再攻撃を用いた敵対的サンプルの矯正の試み

森本 文哉[1,a)]　玉城 大生[1,b)]　小野 智司[1,c)]

概要：深層ニューラルネットワークには，入力に対して人間に知覚できない特殊な摂動が加えられた敵対的サンプル（Adversarial Examples: AE）を誤認識してしまう脆弱性が存在する．この脆弱性は認識結果の信頼性が重要なタスクにおいて深刻な問題であり，AE に対する防御手法が研究されている．防御手法の一つのアプローチとして，入力データから AE を検出する手法が提案されているが，これらの手法は AE の検知に留まっており，正しいクラスの識別は行われない．しかし，自動運転における標識認識など，一部のタスクでは AE を検出するだけでは不十分であり，AE の原画像における正しいラベルを認識することが求められている．このため本研究では，検出された AE に対して再度攻撃を加えることで，原画像の正しいラベルを推定する手法を提案する．

An Attempt of Counter-attack-based Rectification of Adversarial Examples

Abstract: Deep neural networks are vulnerable to misrecognition of Adversarial Examples (AEs), in which special perturbations are applied to inputs that are hardly perceptible to humans. This vulnerability is a serious problem in tasks where reliability of recognition results is important. As one of defense approaches against AEs, methods that detect AEs from input data are proposed; however, these methods are limited to detection of AEs only and do not attempt to recognize true classes of AEs. Some tasks such as sign recognition in autonomous driving require recognition of the correct label in the original image of AEs. For this reason, this study proposes the method of estimating the correct label in the original image by counter-attacking the detected AEs.

1. はじめに

深層ニューラルネットワーク（Deep Neural Network: DNN）は，画像分類や音声認識など様々な分野で高い性能を示しており，実応用が進んでいる．一方，近年の研究により，DNN に基づく学習器は入力データに対して，人間の知覚が困難な程度に微小でかつ特殊な摂動が加えられた敵対的サンプル（Adversarial Examples: AE）を誤認識してしまう脆弱性を有することが明らかにされている．この脆弱性は，自動運転における道路標識や画像に基づく個人認証など，セキュリティが重要なシステムにおいて深刻な問題であり，実世界で AE が悪用される可能性がある．

上記のような DNN をシステムに採用することの危険性を考慮して，AE に対する防御手法を有する DNN の研究も広く行われている．例えば，AE に対する防御手法として，モデルの訓練時に AE を学習させる敵対的訓練 [1] や画像変換によって AE の影響を弱める入力変換 [2,3] などが提案されている．一方，入力サンプルの特徴から AE を判別する検出手法 [4] も提案されている．検出手法は通常サンプルの認識精度を保証できるものの，AE を検知することに留まっており，攻撃前の画像における正しいカテゴリの認識までを考慮しない．AE として検出された入力を単純に棄却することが可能なタスクが多い一方で，上記の点が問題となるタスクも存在する．例えば自動運転における標識認識において，一時停止の標識に対して攻撃が加えられた際にそれを AE として検出はできるものの，防御手法のみでは一時停止の標識であることを認識することができず，何らかの後処理が必要となる．

本研究では，防御手法により検出された AE に対するラベルの矯正手法，すなわち，攻撃前の原画像における正しいラベルを推定する手法を提案する．本手法は，AE に対して再度攻撃を行うことで，誤分類されていた分類結果を

[1] 鹿児島大学
Korimoto, Kagoshima, Kagoshima 8900065, Japan
[a)] k6838914@kadai.jp
[b)] k8026496@kadai.jp
[c)] ono@ibe.kagoshima-u.ac.jp

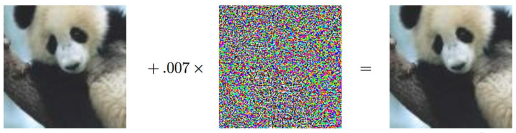

図 1: AE の一例 [6]

原画像の分類結果に矯正する．本手法は，DNN の入力信号の種別やタスクに依存せずに適用できる汎用性の高い手法であることに特徴がある．実験により，比較手法より高い矯正性能を有し，広範な攻撃手法に対して適用可能であることを示す．

