



2020-04-26

「Udcm:

——CT 影像中肺结节病灶筛查

编译日期: 2020-04-26

任何建议及错误信息请发送至邮箱

junjc9@qq.com



本作品采用知识共享署名-非商业性使用 4.0 国际许可协议进行许可。访问<http://creativecommons.org/licenses/by-nc-nd/4.0/>查看该许可协议。

# 目 录

概 览	vii
第一章 数据预处理	1
1.1 数据集	1
1.1.1 简述	1
1.1.2 DICOM 数据格式	1
1.1.3 注释信息 (.xml)	2
1.2 加载 DICOM 文件，并添加缺失的元数据	4
1.3 灰度值转为 CT 值	5
1.4 重采样	6
1.5 三维可视化	7
1.6 肺分割	8
1.6.1 肺实质切割	8
1.7 数据归一化	9
1.8 数据中心化	10
第二章 深度学习	11
2.1 卷积神经网络	11
2.2 数据处理	11
2.3 模型评估方法	12
2.4 性能度量	13
2.4.1 主要参数指标	13
2.4.2 ROC 曲线	14
2.4.3 AUC 值	14
附录 神经网络结构图	17
参考文献	19



# 插图索引

1-1	提取肺结节标注信息	4
1-2	从左至右依次为横断面、矢状面和冠状面	5
1-3	灰度值转为 HU 单元	7
1-4	多层面重建 (MPR) 后的肺部模型	8
1-5	肺实质分割	9
2-1	传统机器学习的数据处理流程	11
2-2	深度学习的数据处理流程	11
2-3	线性模型的 roc 曲线	14
1-1	Unet[5] 网络结构	17
1-2	tikz spy 局部放大效果	18



# 表格索引

1-1	LIDC 统计信息 . . . . .	1
1-2	常见物质的放射剂量 . . . . .	6
2-1	模型结构 . . . . .	12
2-2	模型结果对比 . . . . .	13





# 概 览

肺癌是全球发病率和死亡率最高的恶性肿瘤, 每年全球约 140 万人因肺癌死亡, 占有恶性肿瘤死亡人数的 18.4%, 远高于其后的结肠直肠癌、胃癌和肝癌等 [1]。早期肺癌症状很容易被患者忽视, 通常很不明显甚至没有症状, 一旦患者出现咳嗽长期不愈并且伴随咳血等症状时, 表明肺癌大多已经出现转移并进入中晚期, 此时病理特征复杂, 恶化速度快, 已经错过了最佳治疗时机, 肺癌预后极差, 我国肺癌的 5 年生存率仅为 15.6%。

在医学上, 早期肺癌主要的表现形式为肺结节, 肺结节 (pulmonary nodule, PN) 是指肺内直径小于或等于 3 cm 的类圆形或不规则形病灶, 影像学表现为密度增高的阴影, 可单发或多发, 边界清晰或不清晰的病灶 [2]。不同密度的肺结节, 其恶性概率不同, 依据结节密度将肺结节分为三类: 实性结节 (solid nodule)、部分实性结节 (part-solid nodule) 和磨玻璃密度结节 (groundglass nodule, GGN)。其中, 部分实性结节的恶性概率最高, 依次为磨玻璃密度结节及实性结节。磨玻璃密度结节是指肺内模糊的结节影, 结节密度较周围肺实质略增加, 但其内血管及支气管的轮廓尚可见。实性结节是指其内全部是软组织密度的结节, 密度较均匀, 其内血管及支气管影像被掩盖。部分实性结节是指其内既包含磨玻璃密度又包含实性软组织密度的结节, 密度不均匀。

临床上, 肺癌诊断方法可分为组织学诊断和影像学诊断两类。其中计算机断层扫描 (Computed tomography, CT) 和正电子发射断层扫描 (Positron emission tomography, PET) 是用于检测和诊断肺结节的常见医学影像手段。随着 CT 成像分辨率越来越高, 能够被检测的肺结节体积越来越小, 但我国医学影像领域专业医生缺口巨大, 且平均每位患者的 CT 影像为 200 至 300 张, 普通三甲医院放射科医生日均阅片量为 4 至 6 万张, 沉重的工作负担让医生疲劳, 误诊漏诊率上升, 而且读片并进行靶区勾画耗时较长, 知识容易遗忘, 造成效率低下。

近年来, 深度学习方法在图像处理与目标检测领域取得了极大成功, 以深度学习为基础的计算机辅助诊断 (Computed Aided Detection, CAD) 系统在医学影像的识别与检测方面不但达到了较高的准确率, 还可以减少医师工作量, 提高就诊效率, 有重要的研究意义和应用价值。





# 1

## 数据预处理

### 1.1 数据集

#### 1.1.1 简述

数据集采用为 LIDC-IDRI(The Lung Image Database Consortium), 该数据集由胸部医学图像文件 (.dcm) 和对应的诊断结果病变标注 (.xml) 组成。数据是由美国国家癌症研究所 (National Cancer Institute) 发起收集的, 目的是为了研究高危人群早期癌症检测。该数据集中, 共收录了 1018 个研究实例。对于每个实例中的图像, 都由 4 位经验丰富的胸部放射科医师进行两阶段的诊断标注。在第一阶段, 每位医师分别独立诊断并标注病患位置, 其中会标注三中类别:

- ▶ 大于等于 3mm 的结节
- ▶ 小于 3mm 的结节
- ▶ 大于等于 3mm 的非结节

在随后的第二阶段中, 各位医师都分别独立的复审其他三位医师的标注, 并给出自己最终的诊断结果。这样的两阶段标注可以在避免 forced consensus 的前提下, 尽可能完整的标注所有结果。

