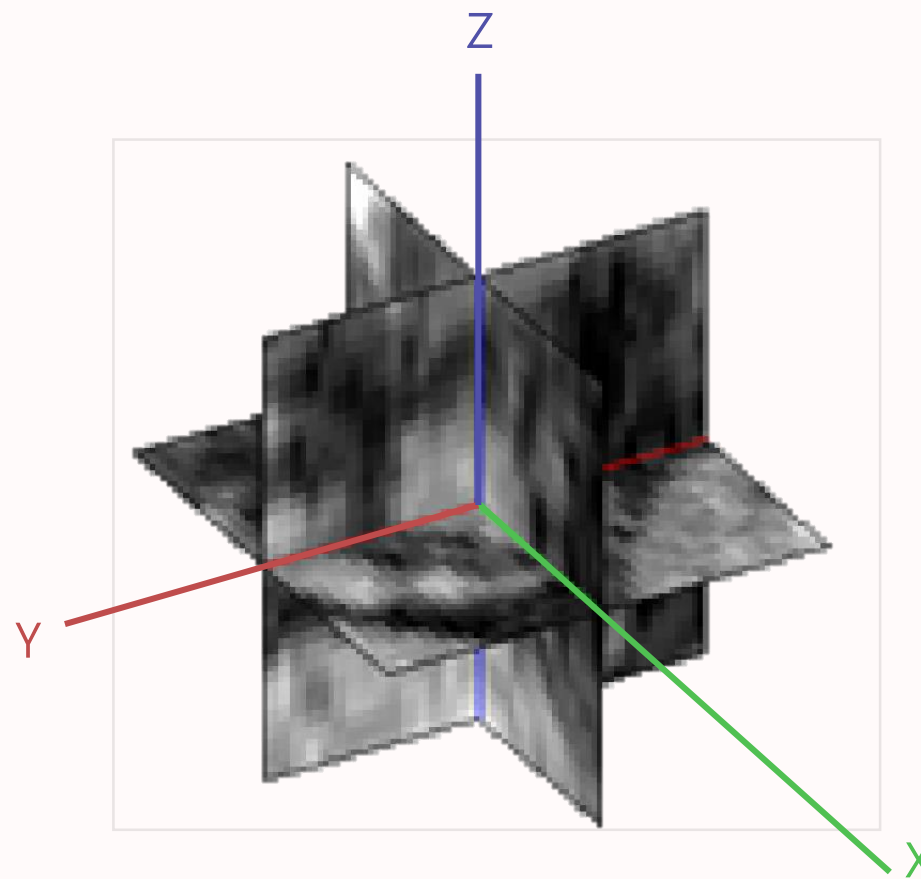
注意：由于肿瘤标记区域范围并不大，为了输入的patch中有足够量的标记区域，建议平移的像素范围在-10~10pixel之间。

2.5D切片进度



Submit

Cancel



- 考虑到程序的效率，并没有直接对三维PET/CT进行旋转平移变换，而是对横截面-冠状面-矢状面所组成的一组切面进行变换。
- 考虑到计算机内存的限制，旋转角度分别取 -10° , 0° , 10° ，平移像素分别取-2, 0, 2像素，即每个样本共扩充成**1,008,207**组PET/CT的六通道2.5D的输入，样本大小共**46.4GB**。

