

图像语义分割 (Image Semantic Segmentation) 综述

计算机工程与技术学院

周方全 | 202121081229

邮件地址: 2542154447@qq.com

Abstract

语义分割 (Semantic Segmentation) 是计算机视觉中十分重要的领域, 它是指像素级别的识别图像, 即标注出每个像素所属的对象类别。此项技术目前广泛应用于医学图像与无人驾驶等领域。本文主要从语义分割的基本概念介绍在深度学习引入前后此领域的算法发明与改进, 侧重点在深度算法, 从原始算法全卷积网络 (FCN) 为切入点, 引入一些其改进算法包括: Encoder-Decoder 结构的 U-net, 具有更大感受野的空洞卷积 (Dilated Convolution) 以及加入条件随机场 (CRF)。

Key words- 语义分割; 归一化割; 全卷积网络; 空洞卷积;

I. 归约

我们把问题分成两类, 可以在多项式时间中求出结果的和不可以多项式时间中求出结果的。

假如我们知道了一个问题 X 可以在多项式时间内解决。那我们还能用问题 X 的算法来解决其他问题吗?

这种用一种问题的算法去解决其他问题的扩展方式叫做归约。下面我们给出归约的定义: 如果对于问题 X 任意的一个实例都能用以下两步解决:

- i. 多项式次数的标准计算加上,
- ii. 多项式次数调用可以解决问题 Y 的黑箱。

那么我们就称问题 X 可以在多项式时间内归约到问题 Y 。记为: $X \leq_P Y$ 。

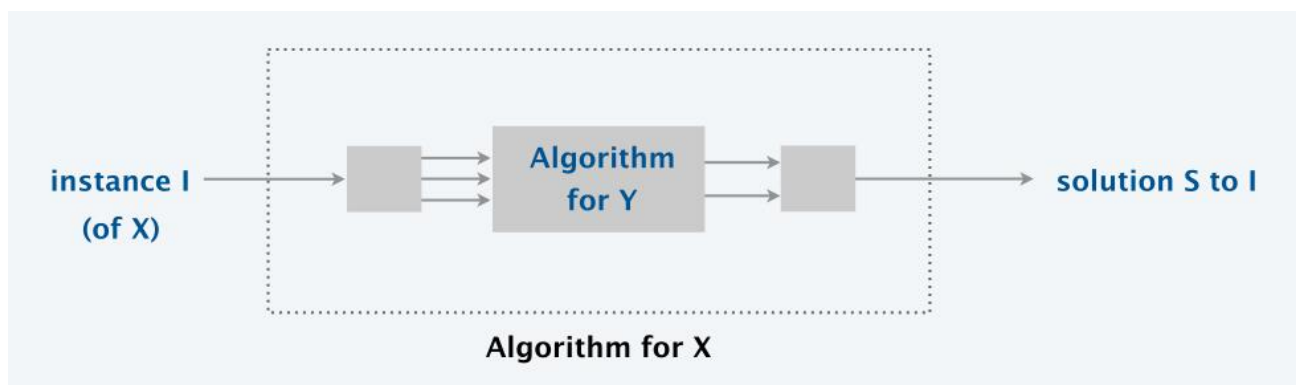


图 1: 归约定义示意图

我们可以用图 1 任一给出一个问题 X 的一个实例我们花费多项式时间把它写入黑箱 (姑且称为问题 Y 的算法, 之所以成为黑箱, 是因为问题 Y 可能也没有很好的已有算法), 所以问题 Y 的实力的大小一定是

多项式的。那么问题 X 就可以得到解决。我们把这种归约称为 Cook 归约。显然归约具有递归性，即如果 $X \leq_P Y$ 并且 $Y \leq_P Z$ 那么， $X \leq_P Z$ 。

我们定义归约，是为了把问题按照求解的难度进行分类。因为简单的问题可以归约到复杂的问题上去，而难的问题不可以归约到简单的问题上来。因为一个用来解决简单的问题的算法在碰到更加难的问题时就会行不通。

用归约设计算法。如果 $X \leq_P Y$ 并且问题 Y 可以在多项式时间中得到解决，那么问题 X 也可以在多项式时间内得到解决。问题 X 可以归约到问题 Y ，那么本质上问题 X 不会比问题 Y 更加难，那么适用于问题 Y 的算法也一定在经过多项式时间处理后适用于问题 X 。

用归约来证明问题是不可解。如果 $X \leq_P Y$ 并且问题 X 不能在多项式时间中得到解决，那么问题 Y 也一定不能在多项式时间内得到解决。

用归约来证明两个问题多项式可解的等价性。如果 $X \leq_P Y$ 并且 $Y \leq_P X$ ，那么我们会得到 $X \equiv_P Y$ 。只要这两个问题一个得到解决，那么另外一个问题就能得到解决。

对不同问题之间建立归约，往往需要很强的技巧性。3 种常用的归约方法：

i. 简单的问题等价性的归约

ii. 从特殊情形到一般情形的归约

iii. 使用小装置来完成归约

后面在实例中我们会一一介绍。

独立集问题的介绍、点覆盖问题的介绍、集合覆盖问题的介绍、3-SAT 问题的介绍

简单的问题等价性的归约的实例

证明：独立集问题 \equiv_P 点覆盖问题

从特殊情形到一般情形的归约的实例

证明：点覆盖问题 \leq_P 集合覆盖问题

使用小装置来完成归约的实例

证明：3-SAT 问题 \leq_P 独立集问题

II. \mathcal{NP}

III.

IV.

Algorithm 1: 找到大小为 K 的团

Input: 一张图 $G(V, E)$, 团的大小 k

Output: 团大小为 k 的顶点集合

```
1 if  $fun(G, k) \neq True$  then
2   | 不存在这样的团;
3   | return None;
4 for 顶点  $v$  in  $V$  do
5   | // 删除顶点  $v$  和 其的相邻边
6   |  $V' = V - \{v\}$ ;
7   |  $E' = E - \{(u, v) : u \in V\}$ ;
8   | if  $fun(G, k) == True$  then
9   |   |  $V = V'$ ;
10  |   |  $E = E'$ ;
11  |   if  $|V| == k$  then
12  |     | break;
13 return  $V$ ;
```

V. \mathcal{NPC}

VI.