语义分割学习笔记

导航

1 前言 **2**

2 准备知识 **2**

2.1 注意力机制 . . . 2

2.2 自注意力机制 . . . 3

2.2.1 为什么使用 LN 而不是 BN，dropout 与 LN 同时使用时，有什么特殊操作? . . . 3

2.2.2 为什么要基于学习的词向量空间? . . . 3

2.2.3 可学习的位置编码? . . . 3

2.2.4 teacher forcing 原理 . . . 3

3 从语义分割的几个桎梏说起 **3**

3.1 CNN 的局限性 . . . 4

3.1.1 UNet . . . 4

3.1.2 PSPNet . . . 4

3.1.3 Deeplab 系列 . . . 4

3.1.4 HRNet . . . 4

3.2 标注数据的局限性 . . . 4

3.3 模型的实时性 . . . 5

3.3.1 DDRNet[4] . . . 5

3.4 模型的泛化性 . . . 5

4 Transformer，它来了 **5**

4.1 开山之作 VIT . . . 5

4.1.1 简介 . . . 5

4.1.2 ViT 的结构 . . . 6

5 附录 **7**

5.1 半监督学习 . . . 7

5.1.1 Mean Teacher . . . 8

1 前言

以下是在学习基于深度学习方法实现语义分割的过程中，总结的论文、学习报告、博客等等，以及针对部分问题提出的个人看法，特此记录！

2 准备知识

2.1 注意力机制

简单的说，注意力机制描述了序列元素的加权平均值，其权重是根据输入的 query 和元素的 key 进行动态计算的。具体地，在注意力机制中，有 4 个概念需要明确

- Query：Query（查询）是一个特征向量，描述我们在序列中寻找什么，即我们可能想要注意什么
- Keys：每个输入元素有一个键，它也是一个特征向量。该特征向量粗略地描述了该元素“提供”什么，或者它何时可能很重要。键的设计应该使得我们可以根据 Query 来识别我们想要关注的元素
- Values：每个输入元素，我们还有一个值向量。这个向量就是我们想要平均的向量
- Score function：评分函数，为了对想要关注的元素进行评分，需要指定一个评分函数 f 该函数将查询和键作为输入，并输出查询-键对的得分/注意力权重。它通常通过简单的相似性度量来实现，例如点积或 MLP。

由此，权重通过 softmax 函数计算：

$$\alpha_i = \frac{\exp\left(f_{attn}\left(Key_i, Query\right)\right)}{\sum_j \exp\left(f_{attn}\left(key_j, query\right)\right)}, out = \sum_i \alpha_i value_i \tag{1}$$

