L-008

深層ニューラルネットワークに対する予備訓練なしのホワイトボックス電子透かし：モデル構造を変更させない AI 著作権への一つの試み

Non-Invasive White-Box Watermarking for Deep Neural Networks: An Approach to AI Authorship Without Altering Model Structure

陳 昱璋 [1)]　朱 江楠 [1)]　顧 玉杰 [1)]　櫻井 幸一 [1)2)]
Yuzhang Chen　Jiangnan Zhu　Yujie Gu　Kouichi Sakurai

1　はじめに

深層ニューラルネットワーク（DNN）は画像認識、自然言語処理、医療診断や自動運転などの分野で重要性を増しており，精度の高い DNN モデルは開発コストが高いため非常に価値がある. そのため，DNN モデルの知的財産を保護することが重要とされ，2017 年に内田らによって提案された DNN 電子透かし技術がこれを可能にする．この技術は，モデル所有者がモデルに電子透かし情報を挿入し，抽出することで所有権を証明する方法である.DNN の発展とともに，電子透かし技術は注目され，知的財産保護の主要な手段となりつつある．

ただし，既存の DNN 電子透かし技術には，ホストモデルを改変する必要があるという大きな欠点がある．この改変はモデルのパフォーマンスに影響を与え，その結果，特に医療診断や自動運転のような分野での電子透かし技術の応用が制限されている．

本論文では，モデルの固有の特徴に基づいた新しい電子透かし技術を提案する．モデル所有者がモデルを変更することなく事前に設定された電子透かしを抽出できるようにする. 我々の方法には二つの主要な特徴がある：

1. ホストモデルを変更しない.
2. 高い電子透かし容量を維持しつつ，主流の電子透かし攻撃に対抗する．

2　DNN 透かしのレビュー

ここでは，DNN 電子透かしの定義と既存の方法の不足点を説明する．

2.1　DNN 電子透かしの定義

DNN 電子透かしは，次の四つの要素 (**WGen**, **WEmb**, **WExt**, **WVer**) で構成される：

- **WGen** はランダムに電子透かし $\mathbf{b} \in \{0,1\}^N$ を生成するアルゴリズムである；
- **WEmb** は，事前に指定された $\mathbf{b}$，ホストモデル H，特定の訓練データ集 D を入力して，秘密鍵集 $\mathcal{K}$ とともに電子透かしを埋め込んだモデル $\tilde{H}$ を出力するアルゴリズである；
- **WExt** は，秘密鍵集 $\mathcal{K}$，モデル $\tilde{H}$，訓練データ集 T を入力として，取り出した電子透かし $\hat{\mathbf{b}}$ を出力する電子透かし抽出アルゴリズムである；
- **WVer** は，指定された $\mathbf{b}$ と取り出した $\hat{\mathbf{b}}$ を比較して検証する．検証が成功した場合に 1 を出力し，そうでない場合に 0 を出力するメカニズムである．

2.2　既存提案の不足

最初の DNN 電子透かし技術は内田らによって提案された [2]. 彼らはモデルの元の損失関数に電子透かしの抽出のための追加の損失関数を加えた．このアプローチにより，モデルはパラメータを適切に修正することで電子透かし情報を抽出できるようになった．電子透かし技術が発展するにつれて，ホワイトボックス電子透かしとブラックボックス電子透かしの二つに分岐した．ホワイトボックス電子透かしは，電子透かしを挿入する際にホストモデル H の内部構造を完全に透明にするものであり [3, 2]，一方，ブラックボックス電子透かしはその逆である．主流のブラックボックス電子透かしは，バックドア攻撃を使用してモデル H に電子透かしを挿入し，特別なトリガー T によってそれらを抽出する [1].

実際のところ，ホストモデル H に関する情報が多いため，ホワイトボックス電子透かしはより効果的であると考えられている [4]. しかし，ホワイトボックスとブラックボックスの両方の電子透かしには共通の特徴がある：電子透かしの挿入はホストモデル H の修正を伴う．

3　予備訓練なし電子透かし

本論文は電子透かしによるモデル精度の低下問題に着目し，予備訓練なしのホワイトボックス電子透かし方法を提案した．これから提案方法を電子透かしの埋め込みと電子透かしの取り出しに分けて説明する．

3.1　電子透かしの埋め込み

ニューロンの発火値はモデルの固有の特徴として，ホストモデルと非ホストモデルを効果的に区別できるため，私たちは発火値を利用して電子透かしを挿入する．同時に，モデル所有者が電子透かしを抽出するために二つの秘密鍵を生成する．まず，電子透かし情報を定義する：私たちの方法では，電子透かし情報は二進ベクトル $b \in \{0,1\}^N$ で，ここで N はベクトルの長さを表す.

次に，目標中間層 l のニューロンの数を M とし，訓練データと対応するラベルをそれぞれ X^{train} と Y^{train} と表記して，訓練データ集を $T = \{X^{\text{train}}, Y^{\text{train}}\}$ とする．また，ラベル Y^{train} の下での訓練データ X^{train} の標本数を s と設定する．ここでは，T がすべてのラベルのデータを均等に含むことを要求する．

ホワイトボックスの場合，モデルが T にある i^{th} 標本を与えられた時に，目標中間層にあるすべてのニューロンの発火値を取得できる．これらの発火値はベクトル $f_l^i(M \times 1)$ で表される.T のすべての標本を使用すると，発火値の行列 $f(M \times s)$ を得る．この行列では，各列が T の特定の標本における中間層 l のすべてのニューロンの活性値を表し，各行は l の特定のニューロンが T にあるすべての標本にわたる時の発火値を表す．今，この行列の各行の平均を取ると，平均発火値ベクトル $f_l(M \times 1)$ を得ることができる．ここで，i^{th} 要素は中間層 l の i^{th} ニューロンが T にあるすべての標本にわたる時の平均発火値を表す.

