Beyond Dropout: Feature Map Distortion to Regularize Deep Neural Networks (AAAI 2020)

这篇paper是关于dropout的，与集成学习相关，文中引入了一个Rademacher complexity的机器学习理论，比较难懂，有兴趣的话看下面两个学习资料（我自己是看完了这两个才弄懂）：

- b站：https://www.bilibili.com/video/BV1Tk4y1k7m7?spm_id_from=333.337.search-card.all.click
- 知乎：Rademacher复杂度&VC维-最通俗的类比讲解 - 知乎 (zhihu.com)

Background

DL中数据多了容易出现过拟合，从loss角度上看其实是训练集上的平均loss不能很好的代替全局数据的loss，dropout是通过在训练过程中随机丢弃一些神经元或者权重来解决过拟合问题的，也就是一个正则化技术，这个思路其实和集成学习、数据增强有点像，在我理解中这种随机的变动其实是一种隐式的集成学习或者说数据增强，这也是我关注dropout的原因。

Related Work

dropout其实分为两种，一种是随机丢弃特征的，另一种是随即丢弃weight的，但两者处理上都差不多是按照下面这个方式（就是加一个mask）：

$$\hat{\boldsymbol{f}}^l(\boldsymbol{x}_i) = \boldsymbol{f}^l(\boldsymbol{x}_i) - \boldsymbol{m}_i^l \circ \boldsymbol{f}^l(\boldsymbol{x}_i)$$

最开始的方法是确定一个dropout概率后随机dropout，这种肯定不是最优的，后来方法开始把dropout概率和程度放入模型一起优化，目的是更好地降低训练集上的平均loss和全局数据的loss之间的gap。

论文提到通过把训练过程中的一些feature以一定概率随机变为0，以此让model在该训练阶段关注其余非0 feature，这种方法实际上是一种启发式算法，不一定能找到最有情况，暂时还没明白为什么可以看作启发式算法。

Introduction

这篇文章提出的方法是dropout中的扰动feature maps，并且是以优化的形式确定相关参数，扰动优化的目标是最小化generalization bound（实际上表示训练集上的平均loss和全局数据的loss之间的gap，这里是用了一个机器学习中的Rademacher complexity理论）。除了dropout之外，论文的方法不修改其它模块的优化，也就是说像对于图像分类，最后的分类loss还是和正常一样优化的。

Preliminary, Motivation(加粗部分)

- 1

上面提到的训练集上的平均loss和全局数据的loss之间的gap，其实就是在说模型的泛化性能，拿分类为例，真实loss和平均loss如下:

$$R(\boldsymbol{f}^L) = \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{Q}}[\![\ell(\boldsymbol{f}^L(\boldsymbol{x},\mathcal{K}^{:L}),\boldsymbol{y})]\!],$$
$$\hat{R}(\boldsymbol{f}^L) = \frac{1}{N}\sum_{(\boldsymbol{x}_i,\boldsymbol{y}_i)\in\mathcal{D}} \ell(\boldsymbol{f}^L(\boldsymbol{x}_i,\mathcal{K}^{:L}),\boldsymbol{y}_i),$$

已经有理论和实践证明利用dropout事实上能够缩小上面两者的gap，也就是提升模型的泛化性能。

那这种gap怎么样去衡量呢，常用的方法有 PAC、VC维、Rademacher complexity等，**这其中Rademacher complexity已经杯广泛应用了，它能更精确的衡量这个gap，因此论文就是用Rademacher complexity的。**

- 2

先贴出论文里面给得Empirical Rademacher complexity公式:

$$\tilde{R}_D(\boldsymbol{f}^L) = \frac{1}{N}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left|\sup_{k,\mathcal{K}^{:L}}\sum_{i=1}^{N}\sigma_i\boldsymbol{f}^L(\boldsymbol{x}_i,\mathcal{K}^{:L})[k]\right|$$

我去学了下Rademacher complexity相关书上的理论以及推导，感觉这里和定理还是稍微有点不一样，拿分类来说，假如有100类，这里

$$\boldsymbol{f}^L(\boldsymbol{x}_i,\mathcal{K}^{:L})[k]$$

应该是指第i个样本属于第k个类别的概率（激活以后应在0-1之间），也就是它sup的范围是100个类别对应的100个值；书上定义的Rademacher complexity在这里是第k个分类器下的loss，假设我们手里有100个可选用的classfier，那么它sup的范围就是这100个classifier对应的100个值。

这个公式到底衡量了什么呢，这里不展开说，直接说结论（论文里也是没给证明，书上有），先说书上定义的，也就是说Rademacher complexity本意衡量的是100个classfier的差异；那么在这里就是衡量100个类别最后的评分差异大小。