2.2 自注意力机制

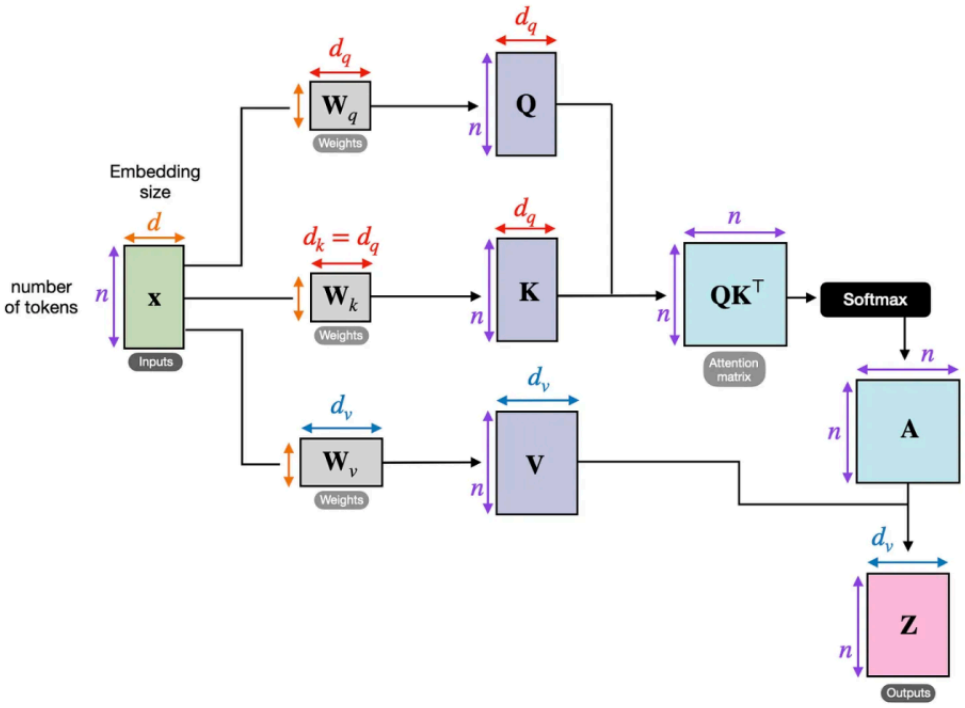

图 1: 自注意力机制流程图

自注意力背后的核心概念是缩放点积注意力（Scaled Dot Product Attention）。目标是建立一种注意力机制，序列中的任何元素都可以关注任何其他元素，同时仍能高效计算。点积注意力将一组查询 Q，键 K 和值 V（三者矩阵尺寸为 $T * d$，T 为序列长度，d 为查询、键或值的维度）。点积注意力的计算方法如下：

$$Attention\left(Q,K,V\right)=softmax\left(\frac{QK^{T}}{\sqrt{d_{k}}}\right)V \tag{2}$$

2.2.1 为什么使用 LN 而不是 BN，dropout 与 LN 同时使用时，有什么特殊操作？

2.2.2 为什么要基于学习的词向量空间？

2.2.3 可学习的位置编码？

2.2.4 teacher forcing 原理

3 从语义分割的几个桎梏说起

CNN 具有两种归纳偏置，

- 局部性，即图片上相邻的区域具有相似的特征；
- 平移不变性 $f\left(g\left(x\right)\right)=g\left(f\left(x\right)\right)$，其中 g 代表卷积操作，f 代表平移操作

当 CNN 具有以上两种归纳偏置，就有了很多先验信息，需要相对少的数据就可以学习一个比较好的模型

3.1 CNN 的局限性

从

3.1.1 UNet

到

3.1.2 PSPNet

再到

3.1.3 Deeplab 系列

CNN 的方法本质上存在着一个巨大的桎梏，就是图像初始阶段输入到网络之时，CNN 的卷积核不会太大，所以模型只能利用局部信息理解输入图像，难免有些一叶障目

3.1.4 HRNet

理论参考：Link

HRNet(图3) 通过**并行多个分辨率分支**以及不同分支之间的信息交互，获取强语义信息 & 精准位置信息。而在此之前几乎所有的网络 (图2) 都是**通过下采样得到强语义信息，再上采样恢复高分辨率信息**，导致**大量的有效信息会在不断的上下采样过程中丢失**。

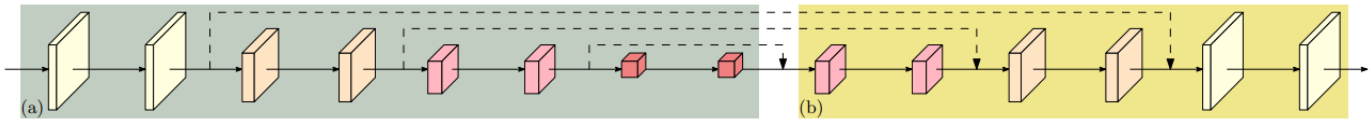

图 2: 由低分辨率恢复到高分辨率的网络结构图

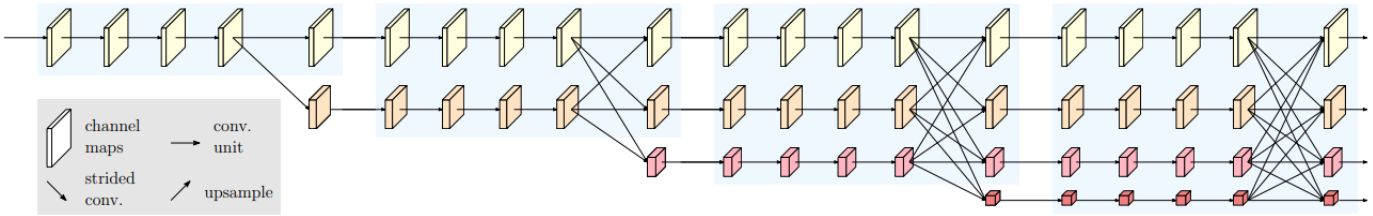

图 3: HRNet 网络结构图

3.2 标注数据的局限性

- 现实的数据往往缺乏标签

- 数据标注需要很大的资源投入————烧很多的钱
- 很多任务难以获取真实标签

3.3 模型的实时性

3.3.1 DDRNet[4]

- **应用场景：**高分辨率图像实时语义分割
- **动机：**
- **创新点：**1. 使用两个分辨率不同的分支，一个用于生成高分辨率的特征图，另一个通过多次降采样提取丰富的语义信息，并在两个分支之间建立多个双边连接，实现高效信息融合；2. 提出一个新的多尺度模块 DAPPM **标签：**一个效果还不错的实时语义分割网络