表 1-1 LIDC 统计信息<sup>1</sup>

Collection Statistics	updated 3/21/2012
数据大小	124GB
图像类型	CT (computed tomography), 243,958 张
图片数	244,527
患者数	1,010
系列数 (Number of Series)	1,018 CT 290 CR/DX
研究数 (Number of Studies)	1,308

#### 1.1.2 DICOM 数据格式

图像文件为 Dicom 格式, 是医疗图像的标准格式, 其中除了图像像素外, 还有一些辅助的元数据如图像类型、图像时间等信息。

<sup>0</sup>参见 <https://wiki.cancerimagingarchive.net/display/Public/LIDC-IDRI>

一张 CT 图像有 512x512 个像素点，在 dicom 文件中每个像素由 2 字节表示，所以每张图片约 512KB 大小。目前测试一共 1012 个病例数据，对于每个实例，可以看作一个三维矩阵  $D: slicer * rows * cols$ ， $D$  中第  $z$  个切片  $y$  行  $x$  列的元素对应的位置为： $(z * rows * cols + y * cols + x) * sizeof(data\_type)$ 。其中  $slicer$  表示切片的个数 (对应每个病例的.dcm 文件数)， $rows$  表示图像的行数， $cols$  表示图像的列数，默认均为 512； $data\_type$  代表数据类型，默认为 short。

eg: 对于病例 LIDC-IDRI-0001，即为  $133 \times 512 \times 512$  的矩阵，一共 133 张切片，每张大小  $512 \times 512$ ，依次按顺序存入二进制文件，每个像素大小为 2 字节（对应 short 类型）。

Listing 1-1 切片的元数据

```
>>> import pydicom
>>> dcm=pydicom.dcmread("000087.dcm")
>>> print(dcm)
(0008,0060)Modality CS: 'CT'
(0008,0070)Manufacturer LO: 'TOSHIBA'
(0010,0020)Patient ID LO: 'LIDC-IDRI-0078'
(0018,0050)Slice Thickness DS: "3.0"
(0020,0020)Patient Orientation CS: ['L', 'P']
(0020,0032)Image Position(Patient) DS:
    [-165.625000,-164.843700, 1541.500000]
(0020,1041)Slice Location DS: "179.0"
(0028,0010)Rows US: 512
(0028,0011)Columns US: 512
(0028,0030)Pixel Spacing DS: [0.650, 0.650]
(0028,0103)Pixel Representation US: 1
(0028,0106)Smallest Image Pixel Value SS: -2048
(0028,0107)Largest Image Pixel Value SS: 1452
(0028,1052)Rescale Intercept DS: "0.0"
(0028,1053)Rescale Slope DS: "1.0"
(7fe0,0010)Pixel Data OW: Array of 524288
    elements
```

### 1.1.3 注释信息 (.xml)

Xml 文件中包含放射科医生对病人 CT 图像中疑似肺结节的标注信息，主要分为三类：

1. 结节 (3mm ~ 33mm): 包含结节的特征信息 (characteristics)、结节的完整轮廓 (roi)
2. 结节 (<3mm): 只显示结节的近似三维重心，若不透明则不标记

## 3. 非结节 (&gt;3mm): 只显示其近似的三维重心, 指出非结节连接区域

Listing 1-2 放射医师的结节标注文件节选

```

1  <readingSession> %专家标注的结节
2  <annotationVersion>3.12</annotationVersion>
3  <servicingRadiologistID>anonymous</servicingRadiologistID>
4  <unblindedReadNodule>
5  <noduleID>3</noduleID> %结节编号
6    <characteristics>
7      <subtlety>5</subtlety> %精细度
8      <internalStructure>1</internalStructure> %内部结构
9      <calcification>6</calcification> %钙化度
10     <sphericity>3</sphericity> %球形度
11     <margin>4</margin> %边缘特征
12     <lobulation>1</lobulation> %分叶征
13     <spiculation>1</spiculation> %毛刺征
14     <texture>5</texture> %纹理特征
15     <malignancy>3</malignancy> %恶性度
16   </characteristics>
17   <roi>
18     <imageZposition>1550.5</imageZposition>
19     <imageSOP_UID>
20       >1.3.6.1.4.1.14519.5.2.1.6279.6001.498089111228175009979138545584
21     </imageSOP_UID> %CT图像号
22     <inclusion>TRUE</inclusion>
23     <edgeMap> %结节边缘坐标
24       <xCoord>364</xCoord>
25       <yCoord>172</yCoord>

```

其中对于 3mm ~ 33mm 结节的 characteristics, 包含了如下信息:

1. Subtlety: 检测难度 (1-5 级, 1 最难, 5 最明显)
2. internalStructure: 内部结构 (4 种, 软组织、液体、脂肪、空气)
3. calcification: 钙化 (6 种情况)
4. sphericity: 球形度 (5 种程度, 但只明确 3 种)
5. margin: 边缘 (5 种程度)
6. lobulation: 分叶征 (5 种情况, 但只明确 2 种)
7. spiculation: 毛刺征 (5 种情况, 但只明确 2 种)
8. texture: 纹理 (5 种情况, 但只明确 3 种)
9. maliynancy: 恶性程度 (1-5, 1 最低, 5 最高)