绪论综述

研究方法

完成情况

成果展示



提取2.5D切片patch

横截面

冠状面

矢状面

CT图像 ∨

PET图像 ∨



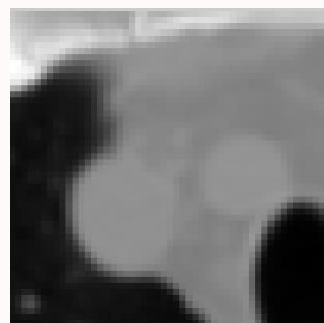
肿瘤-1

描述: CT-横截面
标签: 良性



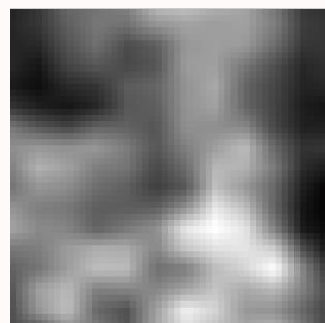
肿瘤-1

描述: CT-冠状面
标签: 良性



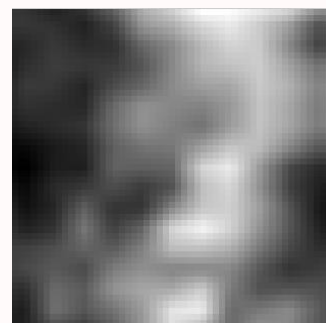
肿瘤-1

描述: CT-矢状面
标签: 良性



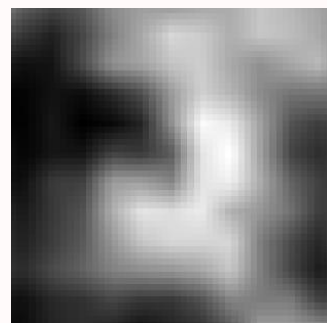
肿瘤-1

描述: PET-横截面
标签: 良性



肿瘤-1

描述: PET-冠状面
标签: 良性



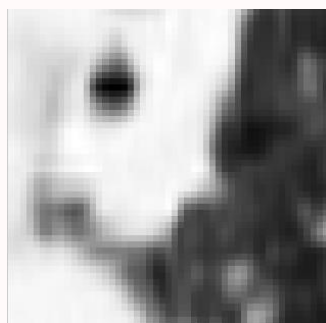
肿瘤-1

描述: PET-矢状面
标签: 良性



肿瘤-2

描述: CT-横截面
标签: 恶性



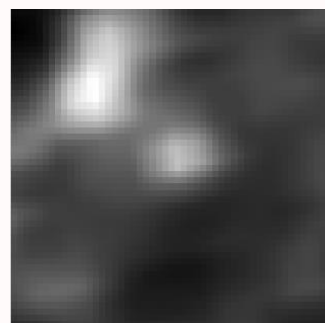
肿瘤-2

描述: CT-冠状面
标签: 恶性



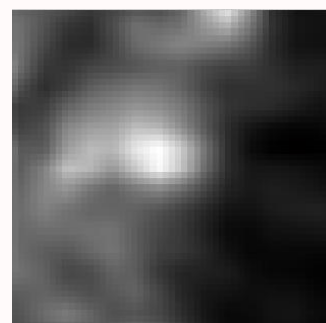
肿瘤-2

描述: CT-矢状面
标签: 恶性



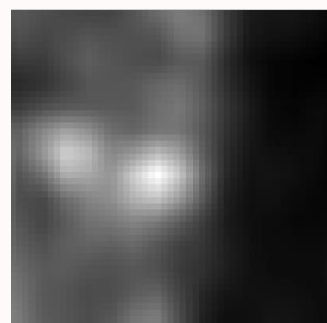
肿瘤-2

描述: PET-横截面
标签: 恶性



肿瘤-2

描述: PET-冠状面
标签: 恶性



肿瘤-2

描述: PET-矢状面
标签: 恶性

绪论综述

研究方法

完成情况

成果展示



研究进展情况

绪论综述

研究方法

完成情况

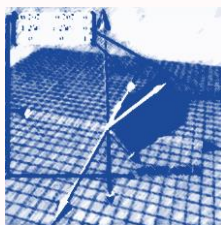
成果展示



CNN测试mnist数据集

2016.01

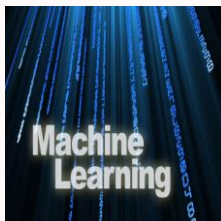
用MATLAB编写的CNN，训练一个手写字体识别的网络，得到的识别准确率达到**97%**，初步证明了CNN对于简单图像分析的可靠性。



用MATLAB做数据扩充

2016.03

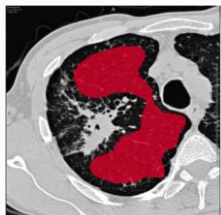
制作好的训练样本是很重要的，也是花了我最大心气的部分。由于数据量很大(**46.4GB**)，在编写MATLAB的时候我学到了很多矩阵相关的小技巧来**加速**。



