MEGEX：勾配系の説明可能なAIに対するデータフリーモデル抽出攻撃

三浦 尭之[1,2,a)]　芝原 俊樹[1]　矢内 直人[2]

概要：データフリーモデル抽出攻撃は，訓練済み機械学習モデルをその入出力から復元するモデル抽出攻撃を，一切の訓練データを用意することなく実現する攻撃である．勾配系の説明可能な AI に対するデータフリーモデル抽出攻撃がより効率的になることが近年示されたが，実際に用いられる勾配は解釈性向上のために加工されていることが多い．本稿ではこうした加工が攻撃成功率に与える影響を調査するため，特に実用されている Integrated Gradient に対する攻撃手法を提案し，実験評価を行った．評価は複数のデータセットを用い，データセットの分類難易度が上がると攻撃にはより正確な勾配が必要になるということを明らかにした．また，攻撃の成功率と生成モデルの学習度合いの関係を考察するため，学習途中の生成モデルが出す画像を可視化し，傾向の違いを確認した．

キーワード：データフリーモデル抽出攻撃，説明可能な AI, SmoothGrad, Integrated Gradient

MEGEX: Data-Free Model Extraction against Gradient based Explainable AI

Takayuki Miura[1,2,a)]　Toshiki Shibahara[1]　Naoto Yanai[2]

Abstract: Data-free model extraction attacks are model extraction attacks, in which an attacker recovers trained machine learning models from their inferences, without any training data. These attacks against Explainable AI have recently been shown to be more efficient. However, explanations in the world are often modified to improve their interpretability. In this paper, we introduce an attack against Integrated Gradient, which is a gradient-based explainable AI and pratically used in the world, in order to investigate a relationship between their modificaitons and attack success rates. We also conduct experiments to verify the difference in attack success rates between several datasets. In order to examine the relationship between attack success rates and the generative model, we visualized the images produced by the generative model.

Keywords: data-free model extraction attack, explainable AI, SmoothGrad, integrated gradient

1. はじめに

近年，機械学習の実社会での利活用の増加を受けて，モデルの知財性やモデルの推論の解釈性に関心が集まっている．モデルの知財性や所有権に関する脅威として，訓練済みモデルをその入出力の情報から復元してしまうモデル抽出攻撃に注目が集まっている [1]．この攻撃では攻撃者がまず大量のラベルなしデータを収集し，攻撃対象となる被害モデルにラベル付けさせ，手元に独自の訓練データを作成する．そして，そのデータを用いて学習を行い自身の手元に同等精度のモデルを復元する（得られたモデルをクローンモデルと呼ぶ）．

しかし，盗みの対象になるようなモデルはラベルなしデータの収集すら困難なデータから学習されているという考察から，攻撃者がラベルなしデータの収集すらなしで行える**データフリー**モデル抽出攻撃が提案されている [2–4]．データフリーモデル抽出攻撃では，攻撃者は深層生成モデ

[1] NTT 社会情報学研究所, NTT Social Informatics Laboratories
[2] 大阪大学, Osaka University
[a)] takayuki.miura.br@hco.ntt.co.jp

ルを用いて，盗みに効果的な入力サンプルを生成し，そのラベルを被害モデルにつけさせる．この攻撃は，データの形式さえ知っていればアルゴリズムのみで行える攻撃であり，近年では商用の推薦サービス [4] などへの応用も確認されている．その一方，通常のモデル抽出攻撃よりも多くのクエリを必要としてしまうことから、非効率的な攻撃しか知られていなかった．

これに対して，被害モデルが説明可能な AI である場合，攻撃者がその説明を利用することで，従来よりも効率的なデータフリーなモデル抽出攻撃が可能となる [5,6]．具体的には，Vanilla Gradient [7] という説明手法で提示される勾配情報をうまく生成モデルの学習に組み込むことにより行われる．Vanilla Gradient がモデルの入力に対する勾配そのものであることから，理論的にも white box 知識蒸留と同等の学習を生成モデルに与えていることがわかる．

しかし，実際のサービスにおける勾配系の説明は，解釈性や安定性向上のために純粋な勾配をより詳細に加工した形式の出力が与えられることが一般的である [8–10]．そのため，Vanilla Gradient のようにきれいに生成モデルの学習の材料になるかどうかは理論的には不明である．著者らは，既存の攻撃を SmoothGrad に対しても行えるフレームワークに拡張し，実験評価を行うことで，SmoothGrad による勾配を用いることはむしろ学習の妨害になるということを確かめた [11]．しかし，用いたデータセットは cifar10 [12] の一通りのみであり，また，SmoothGrad の平滑化パラメータの影響も考慮されていなかった．

本稿では，説明がその有用性向上のために加工されると，説明可能な AI に対するデータフリーモデル抽出がどの程度軽減されるかを調査した．そのために，Google Cloud の説明可能 AI のサービスで勾配系の説明としても実装されている Integrated Gradient[*1] という説明可能な AI が出力する説明を用いたデータフリーモデル抽出攻撃を提案し，それら方式を実装し実験評価を行った．加えて，SmoothGrad の平滑化パラメータもいくつかのパターンを試して，説明に対する加工の度合いと攻撃の成功率の関係を調査した．また，従来の実験評価 [11] では Cifar10 だけだったデータセットを FahionMNIST [13] と SVHN [14] に拡張し，タスクの難しさと攻撃の成功率に関する知見も得た．さらに，攻撃の最中の生成モデルが出力する画像も可視化し，攻撃がうまくいっている場合とうまくいっていない場合で生成される画像の傾向に違いがあることを明らかにした．本稿の貢献は下記にまとめられる；