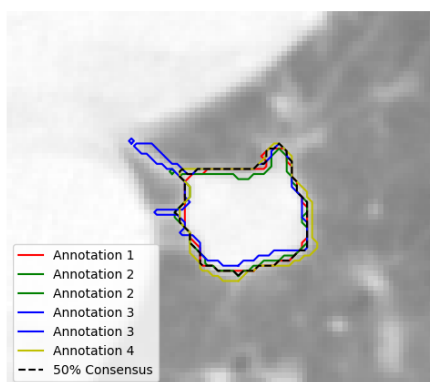
根据 xml 文件内不同医师在 roi 标签中标注的 x,y 坐标可以提取结节轮廓并多平面重建显示其三维结构特征, 详见下图1-1。

**Algorithm 1:** 提取标注信息**Data:** xml 文件路径, dcm 文件路径**Result:** 结节轮廓坐标

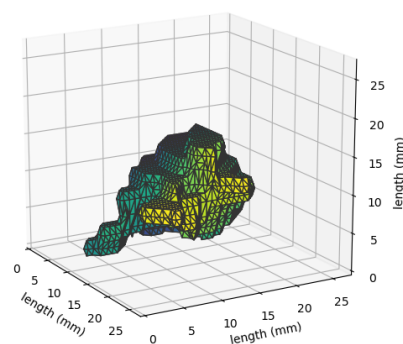
```

1 引入 xml.etree 中的 ElementTree;
2 获取根节点;
3 遍历第一层 ResponseHeader 节点, Return 序列和检查实例标识号;
4 for 遍历第二层 readingSession 节点 do
5     Return 服务放射专家编号, 结节与非结节信息;
6     if 遍历第三层 unblindedReadNodule 节点 then
7         Return 结节编号和特征;
8         for 遍历第四层 roi 节点 do
9             Return 图像层坐标、SOP 标识号、判断标识;
10            for 遍历第五层 edgeMap 节点 do
11                Return x、y 坐标;
12            end
13        end
14    else
15        遍历第三层 nonNodule 节点;
16        Return 非结节编号、图像层坐标和 SOP 标识号;
17        for 遍历第五层 locus 节点 do
18            Return x、y 坐标;
19        end
20    end
21 end

```



(a) 不同放射医师标注的结节



(b) 结节三维结构

图 1-1 提取肺结节标注信息

## 1.2 加载 DICOM 文件, 并添加缺失的元数据

加载必要的 python 模块递归读取存放 CT 切片的文件夹, 配置数据集路径及执行 load\_scan 函数后会得到如1-2图所示结果。

Listing 1-3 使用 pydicom 读取的 dcm 切片

```
1 import pydicom
2 import os
3 import numpy as np
4 import matplotlib.pyplot as plt
5 import cv2
6 from skimage import measure, morphology
7 from mpl_toolkits.mplot3d.art3d import Poly3DCollection
8 import scipy.ndimage as ndi
9 def load_scan(path):
10     slices = [pydicom.dcmread(path)]
11     slices.sort(key = lambda x: float(x.ImagePositionPatient[2]))
12     try:
13         thickness = np.abs(slices[0].ImagePositionPatient[2] - slices[1].
14                             ImagePositionPatient[2])
15     except:
16         thickness = np.abs(slices[0].SliceLocation - slices[1].SliceLocation
17                             )
18     for s in slices:
19         s.SliceThickness = thickness
20     return slices
```



图 1-2 从左至右依次为横断面、矢状面和冠状面

### 1.3 灰度值转为 CT 值

CT 值的单位是 Hounsfield，简称为 Hu，范围是-1024-3071。用于衡量人体组织对 X 射线的吸收率，设定水的吸收率为 0Hu。详见下表 1-2。

在 DICOM 图像读取的过程中，我们会发现图像的像素值有可能不是这个范围，通常是 0-4096，这是我们常见到的像素值或者灰度值，这就需要在图像像素值（灰度值）转换为 CT 值。

<sup>1</sup>参见 [Wikipedia](#)



表 1-2 常见物质的放射剂量<sup>1</sup>

substance 物质	HU 值
Air 空气	-1000
Lung 肺	-500
Fat 脂肪	-100 到 -50
Water 水	0
CSF 脑脊液	15
Kidney 肾	30
Blood 血液	+30 到 +45
Muscle 肌肉	+10 到 +40
Grey matter 灰质	+37 到 +45
White matter 白质	+20 到 +30
Liver 肝	+40 到 +60
Soft Tissue 软组织	+100 到 +300
Bone 骨头	+700(软质骨) 到 +3000(皮质骨)

首先，需要读取两个 DICOM 的 Tag 信息，(0x0028,0x1052): rescale intercept 和 (0x0028,0x1053): rescale slope. 然后通过公式：

$$Hu = pixel \times slope + intercept \quad (1.1)$$

计算得到 CT 值。

Listing 1-4 将灰度像素值转换为对应组织的 Hounsfield 单元 (HU)

```

1 def get_pixels_hu(slices):
2     img = np.stack([s.pixel_array for s in slices])
3     img = img.astype(np.int16)
4     img[img == -2000] = 0
5     for i in range(len(slices)):
6         intercept = slices[i].RescaleIntercept
7         slope = slices[i].RescaleSlope
8         if slope != 1:
9             img[i] = slope * img[i].astype(np.float64)
10            img[i] = img[i].astype(np.int16)
11            img[i] += np.int16(intercept)
12     return np.array(img, dtype=np.int16)

```

## 1.4 重采样

CT 扫描可能的像素间距为 [2.5, 0.5, 0.5]，代表着切片间的距离是 2.5 毫米。Tag 对应为 (0x0018,0x0050): Slice Thickness，对于不同的扫描，切片距离可能不同，这对自动

