HMG: Meta-Gradient与FGSM混合的图数据攻击方法

张逸凯 171840708
南京大学
计算机科学与技术系
zykhelloha@gmail.com

Abstract

生活中图结构无处不在, 最近, 研究者发现很多深度学习技术可以用于图数据. 不过, 深度学习模型的健壮性却是大问题. 在机器学习课程大作业: *KDD Cup 2020*中, 我对图数据进行了黑盒攻击(即被攻击模型结构与参数未知), 提出了*HMG*方法(*Hybrid Meta-Gradient Method for Graph Data Attack*), 对图结构数据采用*Meta-Gradient*解决二层优化, 并使用贪心原则保持图的离散性, 结合已有的*Fast Gradient Sign Method(FGSM)*方法对特征进行攻击, 取得了第*8*名的成绩(*6*月*14*日*18:00*). 值得注意的是, *HMG*是优化的攻击图结构是模型无关的.

1. 介绍 (Introduction)

1.1. 背景 (Background)

在自然界和社会科学中, 图结构无处不在, 包括分子结构、社交网络、论文引用网络和细胞信号通路等. 其中, 最早于1965年被提出的论文引用网络是一种典型的图结构: 论文是节点, 引用关系是边. 研究论文引用网络可以在很多领域得到引用, 比如科学影响评估, 知识发现, 以及技术预见等.

深度学习已经在很多领域获得了成功. 最近, 研究者发现很多深度学习技术可以用于图数据. 不过, 深度学习模型非常容易受到攻击. 一个非常类似原数据的对抗样本, 可以极大地拉低分类器的性能. 这种情况在图数据上也会出现. 2018年的KDD最佳论文就发现极其轻微的扰动就能让节点分类器的准确率大幅下降. [2]

1.2. 任务 (Main Task)

对KDD Cup 2020竞赛组织者提供的图数据进行攻击, 并拉低组织者的节点分类器的准确率, 这里的攻击定义为:

> 为提供的图数据添加不超过500个新的节点, 且每个新的节点最多只能有100条边.

已有的图包括593,486个节点, 每个节点都有一个100维的特征. 其中543,486个节点是训练数据, 50,000个节点是测试数据. 组织者在后台提供一个节点分类器, 为50,000个测试节点进行分类. 分类器不会被发布.

本报告遵循Siggraph原则, 尽量简洁干练.

2. 方法 (Method)

解决该问题的大致思路为: 训练一个替代模型生成测试集的伪标签 $\rightarrow$ 对图数据进行攻击使模型最大程度偏离训练集标签或测试集伪标签. 其中关键为找到攻击后使模型越来越容易出错的优化方向.

为了解决上述关键的优化方向, 以下我将分为两个部分叙述:

- 图结构攻击: Meta-Gradient Method.
- 图特征攻击: Fast Gradient Sign Method.

该方法核心部分引用自 [8], 在更新方式上做了较大改动.

2.1. Meta-Learning

本节将初步介绍基于优化的 Meta-Learning(通过MAML [1] 的例子), 以便您更快理解在图上进行的 Meta-Gradient.

- **元学习(Meta-Learning)**

  即 learning to learn, 学习: 如何更快更好地学习. 学会学习方法. 使其对新任务可以快速适应.

一个更直观的理解: 令一个学习方法为F, 数据集为$\mathcal{D}$, 模型预测函数为$f_{\theta^\star}$, Meta-Learning的目标为学得一个F:

$$F(\mathcal{D}) = f_{\theta^\star}$$

并使模型更快训练得到$f_{\theta^\star}$, 得到的结果更优.

此时令"学习方法"为: 模型的参数初始值θ_0, 即学得一个好的模型参数初始值 θ_0, 对于不同任务, **在训练之后**, 可以很快得到很好的预测函数$f_{\theta^\star}$.

不妨令任务集合为$\mathcal{T}$, 令第i个任务上的损失函数为$l_{\mathcal{T}_i}$, 上述Meta-Learning任务的损失函数形式化为:

$$\mathcal{L}(\theta_0) = \sum_{\mathcal{T}_i \in \mathcal{T}} l_{\mathcal{T}_i}(f_{\theta_i^\star})$$

请注意$f_{\theta_i^\star}$是在任务$\mathcal{T}_i$上以θ_0初始化后模型训练后的预测函数. 如上二层优化是接下来攻击图结构方法的关键.