与四个主流分类器做对比

2016.05

用人为设定的13个图像特征输入到四个传统分类器中(SVM, RF, AdaBoost, BP-ANN)，与不用人为设定特征的CNN进行分类效果的对比。



简易的PET/CT图像分割

2016.06

利用CNN做一个三分类的深度学习分类器，分别是“非肿瘤”，“良性肿瘤”，“恶性肿瘤”。遍历图像每个像素，做图像分割。



安装调试深度学习框架Keras

2016.02

Keras是基于**Theano**的一个深度学习框架，用**Python**语言编写，是一个高度模块化的神经网络库，支持**GPU**和**CPU**，用起来特别简单，适合快速开发。



基于CUDA的GPU加速

2016.04

我们使用Krizhevsky提出的用GPU实现CNN加速的方法，其中CUDA的安装版本是V7.5.17，Visual Studio 2012版本，显卡类型为NVIDIA Tesla K40c。



利用SVM, RF替代softmax优化分类

2016.05

编写代码，将CNN最后的全连接层提取出来放到SVM和RF中，得到的结果不如softmax好，分析原因，需要进一步的调整，这将成为下一步的其中一个研究点。



目前得到恶性肿瘤AUC为0.86

2016.06

使用的优化器是**sgd**函数，**145,800**例训练集和**54**例测试集，敏感度**79.6%**，特异性**79.7%**，训练网络时间**15分钟**，测试时平均每测试一例的时间为**2.60ms**。



64位 Python · Keras



```

Hey you, this is a trial on malignance and benign tumors detection via ConvNets. I'm Zongwei Zhou. :)
Each input patch is 51*51, cutted from 1383 3d CT & PT images. The MINIMUM is above 30 segment pixels.
>> Loading Data ...
The dimension of each data and label, listed as following:
TrData : <148162L, 6L, 24L, 24L>
TrLabel : <148162L, 1L>
Range : -3.43485 ~ 8.03914
        -4.0206 ~ 8.99869
        -3.12587 ~ 7.56241
        -1.25188 ~ 20.0045
        -1.26917 ~ 18.8637
        -1.24568 ~ 18.9091
UaData : <201642L, 6L, 24L, 24L>
UaLabel : <201642L, 1L>
Range : -3.43485 ~ 8.03914
        -4.0206 ~ 8.99869
        -3.12587 ~ 7.56241
        -1.25212 ~ 20.2061
        -1.27542 ~ 18.8637
        -1.24596 ~ 18.9091
Therefore, read in 148162 samples from the dataset totally.
By the way, data.shape[-3:] means that <6L, 24L, 24L>
>> Build Model ...
C:\Users\Administrator\Anaconda2\lib\site-packages\keras\models.py:495: UserWarning: The "class_mode" argument is deprecated in your code.
  warnings.warn('The "class_mode" argument is deprecated, please remove it from your code.')
>> Training ConvNets Model ...
Here, batch_size = 100 , epoch = 10 , lr = 0.01 , momentum = 0.9
Train on 118529 samples, validate on 29633 samples
Epoch 1/10
118529/118529 [=====] - 43453s - loss: 0.4747 - acc: 0.7903 - val_loss: 0.5141 - val_acc: 0.8134
Epoch 2/10
118529/118529 [=====] - 10104s - loss: 0.4072 - acc: 0.8181 - val_loss: 0.4869 - val_acc: 0.8211
Epoch 3/10
118529/118529 [=====] - 15145s - loss: 0.3655 - acc: 0.8374 - val_loss: 0.4562 - val_acc: 0.8178
Epoch 4/10
118529/118529 [=====] - 9613s - loss: 0.3326 - acc: 0.8555 - val_loss: 0.3965 - val_acc: 0.8634
Epoch 5/10
79300/118529 [=====>.....] - ETA: 3004s - loss: 0.3129 - acc: 0.8665

```

Keras深度学习框架配置



作者 MrGiovanni 2016.04.02 19:29*

写了14120字, 被2人关注, 获得了9个喜欢

Keras深度学习框架配置

字数3134 阅读81 评论0 喜欢2

论文引用——3.2 测试平台

项目代码是在Windows 7上运行的, 主要用到的Matlab R2013a和Python, 其中Matlab用于patch的分割和预处理, 卷积神经网络搭建用到了根植于Python和Theano的深度学习框架Keras。Keras是基于Theano的一个深度学习框架, 它的设计参考了Torch, 用Python语言编写, 是一个高度模块化的神经网络库, 支持GPU和CPU, 用起来特别简单, 适合快速开发。

