Expectation-Maximization Attention Networks for Semantic Segmentation

ICCV 2019

Xia Li, Zhisheng Zhong, Jianlong Wu, Yibo Yang, Zhouchen Lin, Hong Liu

汇报人：张灵西

学号：19210240022

总览

- 研究背景与主要贡献
- 理论基础
  - 期望最大化算法
  - 非局部网络
- EMANet介绍
- 实验
- 总结

研究背景与主要贡献

1

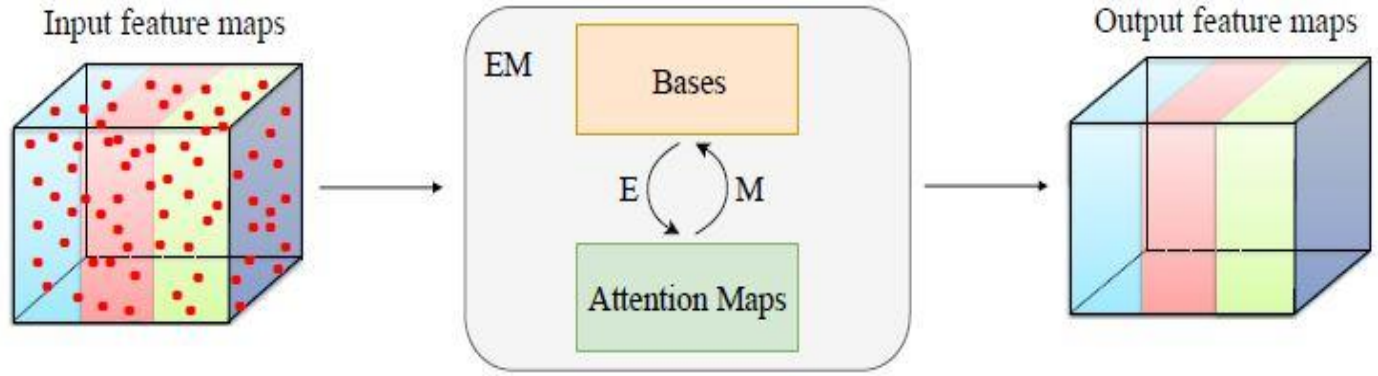
研究背景

- 语义分割：
  - 为每个像素预测类别标签
  - 自注意力机制在自然语言处理领域取得卓越成果，被一系列文章证明了在语义分割中的有效性（Nonlocal）

- 当前障碍：
  - 全卷积网络无法充分捕获长距离信息
  - 自注意力机制需要生成一个巨大的注意力图，其空间复杂度和时间复杂度巨大
  - 每一个像素的注意力图都需要对全图计算

本文主要贡献

- 本文第一次把EM算法加入到注意力机制中，能够得到一个更紧致的参数集合并且大幅减少计算复杂度；
- 将期望最大化注意作为神经网络的轻量级模块进行构建；
- 实验证明EMA算法的优越性高于经典算法



理论基础

2

理论基础

- 期望最大化算法
  - 期望最大化（EM）算法旨在为隐变量模型寻找最大似然解。对于观测数据$\boldsymbol{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_N}\}$，每一个数据点$\boldsymbol{x_i}$都对应隐变量$\boldsymbol{z_i}$。我们把$\{X, Z\}$称为完整数据，其似然函数为$\ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$，$\boldsymbol{\theta}$是模型的参数。
  - E步根据当前参数$\boldsymbol{\theta^{old}}$计算隐变量Z的后验分布，并寻找完整数据的似然$Q(\boldsymbol{\theta} \cdot \boldsymbol{\theta^{old}})$

$$Q(\boldsymbol{\theta} \cdot \boldsymbol{\theta^{old}}) = \sum_{\boldsymbol{Z}} p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta^{old}}) \ln p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})$$

  - M步通过最大化似然函数来更新参数得到$\boldsymbol{\theta^{new}}$

$$\boldsymbol{\theta^{new}} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta} \cdot \boldsymbol{\theta^{old}})$$