2. 関連研究

2.1 敵対的攻撃

敵対的攻撃は，攻撃者が意図的に AE を生成する手法である．AE の一例として Goodfellow らによって生成された AE を図 1 に示す [6]．AE $\boldsymbol{x}'$ は入力画像 $\boldsymbol{x}$ に対して微小な摂動 $\boldsymbol{\delta}$ を加えることで生成され，以下の式で表される．

$$\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{\delta}, \quad s.t.\ C(\boldsymbol{x}') \neq C(\boldsymbol{x})$$

ここで $C(\cdot)$ は分類器の分類結果である．また，摂動 $\boldsymbol{\delta}$ は L_p ノルムが ε を下回ることとする（$\| \boldsymbol{\delta} \|_p < \varepsilon$）．$\| \boldsymbol{\delta} \|_p$ は以下の式で表される．

$$\| \boldsymbol{\delta} \|_p = (|\delta_1|^p + |\delta_2|^p + \cdots + |\delta_n|^p)^{\frac{1}{p}}$$

ここで $\delta_1, \delta_2, \cdots, \delta_n$ は摂動 $\boldsymbol{\delta}$ の各要素である．

敵対者の知識は，ホワイトボックス（White-Box: WB）とブラックボックス（Black-Box: BB）に大別される．WB ではモデルの勾配やパラメータに関する完全な知識を有する．一方，BB ではモデルに関する知識が無く，得られる情報には様々な程度がある．WB ではモデルの勾配情報を用いた，勾配ベースの攻撃 [1, 6–10] が強力である．BB においては，予測結果とその予測確率を用いるスコアベース攻撃 [11] や，予測結果のみから攻撃を行う決定ベース攻撃 [12, 13] がある．

敵対者の目標は，単に元の分類結果とは異なるラベルに誤分類させる非標的型攻撃と，サンプルを選んだ標的ラベルに意図的に誤分類させる標的型攻撃の 2 種類がある．一般に，特定のラベルに誤認識させる必要がある標的型攻撃より非標的型攻撃の方が容易である．多くの非標的型攻撃は正しい予測の信頼度を下げることで機能し，標的型攻撃は標的とする予測の信頼度を上げることで機能する．

2.1.1 勾配ベース攻撃

Goodfellow らは，Fast Gradient Sign Method（FGSM）を提案した [6]．FGSM はモデルの勾配に基づいて敵対的サンプルを生成する手法であり，反復を行わないワンステップな攻撃手法である．FGSM の目的は，勾配に沿ってワンステップずつ進むことで，勾配損失 $L(\theta, \boldsymbol{x}, y)$ を増加させることである．

Kurakin らは，FGSM の攻撃性能を改善し，Basic Iterative Method（BIM）を提案した [7]．BIM は，FGSM をより小さなステップサイズで数回実行するため，I-FGSM とも呼ばれる．

Madry らは，BIM をさらに改良して Projected Gradient Descent Method（PGD）を提案した [1]．BIM と PGD の違いは，BIM は元の入力で初期化する一方，PGD はランダムな点で初期化し，ランダムな攻撃再開を行う．ただし，その他の点では同じであるため，しばしば同じ手法として扱われる [14]．

Moosavi-Dezfooli らは，DeepFool を提案した [8]．この攻撃手法は，FGSM などの摂動パラメータを手動で設定する必要がある手法と比較して，攻撃に成功する最小の摂動量を求めることに特化した手法である．DeepFool の原理は，移動距離が非常に小さいとき，決定境界は線形と見なすことができ，テイラー展開を用いて線形化することで直交ベクトルを求め，直交ベクトルに沿って敵対的サンプルを探索する．

Carlini らは，敵対的サンプルの生成過程を，通常サンプルと敵対的サンプルの差を最小化するという最適化問題に変換した CW という攻撃手法を提案した [9]．CW は摂動を非常に小さくすることが可能であり，攻撃成功率が高い．

Papernot らは，Maximal Jacobian-based Saliency Map Attack（JSMA）を提案した [10]．この手法は，ヤコビアン行列を計算し，ヤコビアン行列に従って敵対的な顕著性マップを得るものである．敵対的顕著性マップにおいて最大の値を持つ画素を，貪欲アルゴリズムを用いて選択し，その画素に摂動を与える．摂動されたピクセル数が上限に達するまで，上記のステップを繰り返し，敵対的サンプルを生成する．

