对抗训练在 NLP 中的应用实验报告

1. 背景

GAN 之父 Ian Goodfellow 在 15 年的 ICLR[1]第一次提出了对抗训练这个概念，简而言之，就是在原始输入样本上加一个扰动，得到对抗样本后，用其进行训练，提升模型的训练效果。为将其迁移到 NLP 任务中，Goodfellow 在 17 年的 ICLR[2]中提出了可以在连续的 embedding 上做扰动。本报告首先简单的介绍了一下什么是对抗训练，然后介绍一下常见的几种方法，最后在文本分类模型 TextCNN[3]的基础上实现了 FGSM[1]、FGM[2]、PGD[4]和 Free[5]这几种对抗训练的方法，并比较和分析实验结果。

2. 什么是对抗训练

对抗训练是一种通过对原始样本添加扰动构造一些对抗样本，让模型去训练，提高模型在这些对抗样本上的分类能力，同时一定程度上也能提升模型的表现和泛化能力。

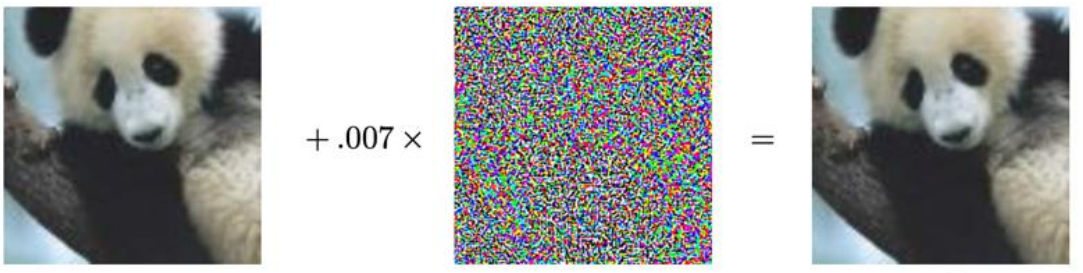
对于一个样本来说，其分类结果是由深度网络中大量参数和激活函数的形式所决定的，如果以某种方式改变样本使得激活函数朝着反方向变化，那么这种变化会形成“雪球效应”使得分类器改变最终的分类概率。在深度网络中，loss 间接的反映了分类结果的好坏，如果对抗样本使得函数的 loss 增大以至于分类结果出错，那么就算攻击成功了。

根据以上思路，Madry 在 2018 年的 ICLR[4]中总结了之前的工作，并从优化的视角，将问题重新定义成了先优化找最大值再找最小的问题，也被称为 Min-Max 公式：

$$\min_{\theta} \rho(\theta), \quad where \quad \rho(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\max_{\delta\in\mathcal{S}} L(x+\delta, y; \theta)]$$

该公式分为两个部分，一个是内部损失函数的最大化，一个是外部经验风险的最小化。因此问题也主要集中在两个方面，如何构造干扰性强的对抗样本使模型犯错和如何更新参数增强模型的表达能力。

对于图像类的数据只需要在原始的像素数据上通过某种方式增加一个很小的扰动，就可以产生有效的攻击，如下图所示。



而对于文本类的数据，一般编码成 index 的形式，是离散的，所以无法的原始数据上

进行扰动，而 word embedding 是连续的，所以可以在 word embedding 上扰动。不过 word embedding 也是参数，是通过学习出来的，因此文本上的对抗训练可以看作是一种正则化的手段，能够使得 word embedding 的质量更好，避免过拟合，从而取得出色的表现。

3. 常用的对抗训练的方法

- **FGSM (Fast Gradient Sign Method)**[1]

FGSM 是 Goodfellow 提出的对抗训练时方法，假设当前输入的梯度为：

$$g = \Delta_x L(x, y; \theta)$$

那么扰动值和对抗样本定义为: $\delta = \varepsilon * sign(g)$，$\tilde{x} = x + \delta$。可以理解为将输入样本向着梯度的方向增加，这样得到的对抗样本就能造成损失的增加，从而促进模型更进一步的学习。

- **FGM (Fast Gradient Method)**[2]

FGSM 在每个方向上都走相同的一步，Goodfellow 后续对 FGSM 改进，提出了 FGM，根据具体的梯度大小进行 scale，类似于加了学习率的梯度上升法：

$$\delta = \varepsilon * \frac{g}{||g||_2}$$

- **PGD (Projected Gradient Descent)**[4]

PGD 可以看作是对于 FGSM 或者 FGM 的近一步改进，FGSM 直接通过ε参数只经过了一步算出了扰动值，这样得到的扰动可能不是最优的。PGD 进行了改进，多迭代几次，慢慢找到最优的扰动值，具体的迭代公式：

$$\delta_{t+1} = \alpha * \frac{g_t}{||g_t||_2}$$，α为迭代的步长，且$||\delta_t||_2 \leq \varepsilon$

