Analysis and Utilization of Hidden Information in Model Inversion Attacks

蒋心雨

摘要

深度学习的广泛应用引起了人们对深度神经网络隐私问题的关注。模型反演攻击的目的是从给定的神经网络中重构每个私有训练样本的具体细节。然而，受限于有用信息的可用性，重构有特色的私人训练样本还有很长的路要走。本文采用信息熵的方法，研究了重构特殊私有训练样本的需求。我们发现需要更多的信息来重建不同的样本，并提出利用经常被忽略的隐藏信息来实现这一目标。为了更好地利用这些信息，我们提出了 Amplified-MIA。在 Amplified-MIA 中，在目标网络和攻击网络之间插入一个非线性放大层。这个非线性放大层还包含一个非线性放大函数。给出了非线性放大函数的定义，并推导了非线性放大函数对隐信息熵的影响。提出的非线性放大函数可以放大较小的预测向量条目，放大同一类别中不同预测向量之间的差异。这样，攻击网络可以更好地利用隐藏的信息，重构出有特色的私有样本。通过各种实验，实证分析了非线性放大函数对重建结果的影响。在三个不同的数据集上的重建结果表明，所提出的 Amplified-MIA 在几乎所有任务上都优于现有的工作。特别是在最困难的人脸重建任务上，与直接反演相比，该方法的像素精度得分提高了 68%。

关键词：深度学习；安全；隐私；深度神经网络；模型反演攻击

1 引言

通过将一个图像的预测向量经过放大函数后输入反演模型，进而反演得到原始输入图像。放大层中的放大函数放大该向量中的小条目，同时轻微放大大条目，能够恢复同一类别中不同预测向量之间的差异，实现了攻击模型对目标模型输出的预测向量恢复不同的私人训练图像的功能。

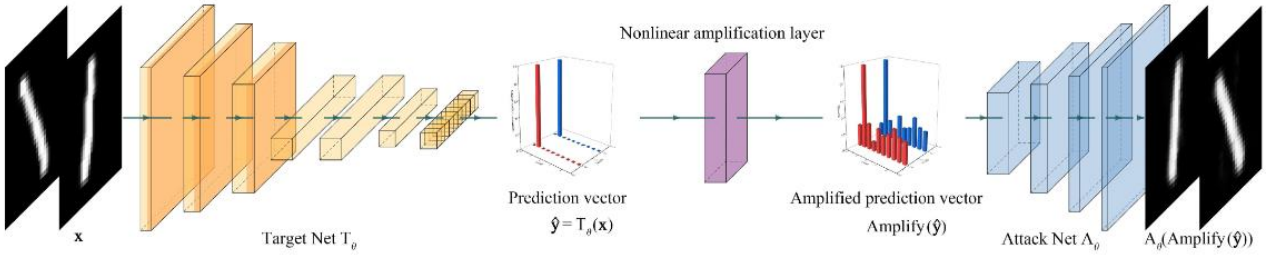

图 1. Amplified - MIA 的架构概述。在 Target Net 和 Attack Net 之间插入包含非线性放大函数的非线性放大层。利用这种非线性放大函数，可以对小的预测向量条目进行放大，并且可以重构出具有区分性的样本。

2 论文创新点

1、分析了使用信息熵重构具有区分性的私有样本的要求，并将预测向量中的信息分为类别信息和隐藏信息。基于需要更多的信息来重建有特色的样本的观察，提出利用很少使用的隐藏信息来实现这一目标。

2、在目标网络和攻击网络之间引入非线性放大层。非线性放大层包含一个非线性放大函数，可以对预测向量中的小项进行放大。将这一框架称为 Amplified-MIA。同时，给出了非线性放大函数的定义，并讨论了其条件对重建结果的影响。

3、发现函数 $y = x^{\alpha}$，$0 < \alpha < 1$，满足非线性放大函数的定义。还推导了这种非线性放大函数对隐藏信息的影响。通过优化 α，达到最优的重建质量。并实证分析了非线性放大函数和目标网络深度对重建结果的影响。还将 Amplified - MIA 的重建结果与无 softmax 层的直接反演结果进行了比较。