  - EM算法被证明会收敛到局部最大值处，且迭代过程完整数据似然值单调递增
  - 高斯混合模型是EM算法的一个范例，它把数据用多个高斯分布拟合。其$\boldsymbol{\theta_k}$为第k个高斯分布的参数$\boldsymbol{\mu_k}$，$\boldsymbol{\Sigma_k}$，隐变量$\boldsymbol{z_{nk}}$为第k个高斯分布对第n数据点的“责任”。E步更新“责任”，M步更新高斯参数

理论基础

- 非局部网络（nonlocal）
  - 核心算子：

$$\boldsymbol{y_i} = \frac{1}{C(\boldsymbol{X})}\sum_{\forall j} f(\boldsymbol{x_i}, \boldsymbol{x_j}) g(\boldsymbol{x_j})$$

  - 其中， f(·,·)表示广义的核函数，C(X)是归一化系数，它将第i个像素的特征$\boldsymbol{x_i}$更新为其他所有像素特征经过g变换之后的加权平均$\mathbf{y_i}$，权重通过归一化后的核函数计算，表征两个像素之间的相关度。

EMANet介绍

3

期望最大化注意力机制

- 本文提出的EMA算法包含三个操作：
  - responsibility estimation (A_E)
  - likelihood maximization (A_M)
  - data re-estimation (A_R)
- 输入的特征图为$X \in R^{N*C}$，基初始值$\mu \in R^{K*C}$，A_E估计隐变量$Z \in R^{N*K}$，即每个基对像素的权责。第k个基对第n个像素的权责可以计算为：

$$z_{nk} = \frac{K(x_n, \mu_k)}{\sum_{j=1}^{K} K(x_n, \mu_j)}$$

期望最大化注意力机制

- responsibility estimation (A_E)的功能和EM算法中的E部分一样。

$$Z = softmax(\lambda X(\mu^{\top}))$$

- likelihood maximization (A_M)的功能和EM算法中的M部分一样。

$$\mu_k = \frac{\sum_{n=1}^{N} z_{nk} x_n}{\sum_{n=1}^{N} z_{nk}}$$

- data re-estimation (A_R)A_E，A_M运行T次。重新计算X，记作$\tilde{X}$。

$$\tilde{X} = Z\mu$$

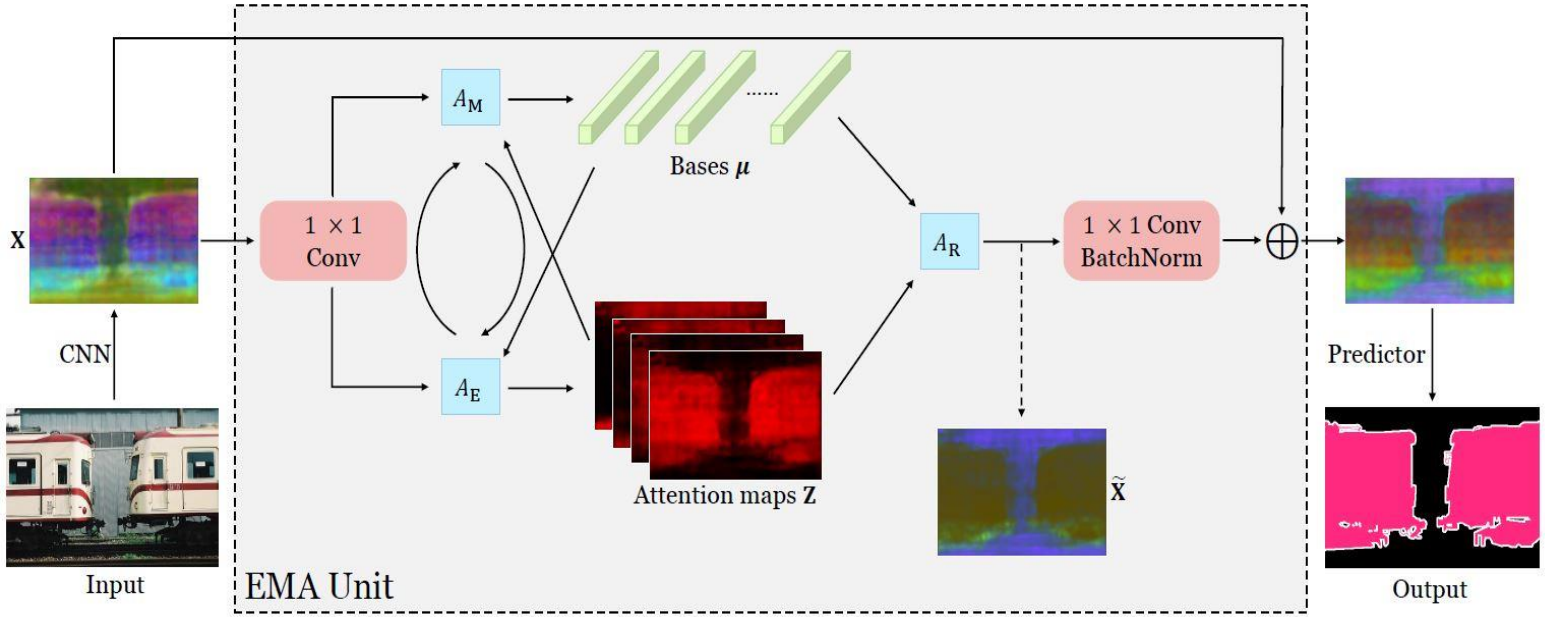
- EMA将复杂度从Nonlocal的$O(N^2)$降低至$O(NKT)$。T作为一个常数，可以被省去。因此，EMA复杂度仅为$O(NK)$。K<<N，复杂度得到显著降低