- **Free (Free Adversarial Training)**[5]

从 FGSM 到 PGD，主要是优化对抗扰动的计算，虽然取得了更好的效果，但计算量也一步步增加。对于每个样本，FGSM 或 FGM 都是两次前后向的计算，一次是原始样本$\boldsymbol{x}$的，另一次是对抗样本$\mathbf{x} + \delta$的。而 PGD 则计算了$K + 1$次，消耗了更多的计算资源。因此 Free 在 PGD 的基础上进行了训练速度的优化。

Free 的思想是在对每个样本$\boldsymbol{x}$连续重复M次训练，更新方式上和 FGSM 比较像，不过在计算δ时时复用了上一步的梯度，又和 PGD 一样，整体训练的 epoch 相当于乘以了M。δ的更新公式为：

$$\delta_{t+1} = \delta_t + \varepsilon * sign(g)$$

4. 对抗训练实验和效果分析

- **实验设置**

本实验以 TextCNN 为 Baseline 的模型，在此基础上增加对抗样本，比较对抗训练方法的准确率、召回率和 F1 值（计算方法见：sklearn）。

TextCNN 的代码来源于 github 项目 Chinese-Text-Classification-Pytorch，TextCNN 的原理可以参见作者的博客：中文文本分类 pytorch 实现。

对抗训练的代码参考知乎：NLP 中的对抗训练+PyTorch 实现 ，同时参考 github 项目 fast_adversarial 的实现思路。

我的实验代码见：TextCNN-Adversarial-Training-in-NLP，具体实验的执行方式见项目 ReadMe。

训练数据集来源于上述 TextCNN 作者从 THUCNews 中抽取了 20 万条新闻标题，一共 10 个类别，每类 2 万条，文本长度在 20 到 30 之间。

- **实验结果**

指标数据：

方法	acc	micro-precison	micro-recall	micro-f1
Baseline	89.91%	0.8993	0.8991	0.8991
FGSM	91.37%	0.9138	0.9137	0.9136
FGM	89.96%	0.9004	0.8996	0.8997
PGD	90.12%	0.9015	0.9012	0.9009
Free	89.64%	0.8967	0.8964	0.8964

性能数据：

方法	训练时间	stop epoch	每个 epoch 时间	Test loss	参数配置
Baseline	23.5 分钟	4	5.8 分钟	0.34	
FGSM	107 分钟	8	13.3 分钟	0.29	$\varepsilon = 0.1$
FGM	32 分钟	4	8 分钟	0.33	$\varepsilon = 0.1$
PGD	100 分钟	4	25 分钟	0.33	$\varepsilon = 0.1, K = 3, \alpha = 0.1$
Free	95 分钟	3	31.6 分钟	0.38	$\varepsilon = 0.1, M = 3$

注：(1)各方法的参数配置见 models；(2)上述详细的实验指标见 log。

- **数据分析**

✓ 从实验指标看，FGSM 方法的指标是最好的，有一个可能的解释是 TextCNN 模型结构太简单了，太复杂的方法反而不会带来提升；FGM 方法理论上会比 FGSM 方法好一点，不过需要仔细的调整ε的值；

✓ FGSM 有一个ε参数可调，还尝试过$\varepsilon = 0.05$和$\varepsilon = 0.01$，acc 分别为 91.15%和

90.47%，指标均比$\varepsilon = 0.1$时差，可以调大了再试试；

✓ PGD 方法因为训练速度比较慢，而且可以调的参数比较多，因此没有尝试太多组参数，多尝试几组应该还会有收益；

✓ Free 方法由于每个样本需要连续的更新 M 次，所以整体的 epoch 是最多的，但是效果却是最差的。除了上述第一条原因外，Free 也有自己的缺点，Free 每次的扰动都是根据前一次样本的梯度计算出来的，对于当前样本不一定是最优的，后续还有 FreeLB 方法对其进行改进。

- **后续展望**

✓ 这几种对抗训练的方法还不少参数可以调，后续时间充分可以进一步调整，上述结果已经初步证明了对抗训练在 NLP 中的效果；模型本身也有一些参数需要配合对抗训练去调整的，比如学习率、dropout 等；

✓ 后续可以把 TextCNN 换成 Bert 等复杂的模型，增大模型的复杂度，让模型有更多的空间可以学习；

✓ 以上几种方法有各自的缺点，后续可以尝试其他类 PGD 的改进方法，比如 FreeLB、YOPO、SMART 等方法；

5. 参考

[1] Explaining and Harnessing Adversarial Examples

[2] Adversarial Training Methods for Semi-Supervised Text Classification

[3] Convolutional Neural Networks for Sentence Classification

[4] Towards Deep Learning Models Resistant to Adversarial Attacks

[5] Adversarial Training for Free!