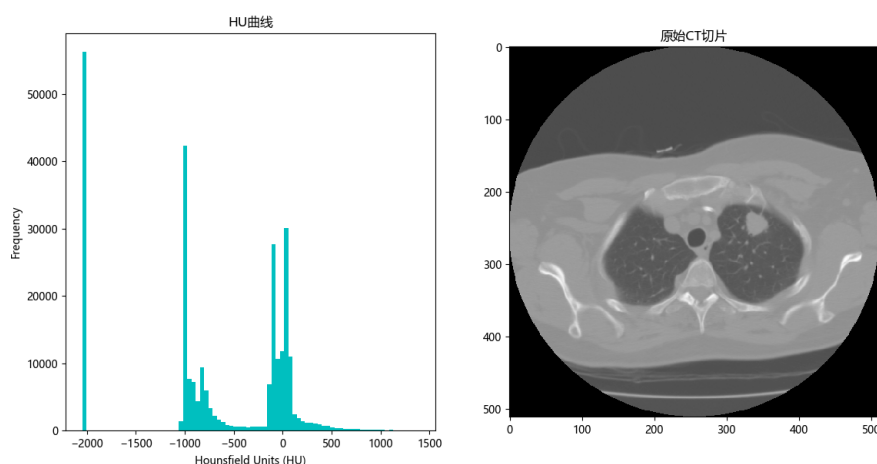


图 1-3 灰度值转为 HU 单元

分析造成一定的影响。

常用的方法是整个数据集重新采样为相同分辨率的切片。例如将所有切片采样为 [1, 1, 1] 毫米的间距，这样就可以使用 3D 网格，随机的切片厚度的也不影响批量化自动分析。

Listing 1-5 重新采样至相同分辨率

```
1 def resample(img, scan, new_spacing=[1,1,1]):
2     spacing = np.array([scan[0].SliceThickness] + scan[0].PixelSpacing,
3                       dtype=np.float32)
4     resize_factor = spacing / new_spacing
5     new_real_shape = img.shape * resize_factor
6     new_shape = np.round(new_real_shape)
7     real_resize_factor = new_shape / img.shape
8     new_spacing = spacing / real_resize_factor
9     img = ndi.interpolation.zoom(img, real_resize_factor, mode='nearest')
10    return img, new_spacing
```

## 1.5 三维可视化

常用 CT 切片三维重建方式为多层面重建 (MPR)，该方法方便快捷，计算量较小，而且能让我们直观的观察肺部的立体结构，见下图 1-4。

```
1 def plot_3d(img, threshold=-300):
2     p = img.transpose(2,1,0)
3     verts, faces = measure.marching_cubes(p, threshold)
4     fig = plt.figure(figsize=(10, 10))
5     ax = fig.add_subplot(111, projection='3d')
6     mesh = Poly3DCollection(verts[faces], alpha=0.70)
7     face_color = [0.45, 0.45, 0.75]
```

```

8     mesh.set_facecolor(face_color)
9     ax.add_collection3d(mesh)
10    ax.set_xlim(0, p.shape[0])
11    ax.set_ylim(0, p.shape[1])
12    ax.set_zlim(0, p.shape[2])
13    plt.show()

```

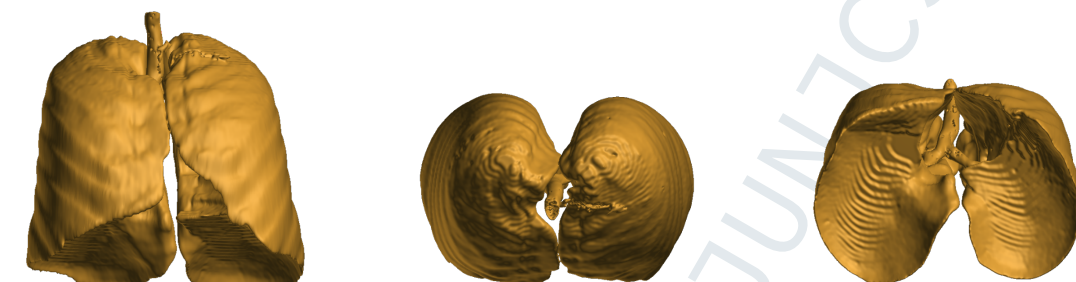


图 1-4 多层面重建 (MPR) 后的肺部模型

## 1.6 肺分割

### 1.6.1 肺实质切割

为了降低问题的复杂度，我们先对肺部进行切割，主要使用图像处理中的区域生长法和形态学处理。其中连通分量分析（Connected Component Analysis）是最重要的步骤，它一种在 CVPR 和图像分析处理的众多应用领域中较为常用和基本的方法，指将图像中的各个连通区域找出并标记。

- ▶ 图像阈值化
- ▶ 建立连通区域，确定相邻像素的标签，在二值图像中计数为 1
- ▶ 对于每个轴向切片，确定最大的固体连通区域，将其他区域置 0。
- ▶ 仅保留最大的气袋

```

1  def largest_label_volume(im, bg=-1):
2      vals, counts = np.unique(im, return_counts=True)
3      counts = counts[vals != bg]
4      vals = vals[vals != bg]
5      if len(counts) > 0:
6          return vals[np.argmax(counts)]
7      else:
8          return None
9
10 def segment_lung_mask(img, fill_lung_structures=True):
11     binary_img = np.array(img > -320, dtype=np.int8)+1
12     labels = measure.label(binary_img)
13     background_label = labels[0,0,0]