1. 深度双分辨率网络

2. DAPPM 作者认为，仅使用一个 3×3 卷积或 1×1 卷积将所有的多尺度上下文信息融合在一起是不够的。所以采用类似 **Res2Net** 的结构，首先对特征图进行上采样（灰色块），然后使用更多的 3×3 卷积，以层次残差 hierarchial-residual 的方式融合不同尺度的上下文信息（紫色块）。最后将所有得到的特征 concatenate 并使用一个 1×1 卷积压缩。此外，为了便于优化，还添加了一个 1×1 卷积映射作为 shortcut。

3.4 模型的泛化性

辛辛苦苦训练好了一个模型，换了一个场景后，模型的性能出现很大下滑。按传统方式解决这一问题，是将新场景的数据采集一遍，然后标注好，再重新训练。这工作量一言难尽。。。

4 Transformer，它来了

针对 CNN 模型不能在一开始就从全局理解输入图像的问题，基于 Transformer 的方案，是将输入的图像 Token 化，然后利用自注意力机制使得模型一开始就以全局的角度去理解图片。

4.1 开山之作 VIT

4.1.1 简介

ViT 是 2020 年 Google 团队提出的将 Transformer 应用在图像分类的模型，虽然不是第一篇将 transformer 应用在视觉任务的论文，但是因为其模型“简单”且效果好，可扩展性强，成为了 transformer 在 CV 领域应用的里程碑著作，也引爆了后续相关研究

ViT 中最核心的结论是，**当拥有足够多的数据进行预训练的时候，ViT 的表现就会超过 CNN，突破 transformer 缺少归纳偏置的限制，可以在下游任务中获得较好的迁移效果**

4.1.2 ViT 的结构

ViT 将输入图片分为多个 patch（16×16），再将每个 patch 投影为固定长度的向量送入 Transformer，后续 encoder 的操作和原始 Transformer 中完全相同。但是因为对图片分类，因此在输入序列中加入一个特殊的 token，该 token 对应的输出即为最后的类别预测