1）九州大学 大学院システム情報科学研究院
2）情報処理学会，電子情報通信学会

次に，$b \in \{0,1\}^N$ と $f_l(M \times 1)$ を利用し，以下の二つの式を用いて二つの秘密鍵，$A(N \times M)$ と $D(M \times 1)$ を生成する．

$$\text{Sigmoid}(x) = \frac{1}{1+e^{-x}} \tag{1}$$

$$\text{Thresholding}(x) = \begin{cases} 0 & \text{if } x < 0.5 \\ 1 & \text{if } x \geq 0.5 \end{cases} \tag{2}$$

秘密鍵生成は具体的に三つのステップに分けられる：

1. 標準正規分布を利用して，長さ M のベクトル $\mu(M \times 1)$ を生成する．
2. 勾配降下法を利用して，N のサイズの行列 $A(N \times M)$ を生成し，秘密鍵 A が
$$b = \text{Thresholding}[\text{Sigmoid}(\mu \times A)] \tag{3}$$
を満たすようにする．
3. f_l と μ を利用して，秘密鍵 $D(M \times 1)$ を生成する：$D = \mu - f_l$.

ここで $D = \mu - f_l$ を変形して $\mu = D + f_l$ とすると，

$$b = \text{Thresholding}[\text{Sigmoid}((D + f_l) \times A)] \tag{4}$$

が成立することが得られる．これによりモデルを修正することなく，二つの秘密鍵を利用して事前指定された電子透かし情報を抽出できて，電子透かし埋め込みの部分は終了する．

3.2 電子透かしの取り出し

電子透かしの埋め込み段階で二つの秘密鍵を取得した後，電子透かしの取り出しにも以下の三つのステップがある：

1. 訓練データ集 T のデータをモデルに入力し，目標中間層の平均発火値ベクトル $f_l(M \times 1)$ を取得する．
2. 二つの秘密鍵 $A(N \times M)$ と $D(M \times 1)$ 利用し，以下の式
$$\hat{b} = \text{Thresholding}[\text{Sigmoid}((D + f_l) \times A)] \tag{5}$$
によって抽出された電子透かし情報 $\hat{b}(N \times 1)$ を計算する．
3. b と $\hat{b}$ の間の BER（ビットエラーレート）を計算する.BER の定義は以下です：
$$\text{BER} \triangleq \frac{\text{HD}(b(N \times 1), \hat{b}(N \times 1))}{N} \tag{6}$$

ここでの HD はハミング距離を示し，適切な閾値 Θ を設定する必要がある.BER が Θ より小さい場合は電子透かしの抽出に成功したと見なされ，逆の場合は電子透かしの抽出に失敗したと見なされる．これにより電子透かし取り出しの部分は終了する．

4 実験の検証

実験では，Lenet（Minist），VGG16（CIFAR10），Resnet20（CIFAR100）の三つのモデルとそれに対応するデータセットを使用し，電子透かしの長さ $N = 512$，閾値 0.1 と設定して提案方法が

1. 埋め込んだ電子透かしを正しく取り出せることを検証する．
2. 電子透かし攻撃を対抗できることを検証する．

4.1 正しく電子透かしを取り出すの検証

表 1 の結果に基づき，私たちの方法は事前に指定された電子透かし情報を間違いなく抽出できる．

モデル	Lenet5	VGG16	Resnet20
ACC	0.9835	0.8252	0.579200
BER	0.0	0.0	0.0

表 1　モデルによる ACC と BER の結果

次に，標準正規分布を使用してランダムに二つの秘密鍵生成し，電子透かしを試みて抽出する．

モデル	Lenet5	VGG16	Resnet20
BER	0.4463417968	0.48828125	0.5406347656

表 2　間違う秘密鍵を使用した BER の値

表 2 の結果によると，秘密鍵を知らない状態では電子透かしを正しく抽出することができない．

4.2 攻撃に対抗できる (robustness) の検証

ここでは，モデル微調整攻撃とモデル剪定攻撃の二つの主流攻撃を使用して検証を行う．

微調整攻撃に対しては，新しいデータでモデルを再トレーニングする. 結果は表 3 に示されている．

モデル	Lenet5	VGG16	Resnet20
BER	0.0	0.0	0.0

表 3　微調整攻撃の行った BER の値

剪定攻撃については，モデルの 0.1 未満の重みをすべてゼロにする. 結果は表 4 に示されている．

モデル	Lenet5	VGG16	Resnet20
BER	0.0	0.0	0.0

表 4　剪定攻撃の行った BER の値

実験結果に基づき，私たちの提案は主流の攻撃に効果的に対抗できることが確認された．

5 おわりに

本論文では，予備訓練なしの DNN 電子透かし方法を提案した．これにより，モデルの著作権を保護しつつ，モデルの性能に影響を与えることなくである．そして今後の電子透かし攻撃に対抗できるために，我々は研究を続ける．

参考文献

[1] J. Guo, and M. Potkonjak, “Evolutionary trigger set generation for dnn black-box watermarking”, arXiv:1906.04411, 2019.

[2] Y. Uchida, Y. Nagai, S. Sakazawa and S. Satoh, “Embedding watermarks into deep neural networks”, Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval, pp. 269-277, 2017.

[3] B. D. Rouhani, H. Chen, and F. Koushanfar. “Deepsigns: An end-to-end watermarking framework for ownership protection of deep neural networks”, Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, 2019.

[4] M. Barni, F. Pérez-González, and B. Tondi, “DNN watermarking: Four challenges and a funeral”, Proceedings of the 2021 ACM Workshop on Information Hiding and Multimedia Security, 2021.