Headline

大家好：

这是**2018年度第5篇Arxiv Weekly**。

本文是 模型研究 方向的文章。

Highlight

> 运用在**DenseNet**架构中的全新**Dropout**

Information

Title

Reconciling Feature-reuse and Overfitting in DenseNet with Specialized Dropout

Link

https://arxiv.org/pdf/1810.00091.pdf

Source

- 加利福尼亚大学圣塔芭芭拉分校（University of California, Santa Barbara, UCSB）

Introduction

近年来CNN网络在视觉和模式识别领域成为主流，而DenseNet是最新的CNN backbone，利用feature reuse在classification任务上实现了up-to-date的性能。

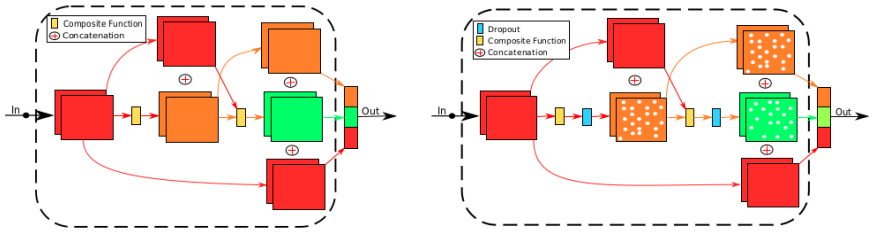
然而，DenseNet和其他的CNN网络一样，面临这overfitting问题，甚至因为模型的复杂和feature的复用而更加严重。为了解决overfitting，DenseNet引入了Dropout模块。然而由于DenseNet结构中的非线性因素（feature之间由于复用并非独立）dropout的效果会减弱；而dropout本身的引入也会导致DenseNet feature reuse效果的下降。

本文针对这些不足，从location、granularity和probability三个方面重新考虑设计dropout模块。本文设计的特殊dropout模块能够迁移到类似存在模型结构相关性的模型中，并辅助那些模型涨点。

实验表明运用了本文设计的dropout模块后，DenseNet达到了state-of-the-art的性能，并且其性能的增益会随着模型的加深而提高。

Keys

传统DenseNet和其中的Dropout模块如下图所示



(a) One *dense block* in the DenseNet (b) DenseNet with standard dropout

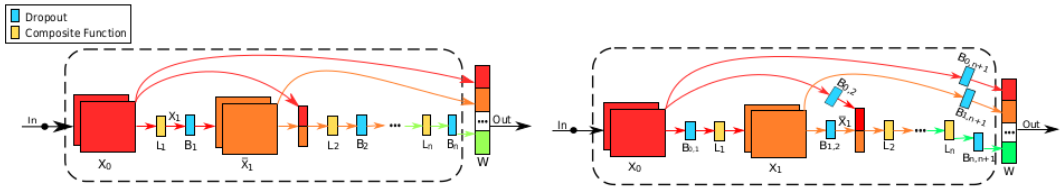
Fig.1 传统DenseNet和dropout基本模式示意图

而这样的模块有如下的不足：

- dropout把原本的feature刺穿，妨碍了feature reuse。
- feature reuse机制下各个layer之间的相关性也降低了dropout的性能。

本文中从三个角度提出了改进版本的specialized dropout

###1. pre-dropout结构



(a) Standard dropout (b) Pre-dropout

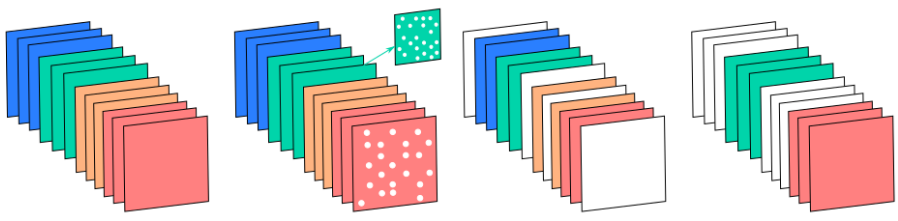
Fig.2 Standart dropout和pre-dropout的对比图

可以看到，原本的feature刚“诞生”立刻被dropout插得千创百孔，而且后面每次reuse都固定在这个“受伤”状态。这样无疑会导致一定程度的feature reuse性能受损。

所谓的pre-dropout是为了解决这个问题而设计的，简单说就是改为在Composition Function（在DenseNet中指BN-ReLU-3×3Conv的组合函数）之前进行dropout。

这样一来，每次需要复用feature时，都能够拿到“完整”的feature，从而最大程度地保留信息。而每一次reuse之前进行不同的dropout，可能够最合理化地增益droput防止overfitting的性能。可以说是一举两得。

###2. channel-wise dropout



(a) Original input (b) Unit-wise dropout (c) Channel-wise dropout (d) Layer-wise dropout

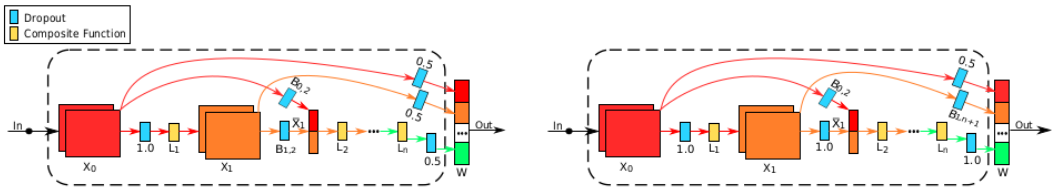
Fig.3 Unit-wise, Channel-wise, Layer-wise dropout对比图