```

```

14     binary_img[background_label == labels] = 2
15     if fill_lung_structures:
16         for i, axial_slice in enumerate(binary_img):
17             axial_slice = axial_slice - 1
18             labeling = measure.label(axial_slice)
19             l_max = largest_label_volume(labeling, bg=0)
20             if l_max is not None:
21                 binary_img[i][labeling != l_max] = 1
22     binary_img -= 1
23     binary_img = 1-binary_img
24     labels = measure.label(binary_img, background=0)
25     l_max = largest_label_volume(labels, bg=0)
26     if l_max is not None:
27         binary_img[labels != l_max] = 0
28     return binary_img

```

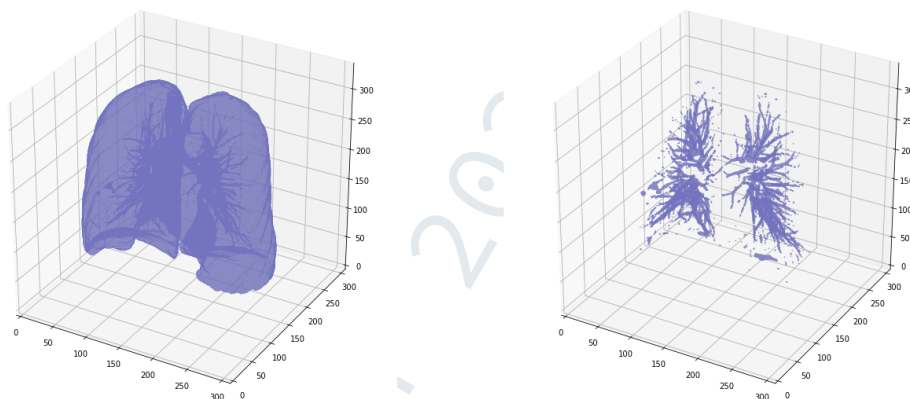



图 1-5 肺实质分割

## 1.7 数据归一化

归一化实质是一种线性变换，即将有量纲的表达式，经过变换，化为无量纲的表达式，成为纯量。而且线性变换不会改变原始数据的数值排序，反而能提高数据的表现。

通过遍历切片发现 CT 值分布在 -1024 到 2000Hu 这个区间，由上表 1-2 可知，超过 400Hu 的是不同辐射密度的骨骼，所以我们将阈值区间定在在 -1000 到 400Hu 之间。

 **注 1.1.** 通过将数据归一化，在下阶段训练神经网络的过程中，能够加速权重参数的收敛速度


Min-Max Normalization:

$$x^* = \frac{x - X_{\min}}{X_{\max} - X_{\min}} \quad (1.2)$$

```
1 MIN_BOUND = -1000.0
2 MAX_BOUND = 400.0
3
4 def normalize(img):
5     img = (img - MIN_BOUND) / (MAX_BOUND - MIN_BOUND)
6     img[img>1] = 1.
7     img[img<0] = 0.
8     return img
```

## 1.8 数据中心化

作为预处理阶段最后的步骤，中心化是指原数据减去它的平均值，经过中心化处理后，原数据的坐标平移至中心点 (0,0)，该组数据的均值变为 0，以此也被称为零均值化。

 **注 1.2.** 对数据进行中心化预处理，可以增加基向量的正交性。

zero centering:

$$x^* = x - \mu \quad (1.3)$$

```
1 PIXEL_MEAN = 0.25
2
3 def zero_center(img):
4     img = img - PIXEL_MEAN
5     return img
```

# 2 深度学习

## 2.1 卷积神经网络

卷积神经网络 [2] 的特征检测层通过训练数据进行学习，所以在使用卷积神经网络时，避免了显示的特征抽取，而隐式地从训练数据中进行学习；由于同一特征映射面上的神经元权值相同，所以网络可以并行学习，这也是卷积网络相对于神经元彼此相连网络的一大优势。卷积神经网络以其局部权值共享的特殊结构在语音识别和图像处理方面有着独特的优越性。权值共享降低了网络的复杂性，特别是多维输入向量的图像可以直接输入网络这一特点避免了特征提取和分类过程中数据重建的复杂度。

卷积神经网络（CNN）主要由卷积层、激活函数、池化层、全连接层组成。

- 卷积层（Conv）：使用卷积核进行特征提取和特征映射
- 激活函数（Activation）：由于卷积也是一种线性运算，因此需要增加非线性映射
- 池化层（Pool）：对输入的特征图进行压缩，一方面使特征图变小，简化网络计算复杂度；一方面进行特征压缩，提取主要特征
- 全连接层（FC）：连接所有的特征，将输出值送给分类器

## 2.2 数据处理



图 2-1 传统机器学习的数据处理流程

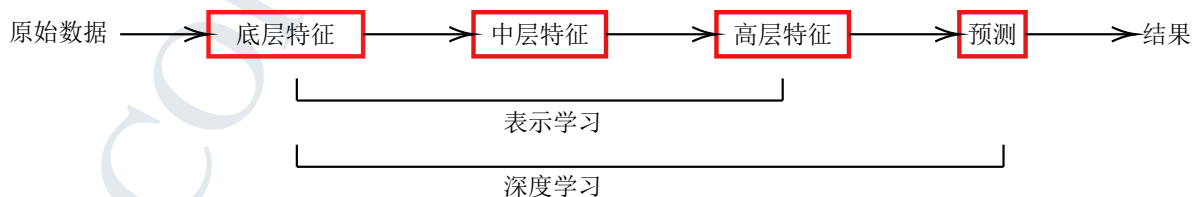


图 2-2 深度学习的数据处理流程

## 2.3 模型评估方法

交叉验证 (Cross validation), 交叉验证用于防止模型过于复杂而引起的过拟合. 有时亦称循环估计, 是一种统计学上将数据样本切割成较小子集的实用方法. 于是可以先在一个子集上做分析, 而其它子集则用来做后续对此分析的确认及验证. 一开始的子集被称为训练集. 而其它的子集则被称为验证集或测试集. 交叉验证是一种评估统计分析、机器学习算法对独立于训练数据的数据集的泛化能力 (generalize)。