期望最大化注意力模块

- 放置两个1*1卷积于EMA前后。前者将输入的值域从R^+映射到R，后者将$\tilde{X}$映射到X的残差空间
- 迭代初值$\boldsymbol{\mu}^{(0)}$的维护采用滑动平均更新方式

$$\boldsymbol{\mu}^{(0)} \leftarrow \alpha\boldsymbol{\mu}^{(0)} + (1-\alpha)\bar{\boldsymbol{\mu}}^{(T)}$$

  - α ∈ [0,1]表示动量；$\bar{\boldsymbol{\mu}}^{(T)}$表示$\boldsymbol{\mu}^{(T)}$在一个mini-batch上的平均

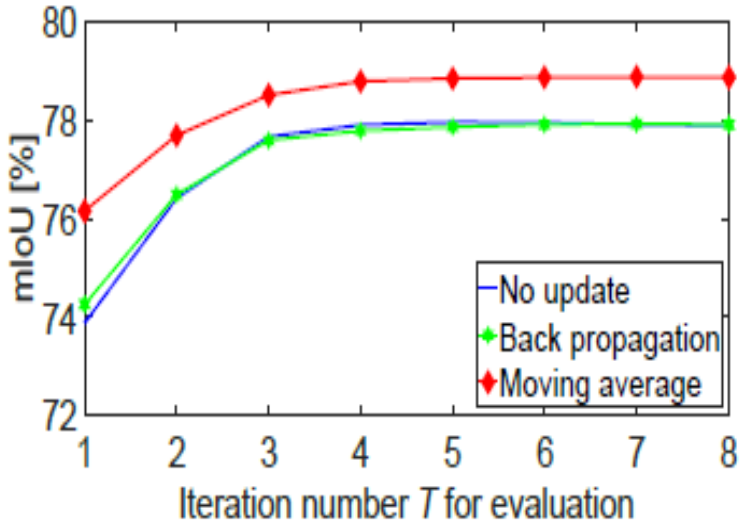


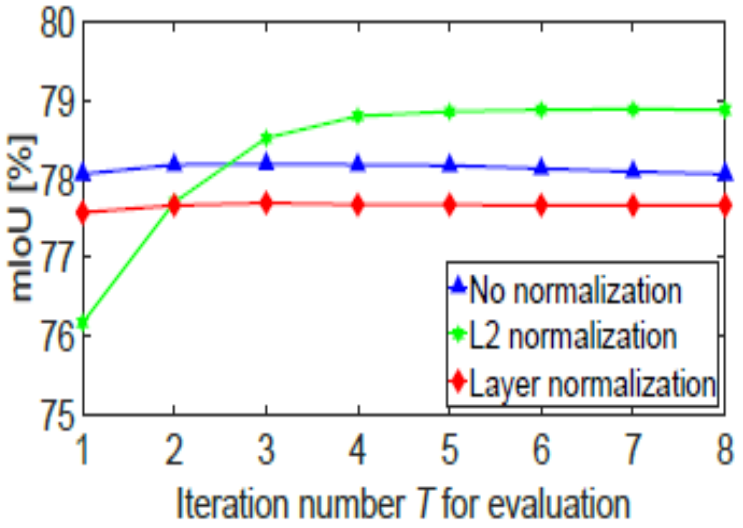
实验

4

实验

- 在PASCOL VOC上的实验，对比不同的 μ 更新方法和归一化方法的影响。可见，EMA使用滑动均值和L2Norm最为有效。





实验

- 不同迭代次数T的对比实验，可以发现，EMA仅需三步即可近似收敛。随着训练时迭代次数的继续增长，精度有所下降。

Training Iterations \ Evaluation Iterations (mIoU %)	1	2	3	4	5	6	7	8
1	77.34	77.52	77.60	77.59	77.59	77.59	77.59	77.59
2		77.75	78.04	78.15	78.15	78.12	78.12	78.17
3			78.52	78.80	78.86	78.88	78.89	78.88
4				78.14	78.25	78.27	78.28	78.27
5					77.70	77.76	77.82	77.86
6						77.85	77.91	77.92
7							77.11	77.14
8								77.24

实验

- EMANet和DeeplabV3、DeeplabV3+和PSANet的对比。

Table 1: Detailed comparisons on PASCAL VOC with DeeplabV3/V3+ and PSANet in mIoU (%). All results are achieved with the backbone ResNet-101 and output stride 8. The FLOPs and memory are computed with the input size 513×513. **SS**: Single scale input during