直接上干货, 拒绝走弯路, 摸索72小时, 提供一切解决方案。

博客链接: <http://www.jianshu.com/p/b8a703df5318>

升级成64位Python下的深度学习框架Keras的优点是: 内存容量很大, 据说可以支持2T数据的读入:)这很好地支持了深度学习数据量超大的特点。



64位 Python · Keras



简易的深度学习框架Keras代码解析与应用

作者 MrGiovanni 2016.04.07 10:46*
写了14120字, 被2人关注, 获得了9个喜欢

简易的深度学习框架Keras代码解析与应用

字数4573 阅读61 评论0 喜欢0

再来一波论文周边吧——是关于代码的

总体来讲keras这个深度学习框架真的很“简易”，它体现在可参考的文档写的比较详细，不像caffe，装完以后都得靠技术博客，keras有它自己的官方文档（不过是英文的），这给初学者提供了很大的学习空间。
这个文档必须要强推！英文nice的可以直接看文档，我这篇文章就是用中文来讲这个事儿。

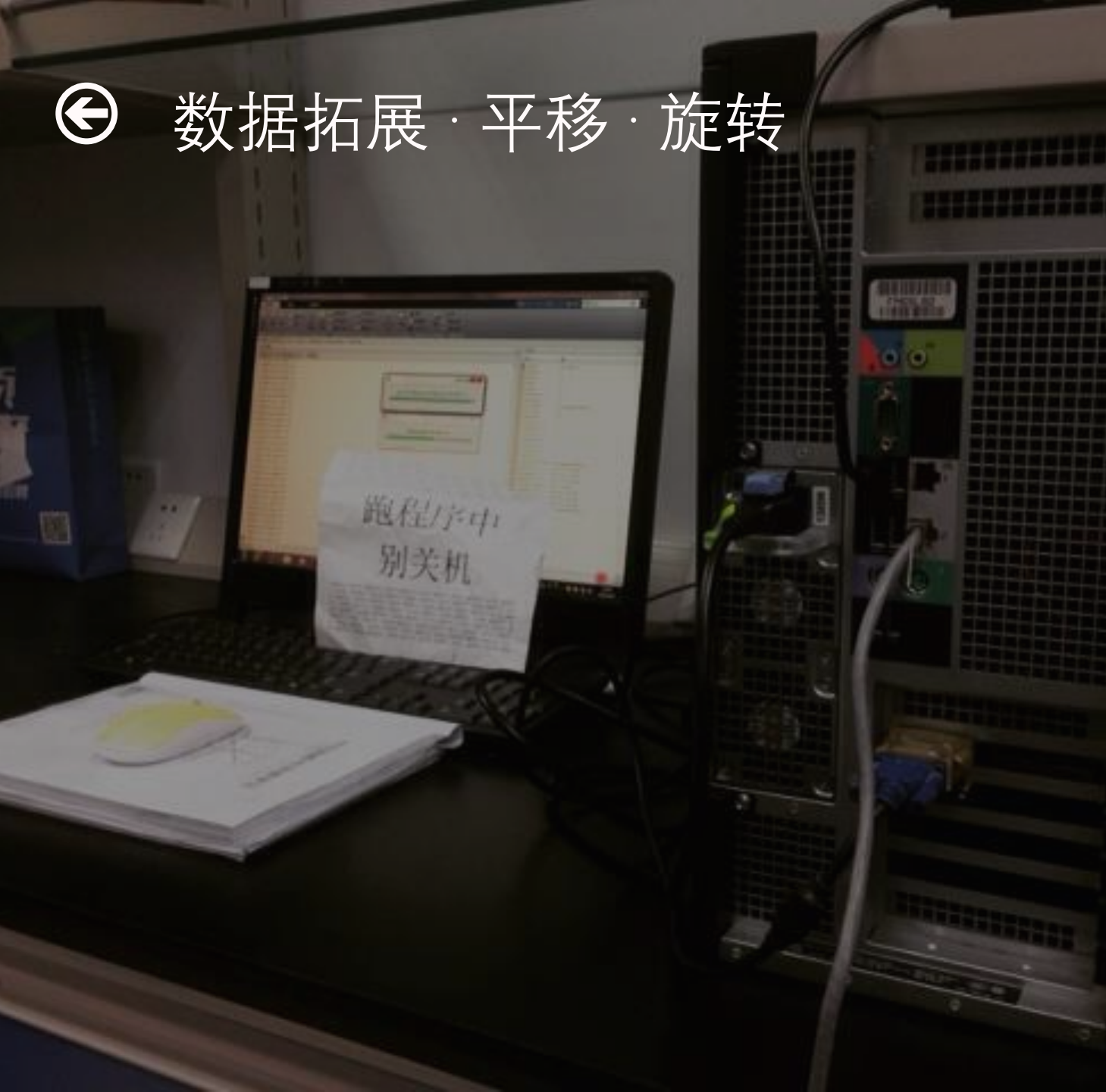
给我用的keras卷积神经网络做了详尽的注释，顺便仔细地排查了一遍代码错误。

博客链接：<http://www.jianshu.com/p/8c36a5e42d6c>

总体来讲keras这个深度学习框架真的很“简易”，它体现在可参考的文档写的比较详细，不像caffe，装完以后都得靠技术博客，keras有它自己的官方文档，十分方便。



数据拓展 · 平移 · 旋转

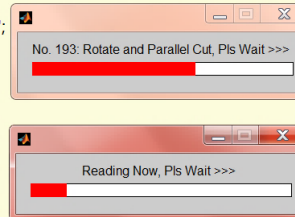


MATLAB快速处理大量三维图像

```
ting.m
clear all; close all; clc;
% matlabpool open; % 新版的MATLAB代码改为 : parpool

%% 参数设置
addr = 'F:\Datasets\F0r_ZhouZongwei_3D\F0r_DeepLearning\';
% 数据库存放的地址
save_addr = 'F:\Datasets\F0r_ZhouZongwei_3D\DATA_mat\';
% 图像的尺寸
PIXEL = 101;
CutRange = 51; % patch的大小:CutRange*CutRange
ThetaStep = 6; % 角度划分的步长
ThetaStart = -3; % -10 0 10
angnum = 1; % 用几个角度
DeltaStep = 4; % 平切的步长