交叉验证一般要尽量满足:

- ▶ 训练集的比例要足够多, 一般大于一半
- ▶ 训练集和测试集要均匀抽样

交叉验证主要分成以下几类:

- ▶ **k-fold cross-validation**: k 个子集, 每个子集均做一次测试集, 其余的作为训练集. 交叉验证重复 k 次, 每次选择一个子集作为测试集, 并将 k 次的平均交叉验证识别正确率作为结果. 优点: 所有的样本都被作为训练集和测试集, 每个样本都被验证一次. 10-fold 通常被使用。
- ▶ **K \* 2 folder cross-validation**: 是 k-fold cross-validation 的一个变体, 对每一 folder, 都平均分成两个集合 s0,s1, 我们先在集合 s0 训练用 s1 测试, 然后用 s1 训练 s0 测试. 优点是: 测试和训练集都足够大, 每一个个样本都被作为训练集和测试集. 一般使用 k=10。
- ▶ **least-one-out cross-validation(loocv)**: 假设 dataset 中有 n 个样本, 那 LOOCV 也就是 n-CV, 意思是每个样本单独作为一次测试集, 剩余 n-1 个样本则做为训练集. 优点: 1) 每一回合中几乎所有的样本皆用于训练模型, 因此最接近母体样本的分布, 估测所得的泛化误差比较可靠. 2) 实验过程中没有随机因素会影响实验数据, 确保实验过程是可以被复制的. 但 LOOCV 的缺点则是计算成本高, 为需要建立的模型数量与总样本数量相同, 当总样本数量相当多时, LOOCV 在实作上便有困难, 除非每次训练模型的速度很快, 或是可以用平行化计算减少计算所需的时间。

表 2-1 模型结构

Type	Input	Kernel	Output
Input	$28 \times 28 \times 1$	$5 \times 5$	$24 \times 24 \times 32$
Loss	$24 \times 24 \times 32$	$2 \times 2$	$12 \times 12 \times 64$
Fully connected	$12 \times 12 \times 64$	$5 \times 5$	$8 \times 8 \times 64$
Softmax	$8 \times 8 \times 64$	$4 \times 4$	$4 \times 4 \times 64$
Convolution	$4 \times 4 \times 64$	$1 \times 1$	$512 \times 1$
Max pooling	$512 \times 1$	$1 \times 1$	$2 \times 1$



## 2.4 性能度量

在分类任务中，人们总是喜欢基于错误率来衡量分类器任务的成功程度。错误率指的是在所有测试样例中错分的样例比例。实际上，这样的度量错误掩盖了样例如何被分错的事实。在机器学习中，有一个普遍适用的称为混淆矩阵 [7](confusion matrix) 的工具，它可以帮助人们更好地了解分类中的错误。

针对一个二分类问题，我们将实例分成正类 (positive) 和负类 (negative) 两种。在肺结节计算机辅助识别这一问题上，一幅肺部 CT 图像中有肺结节被认为是阳性 (positive)，没有肺结节被认为是阴性 (negative)。所以在实际检测时，就会有如下四种情况：

- ▶ 真阳性 (True Positive, TP): 检测有结节，且实际有结节；正确肯定的匹配数目；
- ▶ 假阳性 (False Positive, FP): 检测有结节，但实际无结节；误报，给出的匹配是不正确的；
- ▶ 真阴性 (True Negative, TN): 检测无结节，且实际无结节；正确拒绝的非匹配数目；
- ▶ 假阴性 (False Negative, FN): 检测无结节，但实际有结节；漏报，没有正确找到的匹配的数目。

### 2.4.1 主要参数指标

正确率 (Precision):

$$Precision = \frac{TP}{TP + FP} \quad (2.1)$$

真阳性率 (True Positive Rate, TPR)，灵敏度 (Sensitivity)，召回率 (Recall):

$$Sensitivity = Recall = TPR = \frac{TP}{TP + FN} \quad (2.2)$$

真阴性率 (True Negative Rate, TNR)，特异度 (Specificity):

$$Specificity = TNR = \frac{TN}{TP + FN} \quad (2.3)$$

表 2-2 模型结果对比

Models	Accuracy	Sensitivity	Specificity
CNN	84.15%	84.15%	84.15%
LP-3DCNN[3]	86.75%	84.15%	84.15%
Faster-RCNN[4]	85.23%	84.15%	84.15%
Mask-RCNN[1]	94.56%	84.15%	84.15%
Inception_ResNet_V4[6]	88.21%	84.15%	84.15%