4、在 3 个著名的数据集上进行了多种实验，结果表明 Amplified - MIA 在 MNIST 重建任务上能够利用大约 52 比特的额外信息。其他实验表明，Amplified - MIA 在几乎所有任务上都优于之前的工作，在重建图像的像素精度评分上比直接反演方法提高了 68 %，在最难的人脸重建任务上比 Yang 等人的工作提高了 26 %。

3 本文方法

3.1 放大函数

本文所使用的放大函数定义如下：

Definition 1: Let $f : [0,1] \rightarrow [0,1]$ be a twice-differentiable function. f is called a nonlinear amplification function if it satisfies the following conditions:

- $x \leq f(x)$.
- $f(0) = 0, f(1) = 1$.
- if $x < y$, then $f(x) < f(y)$, i.e., $f(x)$ is monotonically increasing.
- $f(x)$ is concave down.

- 保证条目数值放大。
- 临界值 0 和 1 的分类值保持不变。
- 函数单调递增，保证放大前后预测向量各个条目之间的关系不变。
- 函数向下凹，且在 x=0 处的斜率大，使得小条目尽可能放大，大条目稍微放大。

3.2 理论分析

机器学习的特性使得模型会增加一个输入属于目标类的可能性，同时减小属于其他类别的可能性，因此同一类别不同预测向量之间的差别减小，隐藏信息减少。而预测向量信息=分类信息+隐藏信息，其中分类信息表示预测向量的分类结果，隐藏信息表示分类结果为同一类的不同预测向量之间的差别。

为了恢复隐藏信息，将一个目标模型的预测向量通过非线性放大层，该层包含一个非线性放大函数，能够放大该向量中的小条目同时轻微放大大条目，恢复同一类别中不同预测向量之间的差异。因此，如果计算出当放大前后的信息熵差最大时的α值，就使得攻击模型能够尽可能恢复不同的私人训练图像。

将一张图片输入分类器获得预测向量，对于每个类别 c，设 ic 表示样本属于类别 c 的概率，ic 满足 ic∈[0,1]，且

$$\sum_{c=0}^{N-1} i_c = 1$$

设 Ic 是一个随机变量，其累积分布函数 F(ic)=Pr(Ic≤ic)。令 F(ic)的导数 F'(ic)= f(ic)，那么 f(ic)为 Ic 的概率密度函数，因此随机变量 Ic 的微分熵 h(Ic)和预测向量中隐藏信息的微分熵 h_{hidden} 分别为

$$h\left(I_c\right) = -\int_0^1 f\left(i_c\right) \log_2 f\left(i_c\right) di_c$$

$$h_{hidden} = \sum_{c=0}^{N-1} h\left(I_c\right)$$

预测向量放大后的隐藏信息微分熵如下：

$$h_{hidden}^{amp} = h_{hidden} - \left((1-\alpha) \sum_{c=0}^{N-1} \log \xi_c + N \log \frac{1}{\alpha} \right)$$

解得当放大后隐藏信息微分熵最大时的 α 为：

$$\alpha = -\frac{N}{\ln 2 \sum_{c=0}^{N-1} \log \xi_c}$$

4 复现细节

4.1 与已有开源代码对比

本文代码开源，github 链接为 https://github.com/zhangzp9970/Amplified-MIA，在整理数据集之后，使用 Amplified-MIA 模型处理数据，完成了对 MNIST 与 FaceScrub 人脸数据集的反演攻击。

4.2 实验环境搭建

代码运行环境： Linux-18.04-Ubuntu； Python 版本: 3.8; 相关依赖： anaconda、pytorch-cuda=11.8、zhangzp9970；

4.3 代码实现

4.3.1 反演 MINIST 数据集代码

```
for i, (im, label) in enumerate(tqdm(test_dl, desc=f"epoch {epoch_id}")):
    im = im.to(output_device)
    label = label.to(output_device)
    bs, c, h, w = im.shape
    optimizer.zero_grad()
    out = target_classifier.forward(im)
    #计算类别概率分布
    after_softmax = F.softmax(out, dim=-1)
    #一个向量为【0.1，0.9】，将返回【pow(0.1,1 / 11),pow(0.9,1 / 11)】，这是放大后的向量
    after_softmax = target_amplification.forward(after_softmax)
    # 10->512->1->sigmoid()
    rim = myinversion.forward(after_softmax)
    #一批数据,图像之间均方差的均值
    # MSE越小，两张图像在像素级别上的相似度越高
    mse = F.mse_loss(rim, im)
    loss = mse
    loss.backward()
    optimizer.step()
```