DeltaStart = -2; % -2 0 2
delnum = 2; % 用几个平切

%% 参数预处理
CtImaFiles = dir(strcat(addr, '*', '_CT3d.raw'));
% 所有三维CT图像的文件名
PtImaFiles = dir(strcat(addr, '*', '_PT3d.raw'));
% 所有三维PET图像的文件名
CTSampleNum = length(CtImaFiles);
% CT图像样本的个数
```



记录了用MATLAB做旋转，平移的 Data Augmentation 的详细过程

博客链接：<http://www.jianshu.com/p/6fe67ab70ff5>

在处理的过程中涉及到很多MATLAB的代码知识，以及加速的小技巧，不过即使是64GB内存的机器，还是会有MemoryError的问题，初步打算将数据预处理也放到Python端。



CUDA GPU加速



深度学习框架Theano利用GPU加速



详细记录了如何用GPU对Keras的深度学习网络进行加速的全过程。

博客链接: <http://www.jianshu.com/p/14fe34a837a9>

检查一下你的电脑是否有NVIDIA系列的显卡, 没有的话就别往下看了。深度学习的GPU加速只支持NVIDIA系列的显卡。用GPU的理由很简单, 就是快, 特别是深度学习的实验, 用CPU的话一天两天的在那儿训练, 根本没办法愉快地调参, 用GPU的话可能就几个小时完事儿了。



AlexNet专题



我看AlexNet



叙述了我眼中的AlexNet与LeNet的区别, 写的很有态度哦

博客链接: <http://www.jianshu.com/p/58168fec534d>

前几周的Paper Review上刚刚注意到AlexNet, 那果断用啊, 话说LeNet虽然好用, 那也是快20年以前的网络结构了, AlexNet是2012年的结构, 不管怎么说肯定好呀。贴一张网络结构的历史



深度学习 VS 传统分类器

分类器	TPR	SPC	PPV	NPV	ACC	AUC
随机森林	0.817	0.830	0.833	0.825	0.824	0.905
支持向量机	0.854	0.762	0.789	0.841	0.808	0.876
AdaBoost	0.822	0.733	0.760	0.804	0.778	0.846
BP-ANN	0.749	0.786	0.784	0.764	0.767	0.877
2.5D-CNN	0.796	0.801	0.798	0.794	0.785	0.860