test. **MS**: Multi-scale input. **Flip**: Adding left-right flipped input. EMANet (256) and EMANet (512) represent EMANet with the number of input channels as 256 and 512, respectively.

Method	SS	MS+Flip	FLOPs	Memory	Params
ResNet-101	-	-	190.6G	2.603G	42.6M
DeeplabV3 [4]	78.51	79.77	+63.4G	+66.0M	+15.5M
DeeplabV3+ [5]	79.35	80.57	+84.1G	+99.3M	+16.3M
PSANet [38]	78.51	79.77	+56.3G	+59.4M	+18.5M
EMANet (256)	79.73	80.94	**+21.1G**	**+12.3M**	**+4.87M**
EMANet (512)	**80.05**	**81.32**	+43.1G	+22.1M	+10.0M

实验

Table 2: Comparisons on the PASCAL VOC test set.

Method	Backbone	mIoU (%)
Wide ResNet [32]	WideResNet-38	84.9
PSPNet [37]	ResNet-101	85.4
DeeplabV3 [4]	ResNet-101	85.7
PSANet [38]	ResNet-101	85.7
EncNet [35]	ResNet-101	85.9
DFN [34]	ResNet-101	86.2
Exfuse [36]	ResNet-101	86.2
IDW-CNN [30]	ResNet-101	86.3
SDN [12]	DenseNet-161	86.6
DIS [23]	ResNet-101	86.8
EMANet	ResNet-101	**87.7**
GCN [25]	ResNet-152	83.6
RefineNet [21]	ResNet-152	84.2
DeeplabV3+ [5]	Xception-71	87.8
Exfuse [36]	ResNeXt-131	87.9
MSCI [20]	ResNet-152	88.0
EMANet	ResNet-152	**88.2**

Table 3: Comparisons with state-of-the-art on the PASCAL Context test set. '+' means pretrained on COCO Stuff.