- Integrated Gradient に対する攻撃方法を提案した．
- 提案攻撃や SmoothGrad への既存攻撃のパラメータを変えたものを複数のデータセットで実験評価し，パラメータ間，データセット間の攻撃精度の違いを明らかにした．結果として，データセットが難しくなるにつれ，より正確な勾配が攻撃に必要になるということが明らかになった．
- 学習途中の生成モデルが出す画像を可視化することで，攻撃がうまくいっている場合とうまくいっていない場合に生成モデルが出す出力に違う傾向があることを明らかにした．

こうした結果は「説明の人間に対する解釈性を上げることでコンピュータに都合のいい情報が落ちて，逆に攻撃に耐性がつく」という可能性を示唆している．

[*1] https://cloud.google.com/explainable-ai

2. 関連研究

関連研究としてモデル抽出攻撃と説明可能な AI に対する攻撃の既存研究に対する本稿の位置づけを説明する．

2.1 モデル抽出攻撃

モデル抽出攻撃は入出力の情報からそのモデルと同等のモデルを復元してしまう攻撃である [1]．盗まれる訓練済みモデルを**被害モデル**，攻撃者が手元に復元したモデルを**クローンモデル**と呼ぶ．被害モデルがクラウドサービスを通して公開されており，利用者が正当なモデルの使用を続ける中，手元に同等のモデルを復元するという設定で議論される [15,16]．攻撃者の目的は「同等精度のクローンモデルを得ること」と「二段階攻撃のための足掛かりとしてクローンモデルを得る」という二つに分類されるが，それぞれの評価指標として，クエリ量当たりのクローンモデルの Accuracy と Fidelity という値がある [17]．Accuracy はテストデータに対する「クローンモデルの精度」で測り，Fidelity はテストデータに対する「被害モデルとクローンモデルの分類結果の一致度で測る」ことが一般的である．

特に近年，盗む価値のあるモデルはそもそも，ラベルなしデータすら集めることも困難な希少なデータから作られているという考察から，深層生成モデルを用いたデータフリーモデル抽出攻撃が提唱されている [2,3]．この方式では，攻撃者は深層生成モデルを用いてクエリ用のラベルなしデータを生成する．このとき，既存の攻撃では勾配を近似計算するだけだったが，本稿の設定での攻撃では説明を通じて攻撃者が勾配を正確に復元することが可能である．

また，被害モデルが推薦システムの場合の攻撃 [4]，あるいは表形式データの場合の攻撃 [18] など、各モデルに調整した攻撃も提案されている．本稿における説明可能 AI への攻撃は，上述した各モデルに調整した攻撃と組み合わせることも可能である．

2.2 説明可能な AI に対する攻撃

説明可能な AI に対するモデル抽出攻撃は，著者らの研究も含めると大きく 4 つある．Milli らは，Vanilla Gradient による説明がつき，活性化関数が ReLU（定義 3.1）である

ニューラルネットに対して，その中間層の次元 d に対して，$O(d \log \frac{d}{\delta})$ で正確な復元が可能であることを示した [19]．また，実験では通常のモデル抽出攻撃における損失関数に説明の二乗距離も組み込んだものを提案し実験を行った．Aivodji ら [20] は，反実仮想的な説明 [21] がついたモデルに対して，説明として出力されるサンプルを攻撃者がクローンモデルの学習に利用することで攻撃が効率的になることを示した．Wang ら [22] は，Aividji らの研究をより発展させた．反実仮想的な説明が敵対的サンプル相当のことを行っている事実に注目し，説明として得られたサンプルを再びクエリに投げることで被害モデルの決定境界の情報を効率的に集めた．

これら 3 つの手法は，通常のモデル抽出攻撃と同様で，攻撃者がラベルなしデータを収集済みであることが仮定されており，データフリー設定での脅威は検討していなかった．本稿の前身になる著者らによる説明可能な AI に対するデータフリーモデル抽出攻撃 [5, 6, 11] では，Vanilla Gradient や SmoothGrad に対する脅威のみ検討していたが，本稿では，Integrated Gradient にもスコープを当て，かつ，実験データセットに FashionMNIST や SVHN なども拡充し，データセット間の傾向の違いなどの考察も行った．

3. 準備

本節では，提案手法の記述に必要な情報として、深層学習、説明可能な AI の説明をする．

3.1 DNN とその学習

提案手法の説明に必要な深層学習の用語を紹介する．

定義 3.1 (DNN)**.** 関数 $f : \mathbb{R}^d \to \mathbb{R}^c$ で，有限個のアフィン変換 $g_i : \mathbb{R}^{m_i} \to \mathbb{R}^{n_i}$ と活性化関数 $\sigma_i : \mathbb{R}^{n_i} \to \mathbb{R}^{m_{i+1}}$ の合成の列 $f = \sigma_n \circ g_n \circ \cdots \circ \sigma_1 \circ g_1$ と表せるものを **DNN** という．ここで，アフィン変換 g_i とは，行列 $A \in M_{n_i m_i}(\mathbb{R})$ とベクトル $b \in \mathbb{R}^{n_i}$ で $g_i(x) = Ax + b$ と表せる関数をいう．活性化関数は本稿では基本的には $\mathrm{ReLU}(x) := \max(x, 0)$ を用いる．また，最終層の活性化関数には，各 i に対して，$\sigma_n(x)_i = e^{x_i} / \sum_{j=1}^{c} e^{x_j}$ となる softmax 関数を用いる．また，softmax 関数に入れる直前の数値を **logit** と呼ぶ．