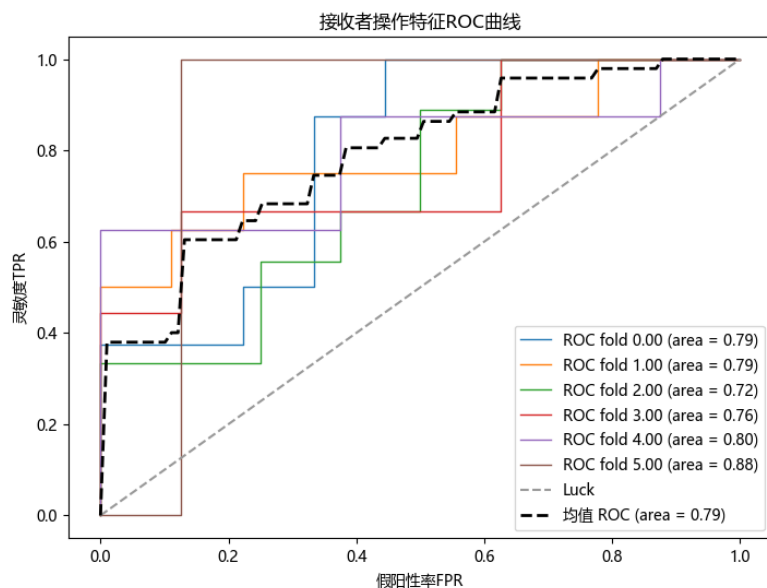


图 2-3 线性模型的 roc 曲线

假阴性率 (False Negative Rate, FNR), 漏诊率 ( $= 1 - \text{灵敏度}$ ):

$$FNR = \frac{FN}{TP + FN} \quad (2.4)$$

假阳性率 (False Positive Rate, FPR), 误诊率 ( $= 1 - \text{特异度}$ ):

$$FPR = \frac{FP}{FP + TN} \quad (2.5)$$

F1 值是精确率和召回率的调和均值, 即  $F1 = 2PR/(P+R)$ , 相当于精确率和召回率的综合评价指标。

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \quad (2.6)$$

## 2.4.2 ROC 曲线

ROC 曲线: 接收者操作特征曲线 (receiver operating characteristic curve), 是反映敏感性和特异性连续变量的综合指标, roc 曲线上每个点反映着对同一信号刺激的感受性。下图是一个 ROC 曲线的示例:

## 2.4.3 AUC 值

AUC 值等效于随机选择的阳性示例的排名高于随机选择的阴性示例的概率。

从 AUC 判断分类器 (预测模型) 优劣的标准:

- AUC = 1, 是完美分类器, 采用这个预测模型时, 存在至少一个阈值能得出完美预测。绝大多数预测的场合, 不存在完美分类器。

- ▶  $0.5 < \text{AUC} < 1$ ，优于随机猜测。这个分类器（模型）妥善设定阈值的话，能有预测价值。
  - ▶  $\text{AUC} = 0.5$ ，跟随机猜测一样（例：丢铜板），模型没有预测价值。
  - ▶  $\text{AUC} < 0.5$ ，比随机猜测还差；但只要总是反预测而行，就优于随机猜测。
- 简单说：AUC 值越大的分类器，正确率越高。



# A

## 神经网络结构图

图 1-1 是一张卷积神经网络示意。使用 tikz 环境, 搭配四种预定义节点 (startstop、process、decision 和 io), 可以容易地绘制出流程图。