图 3. 反演 MINIST 数据集代码

4.3.2 反演 FaceScrub 人脸数据集代码

```
for epoch_id in tqdm(range(1, train_epoches + 1), desc="Total Epoch"):
    for i, (im, label) in enumerate(tqdm(test_dl, desc=f"epoch {epoch_id}")):
        im = im.to(output_device)
        label = label.to(output_device)
        bs, c, h, w = im.shape
        optimizer.zero_grad()
        out = target_classifier.forward(im)
        after_softmax = F.softmax(out, dim=-1)
        after_softmax = target_amplification.forward(after_softmax)
        rim = myinversion.forward(after_softmax)
        mse = F.mse_loss(rim, im)
        loss = mse
        loss.backward()
        optimizer.step()

    if epoch_id % log_epoch == 0:
        writer.add_scalar( tag: "loss", loss, epoch_id)
        writer.add_scalar( tag: "mse", mse, epoch_id)
        save_image2(im.detach(),  fp: f"{log_dir}/input/{epoch_id}.png")
        save_image2(rim.detach(),  fp: f"{log_dir}/output/{epoch_id}.png")
        with open(
            os.path.join(model_dir, f"myinversion_{epoch_id}.pkl"), "wb"
        ) as f:
            torch.save(myinversion.state_dict(), f)
```

图 4. 反演 FaceScrub 人脸数据集代码

4.4 反演 MNIST 数据集结果

数据准备：目标模型的训练数据是 MNIST 训练集，攻击模型的训练数据是 MNIST 测试集。

图 5. 反演 MNIST 数据集对比结果

实验结果说明：图 5 左 1 是训练 MNIST 数字分类器的私有数据，左 2 是论文相应的反演结果，右 1 是复现论文的结果。可以看到反演 MNIST 数据集的效果较好，能较准确得反演出数字的特征。

4.5 攻击 FaceScrub 人脸数据集结果

数据准备：整合 FaceScrub 数据集中男、女演员的裁剪后人脸数据。目标模型的训练数据是 6/7 的 FaceScrub 数据集，攻击模型的训练数据是 1/7 的 FaceScrub 数据集。

图 6. 反演 FaceScrub 人脸数据集对比结果

实验结果说明：图 6 左边两张图片分别是训练 FaceScrub 人脸分类器的私有数据和论文相应的反演结果。右边两张图分别是训练 FaceScrub 人脸分类器的私有数据和复现论文的结果。可以看到反演 FaceScrub 数据集的效果相对差一些，反演结果具有平均的特征。

5 实验结果分析

1、反演 MNIST 数据集的结果显著优于反演 FaceScrub 数据集的结果，这是因为人脸图片具有更多特征，反演难度更难。

2、在反演人脸数据集时，反演每个不同的个体后，重构的图像却很相似，这是因为攻击模型的训练数据不包含私有数据，所以重建私有数据的时候，只能利用辅助数据的特征去合成私有数据特征，这就存在一个问题，就是重构出来的图像是辅助数据的平均效果。

3、后续希望能够利用辅助数据在目标分类器中的分类结果去引导攻击模型生成具有个体特征的数据，提升反演效果，提高攻击性能。

6 总结与展望

模型反演攻击的目的是从给定的神经网络中重构每个私有训练样本的具体细节。例如将反演攻击应用于人脸识别模型上，通过攻击目标模型从而反演出训练目标模型的数据，使得用户的数据泄露，导致用户隐私安全问题。本文通过将一个图像的预测向量经过放大函数后输入反演模型，进而反演得到原始输入图像。利用该非线性放大函数，可以放大预测向量中的小条目，扩大同一类预测向量之间的差异。因此，攻击网络可以很容易地从这些放大的预测向量中学习隐藏信息并重建不同的样本，从而实现了攻击模型对目标模型输出的预测向量恢复不同的私人训练图像的功能。

参考文献

[1] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.