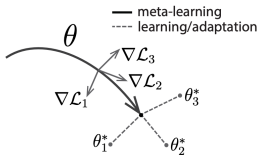

图 1. MAML对Train Tasks求解梯度

更新方向可大致见图1.

2.2. 图结构攻击

2.2.1 结构攻击形式化

如上, 将Meta-Learning学习方法定义为: 攻击后的图$\hat{G}$, 类似地, 任务集合$\mathcal{T}$定义为图的不同块(竞赛中图非常大, 必须分块处理), 可以写出该问题的优化目标:

$$\hat{G}^\star = \underset{\hat{G}}{\operatorname{argmax}}\, \mathcal{L}(\hat{G}) = \sum_{\mathcal{T}_i \in \mathcal{T}} l_{\mathcal{T}_i}\left(f_{\theta_i^\star}(\hat{G})\right) \quad (1)$$

其中: $\theta_i^\star = \underset{\theta_i}{\operatorname{argmin}}\, l_{\mathcal{T}_i}\left(f_{\theta_i}(\hat{G})\right)$

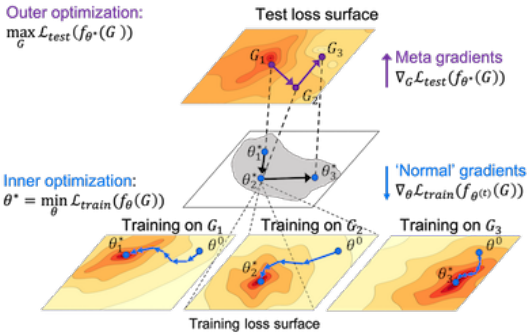

图 2. 单步Meta-Gradient即更新的图示

更新图梯度与模型参数梯度的关系可见图2, 但是请注意上图在单步更新模型之后就更新了图, 与上述方法略有不同.

2.2.2 简化与求解

不妨将优化目标1中损失函数添加负号, 即argmax 转换为 argmin, 使用Meta-Gradient对其$\hat{G}$进行优化, 结合链式法则我们有:

$$\begin{aligned}\nabla_{\hat{G}}^{\text{meta}} &= \sum_{\mathcal{T}_i \in \mathcal{T}} \nabla_{\hat{G}} l_{\mathcal{T}_i}\left(f_{\theta_i^\star}(\hat{G})\right) \\ &= \sum_{\mathcal{T}_i \in \mathcal{T}} \nabla_f l_{\mathcal{T}_i}\left(f_{\theta_i^\star}(\hat{G})\right) \cdot \left(\nabla_{\hat{G}} f_{\theta_i^\star}(\hat{G})\right. \\ &\quad \left. + \nabla_{\theta_i^\star} f_{\theta_i^\star}(\hat{G}) \cdot \nabla_{\hat{G}} \theta_i^\star\right)\end{aligned} \tag{2}$$

其中θ的更新为模型训练时的反向传播. 请注意该方法与 [1] 中的MAML学习框架略有不同, MAML在一次的模型参数更新后即更新学习方法(参数初始值), 但是这里更类似 Reptile 方法 [5] 的多次参数更新后再往那个方向更新学习方法.

2.2.3 求导后离散化: 贪心处理

Meta-Gradient的结果是连续的, 但是在更新图时保持图的离散结构, 即保持邻接矩阵非0即1. 在每一轮迭代时, 选择$\nabla_{\hat{G}}^{\text{meta}}$可攻击的点中梯度最大的边, 即在邻接矩阵对称的两元素, 并进行翻转$(0 \to 1,\ 1 \to 0)$.

2.3. 图特征攻击

上述方法理论上也可用于图特征攻击, 但是这里使用更简单的Fast Gradient Sign Method(FGSM)已经达到了较好的效果.

该方法大致思路为沿损失函数的正梯度方向加入了固定的噪声(一个小量), 并限制了特征的范围(因为竞赛提交后输入模型有归一化处理, 过大的特征没有意义).