进一步，那这个差异对我们要minimize的generalizat gap有什么用呢，下面这个理论就是把上面说的两者放在了一起:

$$R(\boldsymbol{f}^L) \le \hat{R}(\boldsymbol{f}^L) + \frac{2(d^L)^2}{\rho}\tilde{R}_D(\boldsymbol{f}^L)$$
$$+\left(1+\frac{2(d^L)^2}{\rho}\right)\sqrt{\frac{\ln\frac{1}{\delta}}{2N}},$$

$$R(\boldsymbol{f}^L) \leq \hat{R}(\boldsymbol{f}^L) + \frac{2(d^L)^2}{\rho}\tilde{R}_D(\boldsymbol{f}^L) + \left(1 + \frac{2(d^L)^2}{\rho}\right)\sqrt{\frac{\ln\frac{1}{\delta}}{2N}},$$

看我移项后的这个版本应该就看懂了，我们前面的这个gap就是式子左端，它的上界就是右端这个，右端中变量只有我们上面的Rademacher complexity，这就是说我们要获得一个比较小的generalizat gap，可以通过把Rademacher complexity降低来实现；换句话说，如果Rademacher complexity比较大，也就是k个类别输出的值差别比较大，那么泛化性能就比较低，至此为止听起来还是挺合理的，思考一下，如果k个类别输出的值差别比较大，意味着有些类别输出值比别的都高很多，这往往就是过拟合的表现，泛化性能就比较低。

- 3

论文说直接计算Rademacher complexity比较难，一般都是给它找个上界或者是估计它。这里我没有很懂，如果直接按照上面那个式子不是是能实现的吗。我没看这里的引文，没有深究。

论文还提到，近年几篇工作通过减小一个包含ERC的正则项，实现了比较好的性能，论文受此启发想把降低ERC这个idea融入dropout中。

Approach

- 1

论文提出的方法和传统方法一个比较大的区别是，论文没有固定在做dropout时的幅度，也就是下面公式：

$$\hat{\boldsymbol{f}}^l(\boldsymbol{x}_i) = \boldsymbol{f}^l(\boldsymbol{x}_i) - \boldsymbol{m}_i^l \circ \boldsymbol{\varepsilon}_i^l,$$

开头提到了传统方法：

$$\hat{\boldsymbol{f}}^l(\boldsymbol{x}_i) = \boldsymbol{f}^l(\boldsymbol{x}_i) - \boldsymbol{m}_i^l \circ \boldsymbol{f}^l(\boldsymbol{x}_i)$$

m是mask，取0或1，两种方法差别就在最后一项，传统dropout的幅度都是拉满的，要么留下要么drop掉；而论文方法是要么留下要么drop掉一部分，按我理解dropout这个变成了一个连续值上的事情，利于优化。

- 2

前面提到ERC不好直接计算，需要找上界或者近似来做，下面定理给出了一个途径：

Theorem 2 *Let $\mathcal{K}^l[k,:]$ denotes the k-th row of the weight matrix $\mathcal{K}^l$ and $\|\cdot\|_p$ is the p-norm of vector. Assume that $\|\mathcal{K}^l[k,:]\|_p \leq B^l$, and then the ERC of output can be bounded by the ERC of intermediate feature:*

$$\begin{aligned}&\tilde{R}_D(\boldsymbol{f}^L) \leq 2\tilde{R}_D(\boldsymbol{o}^L) \leq 2B^L\tilde{R}_D(\boldsymbol{f}^{L-1}) \leq \cdots \\ &\leq 2^{L-t}\tilde{R}_D(\boldsymbol{f}^t)\prod_{l=t+1}^{L}B^l \leq 2^{L-t+1}\tilde{R}_D(\boldsymbol{o}^t)\prod_{l=t+1}^{L}B^l,\end{aligned} \tag{7}$$

同样，先不管内部细节，直接看结论，结论就是我们最后输出层f^L的ERC的上界是由任意一层输出f^t的ERC决定的。因此，我们要降低最后输出层f^L的ERC，可以采用一种启发式的方式，也就是说降低任意一层输出f^t的ERC。

顺着上面思路，如果我们以降低该层ERC为原则对某层dropout后，那么它之后层的ERC都会受到影响，根据上面Theorem2， 相隔越近的层受影响应该越大，所以离output层越近的层做dropout对于整个model的ERC影响越大。