定義 3.2 (損失関数，目的関数)**.** 教師データ (x, y) に対して，モデル f の推論と正解の距離を測る関数を**損失関数**と呼び，$l : \mathbb{R}^c \times \mathbb{R}^c \to \mathbb{R}_{\geq 0}$ という形の関数を用いて $l(f(x), y)$ と表現する．また，教師データセット $D = \{(x_i, y_i)\}$ に対して，損失関数の平均などを $L(D) = \frac{1}{|D|} \sum_{i=1}^{|D|} l(f(x_i), y_i)$ とおき**目的関数**と呼ぶ．この目的関数を小さくするような *DNN* のパラメーターを探索することが学習である．

深層学習では，DNN における各アフィン変換の行列とベクトルの係数がパラメーターとなるので，$\theta_f \in \mathbb{R}^W$ と表すことができる．学習は**確率的勾配降下法**（stochastic gradient descent, SGD) という手法を用いて行われる．この手法では，ランダムにサンプリングされた訓練データ $D' \subset D$ に対して，パラメーターによる勾配 $\nabla_{\theta_f} L(D')$ を用いて，$\theta_f \leftarrow \theta_f - \eta \nabla_{\theta_f} L(D')$ というふうにパラメーターの更新を行う．ここで $\eta > 0$ は学習率と呼ばれる．

定義 3.3 (生成モデル)**.** $\mathbf{X}$ を生成したいデータの取りうる形式全体の集合とする（本稿では $\mathbf{X} \cong \mathbb{R}^d$）．このとき，関数 $G : \mathbb{R}^r \to \mathbf{X}$ で多変数正規分布に従う $X \sim \mathcal{N}(0, I_r)$ に対して，$G(X)$ が $\mathbf{X}$ 上の所望の分布によく似ているもの[*2]を**生成モデル**という．

3.2 勾配系の説明可能な AI

深層学習は画像認識や自然言語処理で高い精度を発揮するが，その一方でその推論結果に至った根拠が不透明という課題がある．それに対する解決案として，人間にとって解釈性を高める**説明可能な AI** が提案されている．本稿では特に画像分類系のモデルに対して，直感的な意味を説明できる勾配系の説明にスコープを当てる．これから紹介する各説明手法による説明画像は図 5 である．

説明されるモデルをほとんど至る点で微分可能な関数 $f : \mathbb{R}^d \to \mathbb{R}^c$ とする．**Vanilla Gradient** [7] は，画像分類などで分類結果に大きな影響を与えたピクセルをハイライトする勾配ベースの説明手法の一種である．入力 $x \in \mathbb{R}^d$ に対して，勾配

$$\mathrm{VG}(x) := \nabla_x f(x) \in \mathbb{R}^{d \times c}$$

を考える．偏微分の値の絶対値が大きいピクセル x_i は出力結果の値に大きな影響を与えていると解釈できるため，このベクトルは，そういった意味で分類結果に対する説明になっている．しかし，この $\mathrm{VG}(x)$ は入力の変化に敏感すぎるという問題があるため，**SmoothGrad** [8] では次の平滑化を行い，説明として出力している．

$$\mathrm{SG}(x) := \frac{1}{m} \sum_{i=1}^{m} \mathrm{VG}(x + z_i), \qquad (1)$$

ここで $z_i \sim \mathcal{N}(0, \sigma^2 I_d)$ である．原著論文 [8] では，$(\max x - \min x)/10 \leq \sigma \leq (\max x - \min x)/5$，$m \leq 50$ 程度を目安としている．

こうした微分値ベースの Vanilla Gradient や SmoothGrad は分類結果に重要な影響を与えるピクセル x_i の貢献度がそこの近傍で “重要なまま変化しない” 場合は $\frac{\partial f}{\partial x_i} = 0$（貢献度が低いことを意味する）となってしまう問題があった．これを解決したのが **Integrated Gradient** [10] である．ベースラインと呼ばれる点 $x' \in \mathbb{R}^d$ を一つ固定し，$x \in \mathbb{R}^d$ に対して，

$$\mathrm{IG}_f(x, x') := (x - x') \odot \int_0^1 \nabla f(x' + t(x - x'))dt$$

[*2] 例えば，32×32 で RGB の画像全体の集合の中における漁船と思える画像の分布など．

Algorithm 1 : MEGEX

Input: $f, Q, N_G, N_C, \eta_G, \eta_C$
Output: $\hat{f}$

Initialize $G, \hat{f}, q \leftarrow 0$
while $q < Q$ **do**
  for $i \leftarrow 1, \ldots, N_G$ **do** ▷ G の学習
    $z \leftarrow \mathcal{N}(0, I_r)$ ▷ 多変数正規分布に従うノイズ
    $x \leftarrow G(z)$
    $y \leftarrow f(x)$, $g \leftarrow \mathrm{Exp}(x)$ ▷ クエリ結果
    if exp = int **then** ▷ 説明が IG の場合
      $y_2 \leftarrow f((1+\alpha)x)$, $g_2 \leftarrow \mathrm{Exp}((1+\alpha)x)$
      $g \leftarrow (g_2 - g)/\alpha$
    $\hat{y} \leftarrow \hat{f}(x)$
    Compute $\nabla_x L(x)$ with $y, g, \hat{y}$ ▷ 式 (3) で計算
    $\nabla_{\theta_G} \mathcal{L}_f(z) \leftarrow \nabla_x L(x) \cdot \nabla_{\theta_G} G(z)$
    $\theta_G \leftarrow \theta_G + \eta_G \nabla_{\theta_G} \mathcal{L}_f(z)$
  for $i \leftarrow 1, \ldots, N_C$ **do** ▷ $\hat{f}$ の学習
    $z \leftarrow \mathcal{N}(0, I_r)$ ▷ 多変数正規分布に従うノイズ
    $x \leftarrow G(z)$
    $y \leftarrow f(x)$ ▷ クエリ結果
    $\hat{y} \leftarrow \hat{f}(x)$
    Compute $\nabla_{\theta_{\hat{f}}} \mathcal{L}_f(z)$ ▷ 通常の誤差逆伝搬法
    $\theta_{\hat{f}} \leftarrow \theta_{\hat{f}} - \eta_C \nabla_{\theta_{\hat{f}}} \mathcal{L}_f(z)$
  $q \leftarrow q + N_G + N_C$