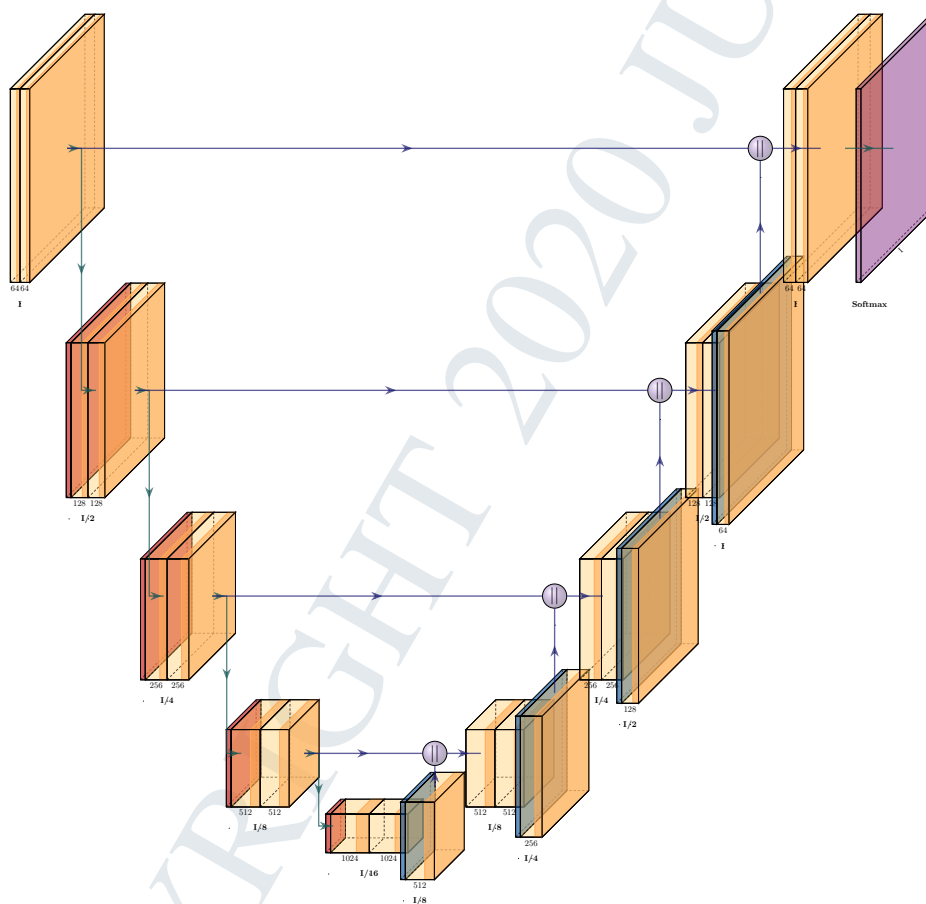


图 1-1 Unet<sup>[5]</sup> 网络结构

<sup>1</sup>Unet 因网络形状似 U 形故而得名

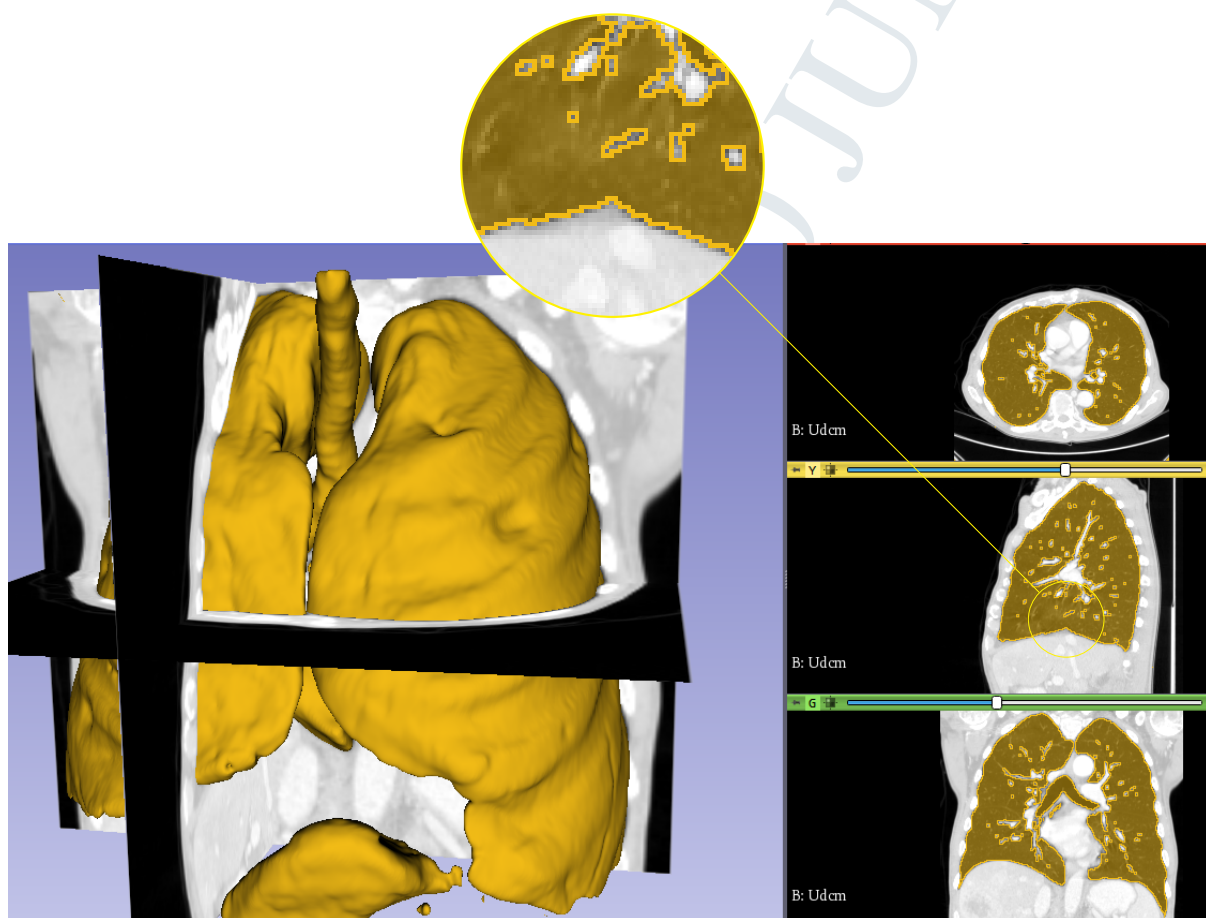


图 1-2 tikz spy 局部放大效果

## 参考文献

- [1] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. “Mask R-CNN”. In: *2017 IEEE International Conference on Computer Vision (ICCV)* (Oct. 2017). DOI: [10.1109/iccv.2017.322](https://doi.org/10.1109/iccv.2017.322). URL: <http://dx.doi.org/10.1109/ICCV.2017.322>.
- [2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016). DOI: [10.1109/cvpr.2016.90](https://doi.org/10.1109/cvpr.2016.90). URL: <http://dx.doi.org/10.1109/CVPR.2016.90>.
- [3] Sudhakar Kumawat and Shanmuganathan Raman. *LP-3DCNN: Unveiling Local Phase in 3D Convolutional Neural Networks*. 2019. arXiv: [1904.03498](https://arxiv.org/abs/1904.03498) [cs.CV].
- [4] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. 2015. arXiv: [1506.01497](https://arxiv.org/abs/1506.01497) [cs.CV].
- [5] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015* (2015), pp. 234–241. ISSN: 1611-3349. DOI: [10.1007/978-3-319-24574-4\\_28](https://doi.org/10.1007/978-3-319-24574-4_28). URL: [http://dx.doi.org/10.1007/978-3-319-24574-4\\_28](http://dx.doi.org/10.1007/978-3-319-24574-4_28).
- [6] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning*. 2016. arXiv: [1602.07261](https://arxiv.org/abs/1602.07261) [cs.CV].
- [7] Wentao Zhu, Chaochun Liu, Wei Fan, and Xiaohui Xie. *DeepLung: Deep 3D Dual Path Nets for Automated Pulmonary Nodule Detection and Classification*. 2018. arXiv: [1801.09555](https://arxiv.org/abs/1801.09555) [cs.CV].