恶性肿瘤识别效果与传统分类器相当

- 撰写毕业论文，圆满完成毕业设计；
- 撰写期刊论文，争取发表；
- 训练一个最优的网络，形成可视化的软件；
- 用CNN全面训练出已有不同医学数据集的最优网络；
- 撰写多篇整个实验的过程攻略，交流共享。

后续思路



使用肿瘤的分割标记信息

难度 ★★☆☆☆

加入2.5D分割信息，构成九通道输入的patch，优点是可利用到肿瘤的大小、形状等信息，并且又加大了训练样本数，预计的学习效果会有提高。



差值patch，用AlexNet模型训练

难度 ★★☆☆☆

由于AlexNet的patch输入时227 × 227像素的，我们要将原来的51 × 51的patch差值。AlexNet与LeNet的优化就在与dropout层。

$$Loss' = \frac{Loss}{\beta}$$

$$= \frac{\frac{1}{m} \times \sum_{i=1}^m (h_K(x^{(i)}) - y^{(i)})^2}{\|distance(l_1, l_2)\|}$$

修改Loss函数，量化定义可分类概念

难度 ★★★★★

找到一个量化的指标来定义一组数据“可分类”的程序，用于CNN与RF的衔接阶段，此时CNN的作用就是学习特征，而这个Loss函数用于评价输出特征的好坏。



用多源数据集进一步做fine-tuning

难度 ★★★★★

首先用非特异性的已训练好的网络(CaffeNet, PlaceNet)参数作为AlexNet前部的参数初始化，再应用到不同的医学数据集中，训练得到不同的特异性的CNN。