の値を説明として出力する．ただし，ここで $\odot$ は各成分の要素ごとの積を表す記号とする．

特に，実装上はステップ数 m を定めて，

$$\mathrm{IG}^{approx}_{f,m}(x, x') = (x - x') \odot \frac{1}{m} \sum_{i=1}^{m} \nabla f(x' + \frac{i}{m}(x - x')) \tag{2}$$

という数値で代替している．ただし，画像分類ではベースラインを $x' = 0 \in \mathbb{R}^d$ と選ぶことが多いため，本稿でも $x' = 0$ とすることとし，$\mathrm{IG}_f(x, x') = \mathrm{IG}(x)$ と略記する．

4. MEGEX

本節では，提案攻撃である MEGEX（Model Extraction against Gradient-based EXplainable AI）を紹介する．本稿では Integrated Gradient に適用する方法を新たに紹介するが，攻撃のフレームワーク自体は [5,6,11] と同様であるため，詳細はそれらを参照されたい．

4.1 概要

MEGEX は深層生成モデルを用いた手法 [2,3] をベースに，被害モデルが勾配系の説明可能な AI である場合を想定している．手法の詳細な流れは，Algorithm 1 のとおりである．Q はあらかじめ決めたクエリ上限，N_G, N_C は学習の割合，η_G, η_C は学習率を表す．

この手法では，深層生成モデル $G : \mathbb{R}^r \to \mathbb{R}^d$ とクローンモデル $\hat{f} : \mathbb{R}^d \to \mathbb{R}^c$ を競合的に学習させている．損失関数を，被害モデル $f : \mathbb{R}^d \to \mathbb{R}^c$ に対して

$$\mathcal{L}_f(z, \theta_{\hat{f}}, \theta_G) := \ell(\hat{f}(G(z)), f(G(z)))$$

と置く．このとき，次の最適化問題を解くことが本手法のゴールである．

$$\min_{\theta_{\hat{f}}} \max_{\theta_G} \mathbb{E}[\mathcal{L}_f(Z, \theta_{\hat{f}}, \theta_G)],$$

ここで，$Z \sim \mathcal{N}(0, I_r)$ である．この最適化を近似的に解くために，$\nabla_{\theta_{\hat{f}}} \mathcal{L}_f$ と $\nabla_{\theta_G} \mathcal{L}_f$ を交互に用いて確率的勾配降下法で学習を行う．通常のモデル抽出攻撃においても，攻撃者はクエリ結果 $f(G(z))$ があれば $\nabla_{\theta_{\hat{f}}} \mathcal{L}_f(z, \theta_{\hat{f}}, \theta_G)$ は計算可能なのでクローンモデルの学習は行うことができる．一方で，生成モデルの学習で必要な $\nabla_{\theta_G} \mathcal{L}_f(z, \theta_{\hat{f}}, \theta_G)$ の計算については，説明を用いた工夫が必要である．まず，

$$L(x) := \ell(\hat{f}(x), f(x))$$

と置くと，$\mathcal{L}_f = L \circ G$ となる．このとき，$x = G(z)$ とおくと合成関数のヤコビ行列は，ヤコビ行列の行列積であるので，

$$\nabla_{\theta_G} \mathcal{L}_f = \nabla_x L(x) \cdot \nabla_{\theta_G} G(z)$$

と表現できる．いま，生成モデル G のパラメータは攻撃者が手元に持っているので勾配 $\nabla_{\theta_G} G(z)$ は誤差逆伝搬法を用いて計算可能である．なので，残りの問題は $\nabla_x L(x)$ が得られるかどうかに帰着される．

ここで，$L = \ell \circ (\hat{f}, f)$ であることから，ℓ の第一変数を y_1，第二変数を y_2 とおくと

$$\begin{aligned} \nabla_x L(x) =& \nabla_{y_1} \ell(\hat{f}(x), f(x)) \cdot \nabla_x \hat{f}(x) \\ &+ \nabla_{y_2} \ell(\hat{f}(x), f(x)) \cdot \nabla_x f(x) \end{aligned} \tag{3}$$

となるがここで通常の攻撃者が持っていない情報は，$\mathrm{VG}(x) = \nabla_x f(x)$ のみとなるため，モデルが $\mathrm{VG}(x)$ の説明を出力すれば攻撃者は完全な $\nabla_x L(x)$ を得られる．ここで説明が SmoothGrad の場合は $\mathrm{SG}(x)$ が $\mathrm{VG}(x)$ を平滑化しただけという関係に注目し，$\nabla_x f(x) \approx \mathrm{SG}(x)$ という想定での攻撃を行う [11].