2.3.1 特征攻击形式化

不妨令特征矩阵为$\boldsymbol{X}$, 模型损失函数为$\mathcal{L}$, 则更新$\boldsymbol{X}$的过程为:

$$\hat{\boldsymbol{X}}_{\boldsymbol{ij}} = \begin{cases} \boldsymbol{X}_{\boldsymbol{ij}} + \epsilon \cdot \operatorname{sign}\left(\nabla_{\boldsymbol{X}_{\boldsymbol{ij}}} \mathcal{L}\right), & \boldsymbol{X}_{\boldsymbol{ij}} \in [lb, ub] \\ \boldsymbol{X}_{\boldsymbol{ij}}, & otherwise \end{cases}$$

其中ϵ为固定的噪声, $lb,\ ub$ 为归一化限制的上下界.

这里的简化与求解较容易. 实现方面我结合了交叉熵损失和负对数似然损失, 得到了一些提优.

2.4. 关于替代模型

如前所述, 我们需要一个被攻击模型的替代者, 这个替代模型的训练不是本次任务的重点, 这里简要说明一下它的实现:

使用`PyTorch`框架, 我构建了一个两层图卷积神经网络(GCN), 如图3:

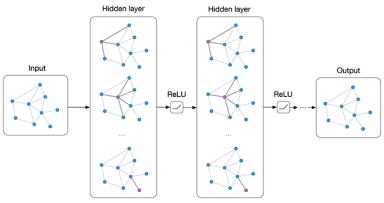

图 3. 图卷积神经网络

不妨令邻接矩阵为A, 度数矩阵D, 其中

$$D_{i,j} = \begin{cases} d_i & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

图上Laplacian算子为$L = D - A$, 对其求解Fourier基后可将其表达式写为(简化版, σ是输出层激活函数):

$$H^{(l+1)} = \sigma\left(D^{-\frac{1}{2}}AD^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$$

替代模型的准确率大致在60+%.

2.5. 竞赛提优技巧

我使用过伪标签融合, 对不平衡结点采样, 构造特征时考虑方差等技巧, 但是都没有很好的效果.

多层感知机的攻击结果集成, 大约降低模型准确率3%.

2.5.1 多层感知机集成

我搭建了一个准确率大概相同的多层感知机, 仅对特征矩阵处理, 其攻击方法与图卷积神经网络对特征的攻击相同, 不再赘述.

其特征攻击结果与上述方法的使用Bagging投票集成, 但是由于多层感知机预测时没有考虑图结构, 使集成效果没有很好.

3. 算法 (Algorithm)

算法关键部分的过程为:

训练一个替代模型(GCN)

$\rightarrow$ 对图结构进行攻击

$\rightarrow$ 对图特征进行攻击

具体细节与伪代码请见算法1.

4. 数值结果 (Numerical Result)

图 4. 6月14日18:00得分与排名

因为数据集不同, 其余图数据攻击的Baseline无法处理大数据集, 所以无法比较.

5. 讨论

”咱们做科研要有可解释性.”
一位NLP学长如是说.

很多人很疑惑为什么Meta-Learning, 一个学习如何学习的方法怎么可以用在这里. 其实使用Meta-Learning只是使用到其处理二层优化时Meta-Gradient的方法, 我希望得到的是仅对局部图的学习就可以得到其他图的攻击策略, 如果要化归到”学习如何学习”的思想上, 那么图G不能被看成输入, 而应该被看成模型的一部分.

如果仔细思考, 就会感受到这种方法攻击优化得到的图是与模型无关的, 攻击的优化方向不是Model Pre-training那种随着当前模型参数梯度更新的, 而是更新了许多步之后所有

Algorithm 1: 图数据攻击

Data: 图邻接矩阵G, 特征矩阵$\boldsymbol{X}$, 被攻击的替代模型f

Result: 攻击后的邻接, 特征矩阵 $\hat{G}, \hat{\boldsymbol{X}}$.

随机加入攻击结点、特征, 开始对 $\hat{G}$ 进行结构攻击 ;

初始化结构攻击迭代次数 attack-iter, Meta-Learning训练迭代次数 train-iter. ;

while *done all attack-iter* **do**

  for *all* $\mathcal{T}_i$ *(每个块))* **do**

    构成数据集$\mathcal{D}_K$;

    for *all train-iter* **do**

      梯度下降更新模型参数 θ ;