具体来说，就是在对第l层做dropout时候，要以最小化第$l+1$层的ERC为目标，也就是下面这个式子：

$$\tilde{R}_D(\boldsymbol{o}^{l+1}) = \frac{1}{N}\mathbb{E}_{\boldsymbol{\sigma}}\sup_{k,\mathcal{K}:l+1}\left|\left\langle\mathcal{K}^{l+1}[k,:]^T,\boldsymbol{g}^l(\boldsymbol{x})\right\rangle\right|,$$

$$\boldsymbol{g}^l(\boldsymbol{x}) = \sum_{i=1}^{N}\sigma_i\hat{\boldsymbol{f}}^l(\boldsymbol{x}_i),$$

$$\hat{\boldsymbol{f}}^l(\boldsymbol{x}_i) = \boldsymbol{f}^l(\boldsymbol{x}_i) - \boldsymbol{m}_i^l \circ \boldsymbol{\varepsilon}_i^l,$$

(这里我有疑问的一点是，$o^{(l+1)}$按文中说法是没有激活过的，那么它应该不满足0-1之间啊，为什么能用ERC公式，当然前面Theorem2里面就有这个问题，Theorem2是从别的paper来的，可能在原paper种有说明吧，这里反正打个问号。)

抛开上面这个问题继续看，根据前面的内容，找到合适的dropout程度，其实等价于求解下面这个优化：

$$\hat{\boldsymbol{\varepsilon}}^l = \arg\min_{\boldsymbol{\varepsilon}^l}\mathcal{T}(\bar{\boldsymbol{x}},\boldsymbol{\varepsilon}^l),\ \ l=1,2,\cdots,L$$

where

$$\begin{aligned}&\mathcal{T}(\bar{\boldsymbol{x}},\boldsymbol{\varepsilon}^l) \\ &= \frac{1}{\bar{N}}\left[\sup_k\left|\left\langle\mathcal{K}^{l+1}[k,:]^T,\boldsymbol{g}^l(\bar{\boldsymbol{x}})\right\rangle\right| + \frac{\lambda}{2}\sum_{i=1}^{\bar{N}}\|\boldsymbol{\varepsilon}_i^l\|_2^2\right],\end{aligned}$$

(论文提到，传统的方法其实就是本文方法的一个特例，也有一定可能性能降低ERC，但是如此随机就完全不考虑语义信息了，也就是语义信息没被利用起来做优化。)

继续看，我们对合适的dropout程度做优化，那就需要梯度下降，来看一下这里的梯度：

$$\frac{\partial \mathcal{T}(\bar{\boldsymbol{x}}, \boldsymbol{\varepsilon}^l)}{\partial \boldsymbol{\varepsilon}_i^l} = -\frac{1}{\bar{N}} \sigma_i s_{\hat{k}} \mathcal{K}^{l+l}[\hat{k}, :]^T \circ \boldsymbol{m}_i^l + \frac{\lambda}{\bar{N}} \boldsymbol{\varepsilon}_i^l,$$

where

$$\hat{k} = \arg\max_k \left| \left\langle \mathcal{K}^{l+1}[k, :]^T, \boldsymbol{g}^l(\bar{\boldsymbol{x}}) \right\rangle \right|,$$

$$s_{\hat{k}} = \operatorname{sign} \left\langle \mathcal{K}^{l+1}[\hat{k}, :]^T, \boldsymbol{g}^l(\bar{\boldsymbol{x}}) \right\rangle.$$

这个梯度结果告诉了我们，我们在第l层确定最优的扰动量时，需要考虑本层的特征输出以及第$l+1$层的变换矩阵。

实际上，要精确计算并实现上面这个梯度下降过程比较耗费时间，论文提出了一个近似手段来优化，也就是下面这个公式：

$$\frac{\partial T^{l+1}}{\partial \boldsymbol{\varepsilon}_i^l} \approx -\frac{1}{\bar{N}} \sigma_i \boldsymbol{u} \circ \mathcal{K}_M^{l+1} \circ \boldsymbol{m}_i^l + \frac{\lambda}{\bar{N}} \boldsymbol{\varepsilon}_i^l,$$

$$\boldsymbol{\varepsilon}_i^l \leftarrow \boldsymbol{\varepsilon}_i^l - \gamma \frac{\partial T^{l+1}}{\partial \boldsymbol{\varepsilon}_i^l}.$$

- 3

至此，这一块关于如何确定扰动量的part就完了，现在就要考虑把这个优化过程和原本就有的优化过程（比如在图像分类中就是最后优化那个CE loss）结合在一起，论文提出是说轮流来，其实就是和GAN差不多，拿图片分类具体来说，就是先训练一步CE loss，再训练几步扰动量，这个过程循环进行。

（此外，作者还把这个dropout方法沿用在了CNN 上，但我目前还没有用到这块，就先不看了）