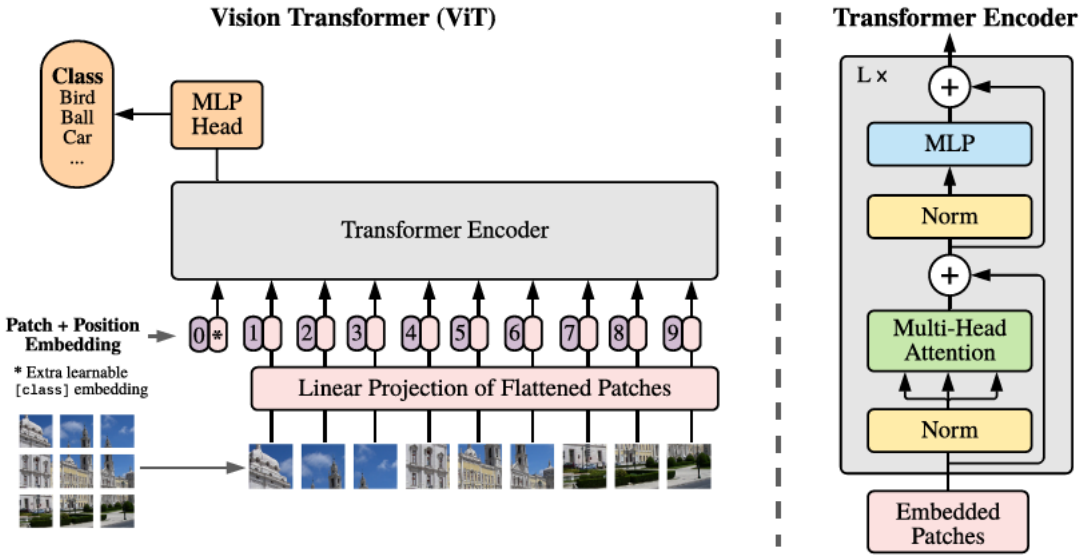

图 4: ViT 结构图

按照上面的流程图，一个 ViT block 可以分为以下几个步骤

- patch embedding：例如输入图片大小为 224×224，将图片分为固定大小的 patch，patch 大小为 16×16，则每张图像会生成 $224 \times 224 / 16 \times 16 = 196$ 个 patch，即输入序列长度为 196，每个 patch 维度 $16 \times 16 \times 3 = 768$，线性投射层的维度为 $768 \times N$ $(N = 768)$，因此输入通过线性投射层之后的维度依然为 196×768，即一共有 196 个 token，每个 token 的维度是 768。这里还需要加上一个特殊字符 cls，因此最终的维度是 197×768。到目前为止，已经通过 patch embedding 将一个视觉问题转化为了一个 seq2seq 问题（**也可以考虑使用 CNN 提取输入序列**）
- positional encoding：ViT 同样需要加入位置编码，位置编码可以理解为一张表，表一共有 N 行，N 的大小和输入序列长度相同，每一行代表一个向量，向量的维度和输入序列 embedding 的维度相同 $= 768$。**注意：**位置编码的操作是 sum，而不是 concat。加入位置编码信息之后，维度依然是 197×768
- LN/multi-head attention/LN：LN 输出维度依然是 197×768。多头自注意力时，先将输入映射到 q, k, v，如果只有一个头，qkv 的维度都是 197×768，如果有 12 个头，则 qkv 的维度是 197×64，一共有 12 组 qkv，最后再将 12 组 qkv 的输出拼接起来，输出维度是 197×768，然后在过一层 LN，维度依然是 197×768
- MLP：将维度放大再缩小回去，197×768 放大为 197×3072，再缩小变为 197×768

一个 block 之后维度依然和输入相同，都是 197×768，因此可以堆叠多个 block。最后会将特殊字符 cls 对应的输出作为 encoder 的最终输出（**另一种做法是不加 cls 字符，对所有的 tokens 的输出做一个平均**）

提醒：**关于 positional encoding**

- 1-D 位置编码：例如 $3 \times 3 = 9$ 个 patch，patch 编码为 1 到 9
- 2-D 位置编码：patch 编码为 11，12，13，21，22，23，31，32，33，即同时考虑 X 和 Y 轴的信息，每个轴的编码维度是 $D/2$

实际实验结果表明，不管使用哪种位置编码方式，模型的精度都很接近，甚至不使用位置编码，模型的性能损失也没有特别大。原因可能是因为 **ViT 是作用在 image patch 上的，而不是 image pixel，对网络来说这些 patch 之间的相对位置信息容易理解一些**

5 附录

5.1 半监督学习

提醒：

- 定义：半监督学习（Semi-Supervised Learning，SSL）属于弱监督学习的一个分支，具体地：弱监督学习可以分为三类：
  - 不完全监督【只有很小的子集有标签】，包含两个方向：
    - 主动学习
    - 半监督学习，按学习方式可分为纯半监督学习和直推学习
  - 不确切监督【只有粗粒度标签】
  - 不准确监督【标签不总是真实】
- 基本假设：未标记数据必须是有意义的，具有潜在价值的样本，而非无用的噪声样本
- 存在的问题:
  - 无效标签样本的有效利用问题
  - 大量无效标签样本的高效使用问题
  - 很多半监督算法都有无监督的影子，因此对于特征的选择非常的敏感

半监督语义分割方法大致可以分为 5 类：

- 类似于 GAN 结构生成对抗
- 一致性正则化【最小化同一图像的不同预测之间的差异】
- 伪标记方法

  - 自训练方法【依赖于现有模型的泛化性，缺乏检测自身错误的机制，同时需要解决伪标签中的类不均衡问题】
  - 互训练方法【基于分歧的策略】
- 基于对比学习
- 混合模式

5.1.1 Mean Teacher

核心思想是强制学生网络和教师网络的预测在存在扰动的情况下一致。其中教师网络的权重是通过学生网络权重的指数移动平均值（EMA）计算得出的。

扰动的方式有四种：

- 基于输入的扰动：CutOut、CutMix、ClassMix、Complex、其他方法
- 基于特征的扰动：交叉一致性训练（CCT）
- 基于网络的扰动，比如使用不同的起始权重
- 混合模式

参考文献

[1] 语义分割综述: 截止 2022, 语义分割总结与展望, https://zhuanlan.zhihu.com/p/538050231.

[2] J. Wang, K. Sun, T. Cheng, et al. Deep High-Resolution Representation Learning for Visual Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020, 43(10), 3349-3364.

[3] M. Xu, Z. zhang, H., Hu, et al. End-to-End semi-supervised object detection with soft teacher, *Computer Vision and Pattern Recognition*, 2021.

[4] Y. Hong, H. Pan, W., Sun, et al. Deep Dual-resolution Networks for Real-time and Accurate Semantic Segmentation of Road Scenes, *Computer Vision and Pattern Recognition*, 2021.