最开始，dropout（也即 unit-wise dropout）是被用在FC layer当中，降低网络的过度冗余。当这个方法被迁移到Conv layer后，其实面临这一个问题，Conv layer获得的feature map中，相邻节点的元素其实是有相关性的。单纯的unit-wise dropout去除的elements中附带的信息，会被相邻的没有drop掉的elements弥补，降低dropout的性能。

所谓的channel-wise和layer-wise dropout就是每次随机删除一个feature map或者一个layer中的所有feature map（当然layer wise仅仅在denseNet中才有意义，正常的架构这么干就drop到啥都不剩了）。

layer-wise策略显然过度激进了，在drop的时候丢失了太大的信息，并且每次drop可选择的范围也有限。因此拍脑袋想和实验表明，对于DenseNet架构，尤其是growth rate相对较大的情况下，选用channel-wise dropout是比较合理的。

###3. stochastic probability schedule



(a) Schedule v_1 (b) Schedule v_3

Fig.4 不同调度策略

针对dropout的另一个没道理的事情是，为什么每个dropout的dropout rate都是一样的呢？[查阅DenseNet的文章你会发现，原结构中使用的是0.2的dropout rate，除了第一个Conv layer的输出外，每个Conv layer的输出都被随机drop掉了20%]

实际上本文发现，之所以每个dropout的dropout rate设置成一样的，是因为之前的作者懒得抠这个细节。而本文作为专门探讨DenseNet中dropout的文章显然需要抠一抠。

因此文章设计了几种dropout rate的概率调度方式，来探索这个问题更优化的解法。

- 第一个layer的输出dropout rate为0，最后一个layer的输入dropout rate为0.5；中间的dropout rate线性递减。
- 第一个layer的输出dropout rate为0.5，最后一个layer的输入dropout rate为1；中间的dropout rate线性递增。
- 最靠近本层的layer输入dropout rate为0，最远离本层的layer输入dropout rate为0.5；中间的dropout rate线性变化。

从DenseNet自身的特点来看，它对high-level信息的利用率高于low-level信息。因此最适合DenseNet的调度策略是第三种。

Results

Structure and method	Depth	Params	C10	C100
FitNet (Romero et al., 2014)	19	-	8.39	35.04
Deeply Supervised Net (Lee et al., 2015)	-	-	7.97	34.57
Highway Network (Srivastava et al., 2015)	-	-	7.72	32.39
ResNet v1 with Stochastic Depth (Huang et al., 2016b)	110	1.7M	5.23	24.58
ResNet v2 (He et al., 2016b)	164	1.7M	5.46	24.33
ResNet v2 (He et al., 2016b)	1001	10.2M	4.92	22.71
Swapout v2 $W \times 2$ (Singh et al., 2016)	20	1.09M	5.68	25.86
Swapout v2 $W \times 4$ (Singh et al., 2016)	32	7.46M	4.76	22.72
DenseNet-BC	76	0.5M	5.21	24.09
DenseNet-BC(standard dropout)	76	0.5M	5.56	24.75
DenseNet-BC(our specialized dropout)	76	0.5M	4.94	23.90
DenseNet-BC	100	0.8M	4.73	23.22
DenseNet-BC(standard dropout)	100	0.8M	5.01	23.80
DenseNet-BC(our specialized dropout)	100	0.8M	4.51	22.33
DenseNet-BC	148	1.5M	4.31	20.76
DenseNet-BC(standard dropout)	148	1.5M	4.60	22.28
DenseNet-BC(our specialized dropout)	148	1.5M	**3.90**	**19.75**

Fig.5 CIFAR上实验结果

文章的主要实验结果如上图所示，可以看到，综合了多方面考虑的specialized dropout确实有不俗表现。其不俗之处主要在于一方面涨点，另一方面在不同的深度下较为稳定地涨点。

文章中还进行了多组对比实验，说明Keys中的策略的有效性。大家不用看基本也能把数据编出来，故略过不表。

Model	Depth	C10	C100
AlexNet	8	10.32	38.71
AlexNet	8	9.78	35.42
VGG-16	16	8.39	35.04
VGG-16	16	7.25	33.71
ResNet v1	110	6.41	27.22
ResNet v1	110	5.45	25.46
ResNet v2	164	5.46	24.33
ResNet v2	164	4.38	23.36

Fig.5 Specialized Dropout在其他backbone上的表现

另外大家肯定关注的一点是，这个方法离开了DenseNet的语境是不是还work。从文章的实验来看，至少是蛮有希望work的。从作者列举的几组实验来看，都有相当不错的涨点。

最后一件重要的事情是，脱离了CIFAR的语境是不是还work。文章对此没有提及。[我猜想也许是穷所以没卡跑ImageNet实验]。故而如果模型研究小组有兴趣，我们可以简单跑一跑这些dropout的思路。如果有明确涨点，就可以作为一个模型的小trick保留下来。

Insights

主要是针对NN的基本组块——Dropout模块的细节探索。针对CIFAR和DenseNet的数据集和backbone进行了改进方案设计和综合实验，取得了不错的结果。

这样的小trick不显眼，但是胜在能赚点数、容易实验和广泛应用、没有太大的代价。如果我们自己仿真验证有效，或者仿真验证了更好的方案，感觉可以长期保留下来，作为模型搭建和训练tricks的积累。