2.1.2 スコアベース攻撃

Narodytska らは，スコアベースの攻撃手法である LocalSearch を提案した [11]．この攻撃手法は，元のラベルの予測確率を最小化することで敵対的サンプルを生成する．貪欲的な局所探索を用いて，元画像から数ピクセル摂動された局所近傍画像を生成し，それらの画像から元のラベルの予測確率が最も小さくなる画像を選択する．上記のステップを，予測ラベルが任意の順位より低くなるまで繰り返す反復探索である．

2.1.3 決定ベース攻撃

Brendel らは，決定ベースの攻撃として Boundary Attack を提案した [12]．決定ベースの攻撃は予測ラベルのみが与えられる場合を想定しており，Boundary Attack では，画像の誤分類を維持した状態で決定境界に沿って原画像に近づけることで摂動量を最小化している．

Chen らは，モンテカルロ法を用いて局所的な決定境

界の近似を行うことで，決定境界の勾配を推定する HopSkipJumpAttack を提案した [13]．AE 近傍の領域から決定境界面に垂直な方向を推定し，二分探索と組み合わせることによって摂動量を最小化する．

2.2 敵対的防御

DNN モデルを実世界に応用する場合，敵対的攻撃により損害を与えることでインセンティブが得られる限り，敵対者からの攻撃を受ける危険性がある．このような攻撃からシステムを保護するため，敵対的攻撃からの防御手法である敵対的防御の研究がなされている．また実環境の多くは予測が困難なランダム性があり，敵対的サンプルを実環境における最悪のケースとして防御することで，システムの頑健性を検証できる．

敵対的防御手法は，主に敵対的訓練 [1,5,6,15]，入力変換 [2,3,16–18]，検出手法 [4,19–22] の 3 種類に大別される．

敵対的訓練は，敵対的攻撃を防ぐ最も一般的なアプローチであり，AE を学習データに含めることで AE に対する頑健性を向上させる．しかし，通常サンプルの精度が低下することや計算コストが大きくなってしまうことが問題に上げられる．

入力変換は，前処理によって入力データに変換を加えることで，AE の影響を弱める手法である．画像分類タスクにおいては，R&P 変換 [2] や JPEG 変換 [3] などによって，入力画像を変換する手法がある．R&P 変換は，入力画像をランダムな画像サイズに変更し，画像の周囲にゼロパディングを行う手法である．また，JPEG 変換では，JPEG 圧縮を通して画像変換を行うことで，AE の影響を弱めることを期待する．しかし，これらの入力変換手法は，すべてのサンプルに同様の変換を適用するため，通常サンプルが変換によって歪み，分類精度が低下する可能性がある．また，画像や音声といった DNN の入力データの種別に応じた処理が必要となる．

検出手法は，入力サンプルの特徴から AE であるかを判別し，入力から除外する手法であり，敵対的訓練や入力変換と異なり通常サンプルの識別精度を保つことが可能である．しかし，自動運転における標識認識などの入力が必要なタスク [23] においての適用が困難であるという問題点がある．

2.3 先行研究

2.3.1 AE の脆弱性に関する研究

Attack as Defense （A^2D）は，図 2 に示すように，AE の脆弱性，すなわち特徴空間において AE は決定境界付近に位置し，再度攻撃を受けると容易に決定境界を超えて分類結果が変わってしまう特性に着目して検出を行う [4]．この脆弱性から，繰り返し探索を行う反復型の攻撃を用いて入力データに再攻撃を加え，再攻撃のコスト，すなわち識別結果を変化させるために必要な反復回数を調べ，攻撃対象サンプルとは別に用意した訓練サンプル群においてあらかじめ調査した AE と通常サンプルに対する攻撃コストの違いから AE を検出することができる．一方，上記のような検出手法は，AE の検出にのみ焦点を置いており，攻撃前の原画像の正しいクラスの識別等は考慮していない．