4.2 Integrated Gradient への攻撃の適用

活性化関数が ReLU のニューラルネットワークは局所的に線形写像とみなすことができる [23]．線形写像の微分値は任意の点で同じ定数となっているのでため，十分小さい摂動 ε に対して，ニューラルネットワークは

$$\nabla_x f(x) = \nabla_x f(x + \varepsilon)$$

が成り立つ．この観察に基づくと，十分小さい実数 $\alpha > 0$ に対して次の式が成り立つ；

$$\frac{\mathrm{IG}((1+\alpha)x) - \mathrm{IG}(x)}{\alpha} = \nabla_x f(x).$$

これより，Integrated Gradient から Vanlla Gradient が

表 1　実験設定

	cifar10	FashionMNIST	SVHN
被害モデル	ResNet34	LeNet5	ResNet34
クローンモデル	ResNet18	LeNet5	ResNet18
被害モデル精度	95.54%	90.7%	96.17%
損失関数	ℓ_1	KL	ℓ_1
クローン opt	SGD	SGD	SGD
クローン lr	0.1	0.01	0.1
生成モデル opt	Adam	Adam	Adam
生成モデル lr	5e-4	5e-4	5e-5
クエリ回数	20×10^6	5×10^6	1×10^6

復元できることがわかる．実装上は IG は式 2 のように $\mathrm{IG}_m^{approx}(x)$ と近似されているので，

$$\frac{\mathrm{IG}_m^{approx}((1+\alpha)x) - \mathrm{IG}_m^{approx}(x)}{\alpha} \approx \nabla_x f(x) \qquad (4)$$

として攻撃を行うこととする．このとき，二回の説明から 1 点の勾配情報を復元していることに注意する（クエリ量がこのときだけ 2 倍になる）．

5. 実験

本節では実験設定と評価結果について説明する．

5.1 実装

本実験の実装は Truong らによる先行研究 [3] において github で公開されている実装をベースとして行った[*3]．ゼロ次勾配推定において近似されていた $\nabla_x L(x)$ を説明を用いて復元した情報に置き換えて行った．微分値の計算は PyTorch の autograd 機能に基づく．

5.2 実験設定

実験設定は基本的に表 1 にあるとおりである．データセットは Cifar10，FashionMNIST，SVHN の 3 種類を使用した．ニューラルネットのモデルについては基本的に Troung らが公開した Github に入っていたものを用いた．Cifer10, SVHN については被害モデルは ResNet34，クローンモデルは ResNet18 とした．FashionMNIST については，被害モデル，クローンモデルともに LeNet5 とした．被害モデルのテスト精度は cifar10, FashionMNIST, SVHN の順に 95.54%，90.7%，96.17%のものを使用した．生成モデルは 5 層のニューラルネットワークを用い，損失関数は「KL-divergence（以降 KL）」，「logit 復元型の ℓ_1 関数（以降 ℓ_1）」を用いた．学習のオプティマイザーは，クローンモデルに対しては SGD，生成モデルに対しては Adam を用いた．それぞれの学習率の組は (0.1, 5e-4), (0.01, 5e-4), (0.1, 5e-5) とした．「盗みの種類」については次の 8 通りを採用した．

- **megex**：Vanilla Gradient に対するデータフリーモデル抽出攻撃 [5,6]．得られた微分 $\nabla_x f(x)$ から手計算で復元した勾配 $\nabla_x L(x)$ を用いた．
- **smooth_m**（$m = 1, 10, 25$）：SmoothGrad に対するデータフリーモデル抽出攻撃．SmoothGrad は $\sigma = 0.1 \times (\max x - \min x)$ で固定し，$m = 1, 10, 25$ のものをそれぞれ検証した．
- **int**：式 4 を用いて，Integrated Gradient から得た勾配を利用（$\alpha = 1e-8$）．Integrated Gradient は式 2 の通りに実装し $m = 50$ とした．
- **dfme**：Truong らの実装そのもの．ハイパーパラメーターなどは github のものをそのまま用いた．
- **random**：生成モデル G の学習を止めた場合．クローンモデルはただラベル付けされたランダムな画像で学習を行うことになる．
- **default_grad**：被害モデルの勾配も自由に使える場合．ホワイトボックス設定の知識蒸留に相当する．Truong らの github にある「compute_gradient」という関数を用いると実現できる．Pytorch の backward 関数で自動的に $\nabla_x L(x)$ を計算している．

[*3] https://github.com/cake-lab/datafree-model-extraction

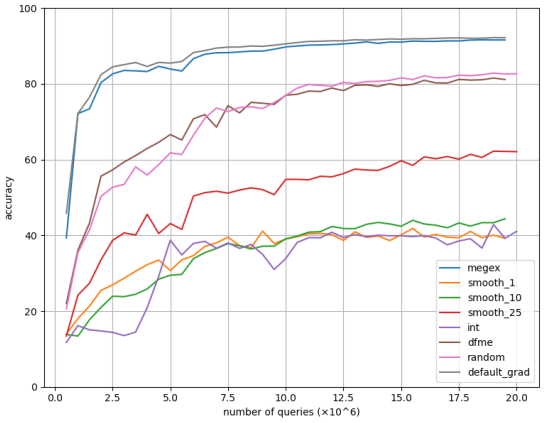

図 1　Cifar10 に対する結果

5.3 結果

各データセットについて表 1 の通りに実験した結果が，図 1, 2, 3 である．横軸はクエリ数で，縦軸はその時点でのクローンモデルのテスト精度になる．

Cifar10 に対しては，既存結果 [6] 同様，megex はベースラインの default_grad と同様に早い段階で高い精度のクローンモデルが手に入っていた．また，smooth については平滑化が多い $m = 25$ の時は最終テスト精度が 60%程度まで到達したのに対して，$m = 1, 10$ の時は 40%程度にとどまった．int についても smooth_1, smooth_10 と同様の結果になった．