      $\theta_i^{\star} = \theta - \alpha \nabla_{\theta} l_{\mathcal{T}_i}\left(f_{\theta}\right)$

    end

    根据当前$f_{\theta^{\star}}$, $\hat{G}$ 和公式2 计算 $\nabla_{\hat{G}} l_{\mathcal{T}_i}\left(f_{\theta_i^{\star}}(\hat{G})\right)$ 并保存 ;

  end

  $\hat{G} \leftarrow \hat{G} - \beta \cdot \sum_{\mathcal{T}_i \in \mathcal{T}} \nabla_{\hat{G}} l_{\mathcal{T}_i}\left(f_{\theta_i^{\star}}(\hat{G})\right)$

  贪心处理的求导后离散化 ;

end

– – –

开始对特征进行攻击 ;

初始化特征攻击迭代次数 attack-iter ;

while *done all attack-iter* **do**

  $\mathcal{L}$ 为替代模型前向传播的损失 ;

  $\hat{G}, \boldsymbol{X}$ 输入网络, 并计算$\nabla_{\boldsymbol{X}} \mathcal{L}$.

  for *all* i, j **do**

    if $\boldsymbol{X_{ij}} \in [lb, ub]$ **then**

      $\boldsymbol{X_{ij}} \leftarrow \boldsymbol{X_{ij}} + \epsilon \cdot \operatorname{sign}\left(\nabla_{\boldsymbol{X_{ij}}} \mathcal{L}\right)$

    end

  end

end

return $\hat{G}, \boldsymbol{X}$

块的方向总和, 如图5, 这也解释了为什么攻击策略能在新的图上有更好的效果.

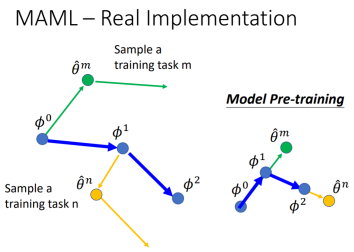

图 5. MAML与Model Pre-training的区别

一点遗憾, 在比赛末期才发现图的攻击和特征的攻击要交叉训练效果才会更好.

6. 结论 (Conclusion)

我提出了一种将Meta-Gradient与FGSM结合的图数据攻击方法: HMG. 重点解决了图数据攻击时的二层优化问题, 在KDD Cup 2020中取得了较好的成绩, 进一步我讨论了该方法降低图深度学习模型准确率的原因, 未来还可以深入研究如何增强深度学习模型的健壮性. 提出鲁棒的防御模型.

7. 致谢 (Acknowledgements)

有很多的材料都给了我很多帮助, 例如: [3], [4], [6], [7] 等.

三个感谢:

- 特别感谢原先和我组队的兄弟: 陈善梁, 林昊. 在前期他们给予我很多支持, 他们离队的原因有很大部分是因为我太独了, 我当时想都是兄弟都我做也没关系, 但是显然他们都不是划水的, 看我做第三题做得很起劲也没有和我说就去做第一题了, 后来即便第三题我做出了成绩他们也离队去做第一题了, 我非常敬佩他们, 真的虽然很震

惊很无奈, 但是这种踏实的做事风格值得我学习, 非常感谢他们.

- 感谢全体助教哥, 你们辛苦了, 感谢马兰霁弘和谭鹏学长在第三题上给予我的帮助.
- 感谢叶翰嘉, 周志华, 詹德川老师, 他们教授机器学习课程, 非常感谢.

三个展望:

- 祝愿助教哥们paper一投就中.
- 祝愿老师们身体健康, 万事如意.
- 不要脸地祝愿我机器学习拿到高分吧, 谢谢!

参考文献

[1] Model-agnostic meta-learning for fast adaptation of deep networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[2] *biendata KDD Cup*, 2020.

[3] Y. Bengio. Gradient-based optimization of hyperparameters. *Neural Computation*, 12(8):1889–1900, 2000.

[4] W. Jin, Y. Ma, X. Liu, X. Tang, S. Wang, and J. Tang. Graph structure learning for robust graph neural networks, 2020.

[5] A. Nichol, J. Achiam, and J. Schulman. On first-order meta-learning algorithms. *CoRR*, abs/1803.02999, 2018.

[6] L. Sun, Y. Dou, C. Yang, J. Wang, P. S. Yu, and B. Li. Adversarial attack and defense on graph data: A survey, 2018.

[7] H. Wang, J. Wang, J. Wang, M. Zhao, W. Zhang, F. Zhang, X. Xie, and M. Guo. Graphgan: Graph representation learning with generative adversarial nets, 2017.

[8] D. Zügner and S. Günnemann. Adversarial attacks on graph neural networks via meta learning. 2019.