[2] Microsoft. (2021). Azure Machine Learning—ML as a Service. [Online]. Available: https://azure.microsoft.com/en-us/services/machine-learning/#product-overview

[3] Amazon. (2021). Machine Learning Image and Video Analysis—Amazon Rekognition—Amazon Web Services. [Online]. Available: https://aws. amazon.com/cn/rekognition/?blog-cards.sort-by=item.additionalFields. createdDate&blog-cards.sort-order=desc

[4] B. Biggio and F. Roli, “Wild patterns: Ten years after the rise of adversarial machine learning,” Pattern Recognit., vol. 84, pp. 317–331, Dec. 2018, doi: 10.1016/j.patcog.2018.07.023.

[5] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, “Membership inference attacks against machine learning models,” in Proc. IEEE Symp. Secur. Privacy (SP), May 2017, pp. 3–18.

[6] E. Tabassi, K. J. Burns, M. Hadjimichael, A. D. Molina-Markham, and J. T. Sexton, “A taxonomy and terminology of adversarial machine learning,” NIST, Gaithersburg, MD, USA, Tech. Rep. NISTIR 8269, 2019, doi: 10.6028/nist.ir.8269-draft.

[7] C. Szegedy et al., “Intriguing properties of neural networks,” 2014, arXiv:1312.6199.

[8] Y. Alufaisan, M. Kantarcioglu, and Y. Zhou, “Robust transparency against model inversion attacks,” IEEE Trans. Dependable Secure Comput., vol. 18, no. 5, pp. 2061–2073, Sep. 2021.

[9] S. Hidano, T. Murakami, S. Katsumata, S. Kiyomoto, and G. Hanaoka, “Model inversion attacks for prediction systems: Without knowledge of non-sensitive attributes,” in Proc. 15th Annu. Conf. Privacy, Secur. Trust (PST), Aug. 2017, pp. 11500–11509.

[10] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing," in Proc. 23rd Secur. Symp. (USENIX Security). San Diego, CA: USENIX Association, Aug. 2014, pp. 17–32. [Online]. Available: https://www.usenix.org/conference/usenixsecurity14/technical-sessions /presentation/fredrikson_matthew

[11] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur. New York, NY, USA: ACM, Oct. 2015, pp. 1322–1333, doi: 10.1145/2810103.2813677.

[12] X. Wu, M. Fredrikson, S. Jha, and J. F. Naughton, "A methodology for formalizing model-inversion at tacks," in Proc. IEEE 29th Comput. Secur. Found. Symp. (CSF), Jun. 2016, pp. 355–370.

[13] P. Prakash, J. Ding, H. Li, S. M. Errapotu, Q. Pei, and M. Pan, "Privacy preserving facial recognition against model inversion attacks," in Proc. IEEE Global Commun. Conf. (GLOBECOM), Dec. 2020, pp. 1–6.

[14] T. Wang, Y. Zhang, and R. Jia, "Improving robustness to model inversion attacks via mutual information regularization," in Proc. AAAI Conf. Artif. Intell., 2020, pp. 11666–11673.

[15] Z. Yang, J. Zhang, E.-C. Chang, and Z. Liang, "Neural network inversion in adversarial setting via back ground knowledge alignment," in Proc. ACM SIGSAC Conf. Comput. Commun. Secur. New York, NY, USA: ACM, Nov. 2019, pp. 225–240, doi: 10.1145/3319535.3354261.

[16] D. Yang, S. Hong, Y. Jang, T. Zhao, and H. Lee, "Diversity-sensitive conditional generative adversarial networks," 2019, arXiv:1901.09024.

[17] Y. Zhang, R. Jia, H. Pei, W. Wang, B. Li, and D. Song, "The secret revealer: Generative model-inversion attacks against deep neural networks," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 250–258.

[18] I. J. Goodfellow et al., "Generative adversarial nets," in Proc. 27th Int. Conf. Neural Inf. Process. Syst. Cambridge, MA, USA: MIT Press, 2014, pp. 2672–2680.

[19] S. Chen, M. Kahla, R. Jia, and G.-J. Qi, "Knowledge-enriched distributional model inversion attacks," in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 16158–16167.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Feb. 2016, pp. 770–778.

[21] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 3rd ed. Upper Saddle River, NJ, USA: Pren tice-Hall, 2006. [22] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based

learning applied to document recognition," Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, doi: 10.1109/5.726791.