FashionMNIST については int と random を除くすべての手法が同等程度の精度の結果となった．また，わずかな

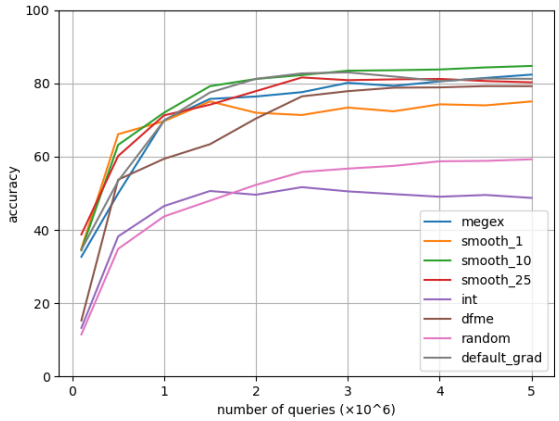

図 2　FashionMNIST に対する結果

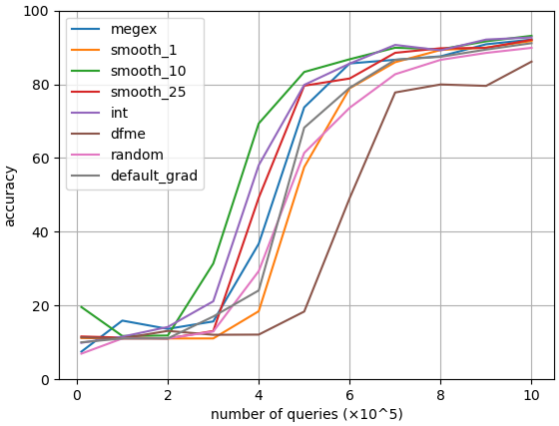

図 3　SVHN に対する結果

がら smooth_1 が他手法と比べると最終的なクローンモデルの精度が悪かった．int については 1.5×10^6 クエリあたりからクローンモデルのテスト精度が若干の減少傾向にあった．

SVHN に関しては，すべての場合で同じような結果となった．dfme による手法に若干の遅れがあった．random も同様によく学習しているので，説明などの情報がいかせているのか否かが不明な結果となった．

5.4　考察

前項の結果を踏まえて，下記 5 つの観点から考察を行う．

5.4.1　int の攻撃精度が低かったこと

Cifar10 と FashionMNIST のとき，int の攻撃がうまくいかなかった原因として，復元の精度があげられる．著者らの先行研究 [24] では，提案の式 4 で 4 層の NN までは復元が可能であるが，ResNet18 などの深い層のモデルでは難しいことが報告されていた．これは局所線形になる領域が細かくなってしまい，現実的な摂動では収めることができなくなっていると考えることができる．本設定でもこのようなことが攻撃成功率低下の原因になっていると考えることができる．

5.4.2　データセット間の比較

勾配系の説明を用いた提案攻撃が成功しているかという観点で見る．まず，Cifar10 では純粋な勾配を用いた megex のみがうまくいくという結果になっていた．FashionMNIST は megex に加えて，smooth_1, 10, 25 も同様にうまくいき，int のみうまくいかないという結果になっていた．そして，SVHN では int も含むすべての提案手法がうまくいっていた．

このような傾向から，データセットの学習が難しくなればなるほど，より正確な勾配が必要になるということ言える．これはより精密な勾配により生成モデルが正確に学習されていることが，今回の難しいデータセットへのモデル抽出では大事であるということを意味している．

5.4.3　SmoothGrad の平滑化回数と攻撃成功率

Cifar10 では，平滑化が $m = 25$ である SmoothGrad に対する攻撃の方が $m = 1, 10$ の場合より良い結果となっていた．式 1 より，SmoothGrad は入力を摂動させ，微分値の平均値をとっている．平滑化の回数が増えることで入力を少しずらしてしまうことの影響が薄れていると考えることができる．FashionMNIST でも smooth_1 が他手法より若干悪いという傾向が見て取れたのは，同様の理由によると考えられる．

データセット間の比較結果からも考えられることであるが，より精度の高い攻撃には生成モデルがしっかり学習していることが重要であると考えられる．勾配の誤差と生成モデルの学習具合の定量的な関係性については今後の課題としたい．

5.4.4　生成画像の観察

本稿では，学習の各ステップで生成モデルが生成した画像をランダムに 25 枚ずつ保存し目視により観察を行った．

学習がうまくいった FahionMNIST の megex の学習初期が図 6，学習終盤が図 7 である．一方で学習がうまくいかなかった FashionMNIST の int の学習初期が図 8 で，学習終盤が 9 である．

学習初期の図 6 と図 8 の 2 つを見比べるとほとんど差がない画像になっているが，終盤の画像である図 7 と図 9 は明らかに違う模様を出力している．

データフリーモデル抽出では生成モデルは「何か一つある分布を学習しようとしている」わけではなく，学習のその段階で，「被害モデルとクローンモデルがより異なる推論をする入力を探す」ということを行っているので，終盤でも図 7 や図 9 のようになることはおかしいことではない．先行研究でも似たような画像が生成されている [2,3]

生成モデルの学習度合いと出力される模様の傾向に関しても今後の課題としたい．

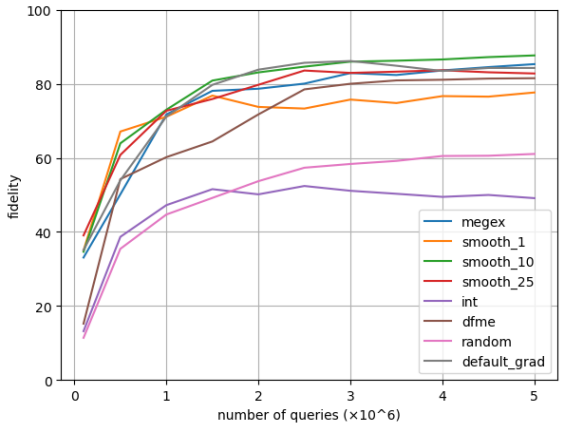

図 4 FashionMNIST に対する Fedility の結果

5.4.5 忠実度（Fidelity）の観察

関連研究の 2.1 項で紹介した忠実度（Fidelity）についても測定した（図 4）．被害モデルの精度が高すぎると Fidelity の値にほとんど差がなくなってしまうので，被害モデルの精度が最も小さかった FashionMNIST のときを図示している．結果としてはほとんど差が出ず，全体として Accuracy < Fidelity という傾向がわずかにみられる結果となった．

6. 結論

本稿では，従来の勾配系の説明可能な AI に対するデータフリーモデル抽出攻撃に対して，Integrated Gradient に対しても適用できる手法を提案し，説明が有用性向上のために加工されるとどの程度攻撃が難しくなるかを評価した．

参考文献

[1] Tramèr, F., Zhang, F., Juels, A., Reiter, M. K. and Ristenpart, T.: Stealing machine learning models via prediction apis, *25th USENIX Security Symposium (USENIX Security 16)*, pp. 601–618 (2016).