绪论综述

研究方法

完成情况

成果展示

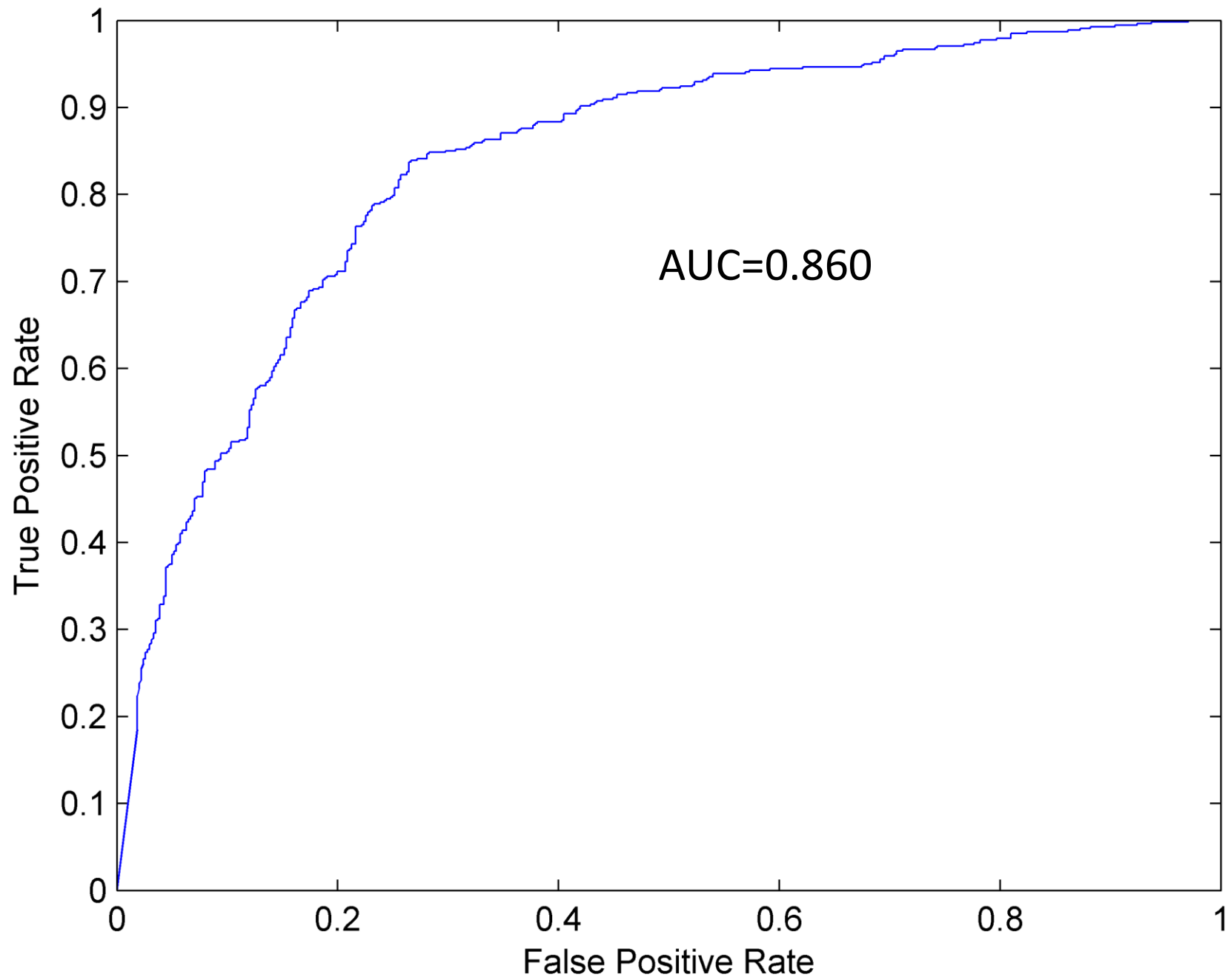


表 A.3 数据拓展到 9.89GB 的 20 次 CNN+SVM/RF 交叉验证结果展示

交叉 验证	CNN+SVM					CNN+RF				
	TPR	SPC	PPV	NPV	ACC	TPR	SPC	PPV	NPV	ACC
No.01	0.81	0.70	0.73	0.79	0.76	0.81	0.70	0.73	0.79	0.76
No.02	0.63	0.85	0.81	0.70	0.74	0.59	0.85	0.80	0.68	0.72
No.03	0.85	0.59	0.68	0.80	0.72	0.85	0.59	0.68	0.80	0.72
No.04	0.78	0.74	0.75	0.77	0.76	0.78	0.74	0.75	0.77	0.76
No.05	0.78	0.85	0.84	0.79	0.81	0.78	0.85	0.84	0.79	0.81
No.06	0.85	0.81	0.82	0.85	0.83	0.85	0.81	0.82	0.85	0.83
No.07	0.70	0.78	0.76	0.72	0.74	0.63	0.78	0.74	0.68	0.70
No.08	0.78	0.70	0.72	0.76	0.74	0.78	0.74	0.75	0.77	0.76
No.09	0.81	0.89	0.88	0.83	0.85	0.81	0.85	0.85	0.82	0.83
No.10	0.93	0.63	0.71	0.89	0.78	0.93	0.63	0.71	0.89	0.78
No.11	0.85	0.81	0.82	0.85	0.83	0.85	0.85	0.85	0.85	0.85
No.12	0.89	0.70	0.75	0.86	0.80	0.89	0.74	0.77	0.87	0.81
No.13	0.78	0.70	0.72	0.76	0.74	0.78	0.70	0.72	0.76	0.74
No.14	0.70	0.85	0.83	0.74	0.78	0.59	0.85	0.80	0.68	0.72
No.15	0.78	0.96	0.95	0.81	0.87	0.81	0.93	0.92	0.83	0.87
No.16	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70
No.17	0.81	0.78	0.79	0.81	0.80	0.81	0.78	0.79	0.81	0.80
No.18	0.81	0.78	0.79	0.81	0.80	0.85	0.78	0.79	0.84	0.81
No.19	0.70	0.70	0.70	0.70	0.70	0.70	0.74	0.73	0.71	0.72
No.20	0.89	0.67	0.73	0.86	0.78	0.89	0.67	0.73	0.86	0.78

注：红色表示比 CNN 高，绿色表示比 CNN 低。



I AM JUST **ON** THE WAY
THANK YOU

致谢

- ▣ 感谢王洪凯老师的付出和指导，以及实验室所有老师、同学的帮助。
- ▣ 感谢大连理工大学本科毕业论文的所有负责老师
- ▣ 感谢**医院提供的PET/CT数据集

周纵苇



专业：计算机科学与技术

学校：大连理工大学

电话：1-480-738-2575

微博：@MrGiovanni

微信：MrGiovanni

邮箱：zongweiz@asu.edu