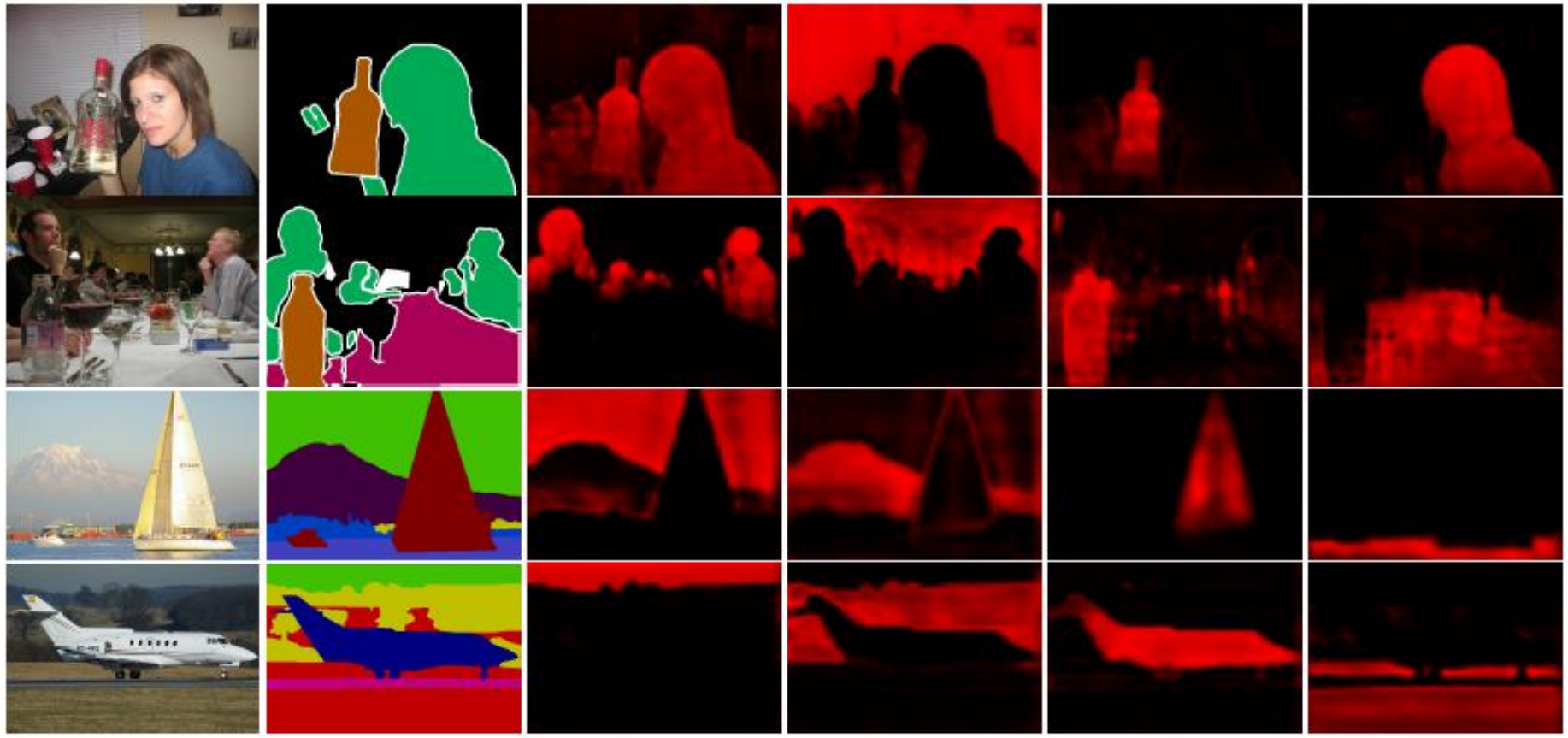
Method	Backbone	mIoU (%)
PSPNet [37]	ResNet-101	47.8
DANet [11]	ResNet-50	50.1
MSCI [20]	ResNet-152	50.3
EMANet	ResNet-50	50.5
SGR [18]	ResNet-101	50.8
CCL [8]	ResNet-101	51.6
EncNet [35]	ResNet-101	51.7
SGR+ [18]	ResNet-101	52.5
DANet [11]	ResNet-101	52.6
EMANet	ResNet-101	**53.1**

Table 4: Comparisons on the COCO Stuff test set.

Method	Backbone	mIoU (%)
RefineNet [21]	ResNet-101	33.6
CCL [8]	ResNet-101	35.7
DANet [11]	ResNet-50	37.2
DSSPN [19]	ResNet-101	37.3
EMANet	ResNet-50	37.6
SGR [18]	ResNet-101	39.1
DANet [11]	ResNet-101	39.7
EMANet	ResNet-101	**39.9**

实验

- 注意力图的可视化。



总结

5

总结

- Nonlocal方法都不能避免庞大的计算量，有很大的矩阵相乘。
- EMANet解决Nonlocal带来的计算量过于庞大。
- E步学习一组注意力图，M步更新一组基，经过迭代，用基和注意力图重构特征。
- 通过基和注意力图重构出高维的、带有全局性的信息的特征,用这个特征再去做分割

THANKS