[2] Kariyappa, S., Prakash, A. and Qureshi, M. K.: MAZE: Data-Free Model Stealing Attack Using Zeroth-Order Gradient Estimation, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13814–13823 (2021).

[3] Truong, J.-B., Maini, P., Walls, R. J. and Papernot, N.: Data-Free Model Extraction, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021).

[4] Yue, Z., He, Z., Zeng, H. and McAuley, J. J.: Black-Box Attacks on Sequential Recommenders via Data-Free Model Extraction, *Proceedings of Conference on Recommender Systems (RecSys)*, ACM, pp. 44–54 (2021).

[5] 三浦尭之，長谷川聡：説明可能な AI に対するデータフリー Model Stealing 攻撃，暗号と情報セキュリティシンポジウム SCIS2021 予稿集, Jan. 2021 (2021).

[6] Miura, T., Hasegawa, S. and Shibahara, T.: MEGEX: Data-Free Model Extraction Attack against Gradient-Based Explainable AI, *arXiv preprint arXiv:2107.08909* (2021).

[7] Simonyan, K., Vedaldi, A. and Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps, *arXiv preprint arXiv:1312.6034* (2013).

[8] Smilkov, D., Thorat, N., Kim, B., Viégas, F. and Wattenberg, M.: Smoothgrad: removing noise by adding noise, *arXiv preprint arXiv:1706.03825* (2017).

[9] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. and Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization, *Proceedings of the IEEE international conference on computer vision*, pp. 618–626 (2017).

[10] Sundararajan, M., Taly, A. and Yan, Q.: Axiomatic attribution for deep networks, *arXiv preprint arXiv:1703.01365* (2017).

[11] 三浦尭之，芝原俊樹，矢内直人：勾配系の説明付きモデルに対するデータフリーモデル抽出攻撃，暗号と情報セキュリティシンポジウム SCIS2022 予稿集, Jan. 2022 (2022).

[12] Krizhevsky, A., Hinton, G. et al.: Learning multiple layers of features from tiny images (2009).

[13] Xiao, H., Rasul, K. and Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, *arXiv preprint arXiv:1708.07747* (2017).

[14] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B. and Ng, A. Y.: Reading digits in natural images with unsupervised feature learning (2011).

[15] Chandrasekaran, V., Chaudhuri, K., Giacomelli, I., Jha, S. and Yan, S.: Exploring connections between active learning and model extraction, *29th USENIX Security Symposium (USENIX Security 20)*, pp. 1309–1326 (2020).

[16] Juuti, M., Szyller, S., Marchal, S. and Asokan, N.: PRADA: protecting against DNN model stealing attacks, *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, IEEE, pp. 512–527 (2019).

[17] Jagielski, M., Carlini, N., Berthelot, D., Kurakin, A. and Papernot, N.: High Accuracy and High Fidelity Extraction of Neural Networks, *29th USENIX Security Symposium (USENIX Security 20)* (2020).

[18] Tasumi, M., Iwahana, K., Yanai, N., Shishido, K., Shimizu, T., Higuchi, Y., Morikawa, I. and Yajima, J.: First to Possess His Statistics: Data-Free Model Extraction Attack on Tabular Data, *arXiv preprint arXiv:2109.14857* (2021).

[19] Milli, S., Schmidt, L., Dragan, A. D. and Hardt, M.: Model reconstruction from model explanations, *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 1–9 (2019).

[20] Aïvodji, U., Bolot, A. and Gambs, S.: Model extraction from counterfactual explanations, *arXiv preprint arXiv:2009.01884* (2020).

[21] Wachter, S., Mittelstadt, B. and Russell, C.: Counterfactual explanations without opening the black box: Automated decisions and the GDPR, *Harv. JL & Tech.*, Vol. 31, p. 841 (2017).

[22] Wang, Y., Qian, H. and Miao, C.: DualCF: Efficient Model Extraction Attack from Counterfactual Explanations, *arXiv preprint arXiv:2205.06504* (2022).

[23] Rolnick, D. and Kording, K.: Reverse-engineering deep ReLU networks, *International Conference on Machine Learning*, PMLR, pp. 8178–8187 (2020).

[24] 三浦尭之，　権英哲，長谷川聡：ReLU ニューラルネットワークにおける Integrated Gradient の Vanilla Gradient への帰着，技術報告 26 (2021).

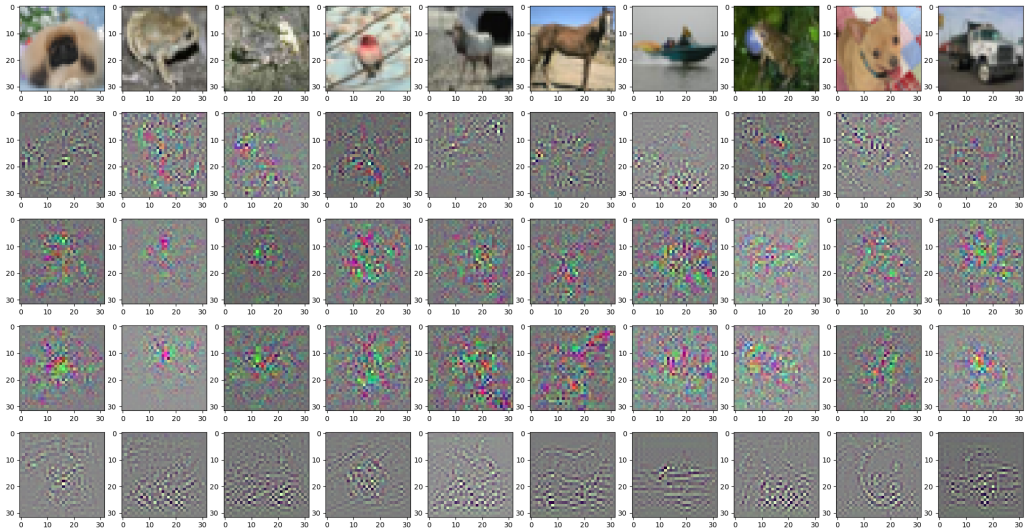

図 5　各種説明画像：上から「元画像」「Vanilla Gradient の説明」「$m = 10$ の SmoothGrad の説明」「$m = 25$ の SmoothGrad の説明」「Integrated Gradient の説明」

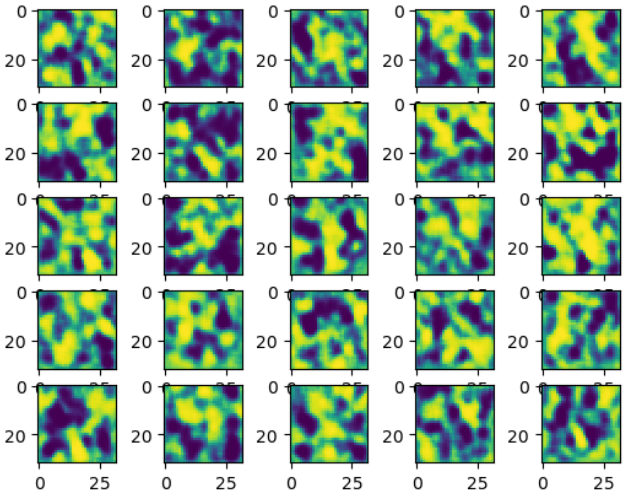

図 6　FashionMNIST 学習初期の生成データ（megex）

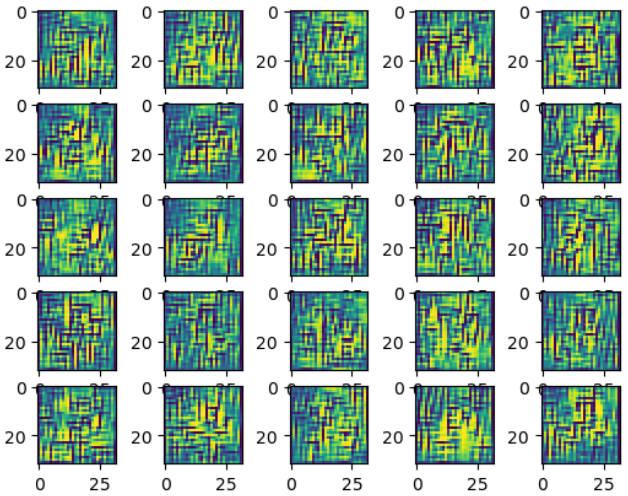

図 7　FashionMNIST 学習終盤の生成データ（megex）

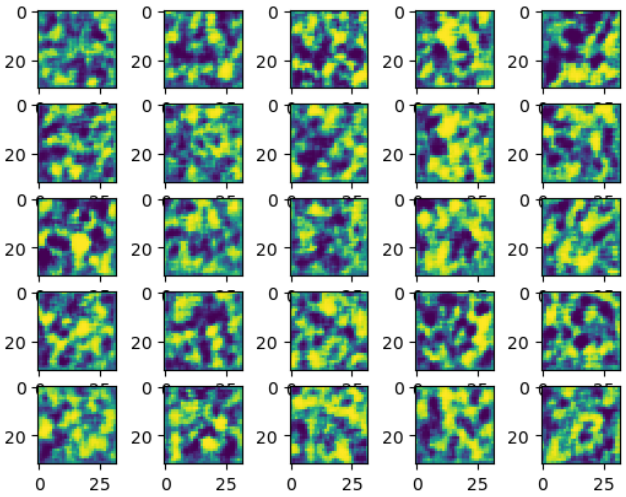

図 8　FashionMNIST 学習初期の生成データ（int）

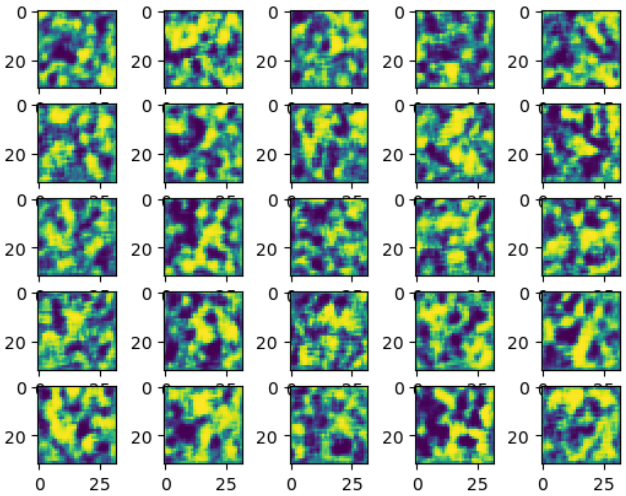

図 9　FashionMNIST 学習終盤の生成データ（int）