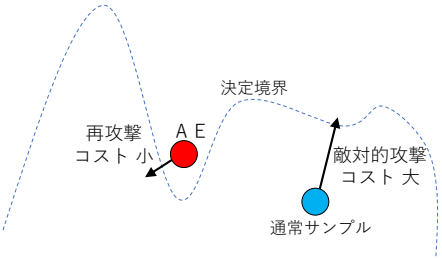

図 2: 特徴空間における敵対的サンプルの位置と脆弱性

2.3.2 AE の矯正に関する研究

上記のような防御手法の問題点に着目し，Kao らは説明可能 AI（eXplainable AI: XAI）を用いた矯正手法を提案し，XAI により推定された注視領域を修正または削除することで，AE を正しいサンプルに戻すことが可能であることを示した [24]．

手法の流れを図 3 に示す．図 3 は提案されたいくつかの手法のうち，Integrated Grad-CAM という説明手法を用いた矯正手法である．Grad-CAM とは，Selevaraju らが提案した DNN モデルの予測根拠の説明手法である [25]．Zhou らが提案した Class Activation Map（CAM）という説明手法には，Global Average Pooling（GAP）層が無いと適用できないという問題点が存在した [26]．そこで，Grad-CAM はモデルの最終畳み込み層までの対象クラスの微分値を用いることで，GAP 層のないモデルに対しても説明手法を適用することが可能となった．この Grad-CAM に対して統合勾配を適用することによって，Integrated Grad-CAM を得る．この手法では，Integrated Grad-CAM を用いて AE の矯正を行う．まず，最終畳込み層を選択し，Integrated Grad-CAM を用いて顕著性マップである注視領域を計算する．得られた注視領域から，元画像を分類するための重要な特徴を妨げることなく，AE における重要な領域を修正する．このとき注視領域は，ランダムに画素を削除するか，Gaussian Blur を用いてランダムに画素をぼかし画素に置換することで画像の修正を行う．

この手法は入力の除外や高コストの計算を必要としないため，既存の防御手法における問題点を解消することができた．しかし，矯正の成功率は攻撃手法に依存して大きく異なる点に問題があった．

3. 提案手法

3.1 キーアイデア

本研究では，防御手法により検出された AE に対して再

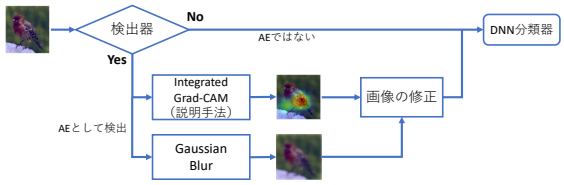

図 3: XAI を用いた敵対的サンプルの矯正手法

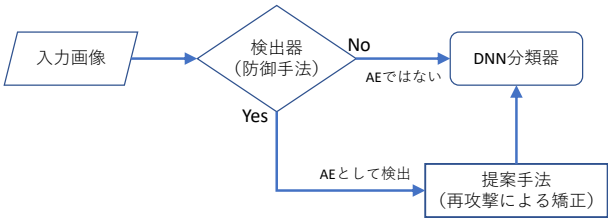

図 4: 提案手法の位置付け

度攻撃を加えることで，原画像の正しいラベルを推定する手法を提案する．すなわち，Kao らによる AE の矯正と同様の目的を，A^{2}D と同様に AE の脆弱性に着目した再攻撃により実現する．本手法は A^{2}D の後処理として位置づけることもできる．

一般に，汎化性能を高めるように学習された分類モデルにおいて，通常サンプルは決定境界から比較的離れている．一方で，人間が知覚できない程度の摂動を付加された AE は決定境界を越えた近傍に位置する．この脆弱性は様々な手法によって生成される AE に共通するため，本手法は多様な攻撃手法に対して適用可能である．タスクに特化した後処理（画像処理用 DNN における平滑化やノイズ付与／除去等）により AE を矯正することも考えられるが，提案手法は DNN の入力データの種別を問わない点に特徴がある．ただし，本稿では画像処理用 DNN を対象としてのみ検証を行うため，音声や自然言語等を対象とする DNN における有用性の検証は今後の課題とする．

3.2 構成と処理手順

提案手法と従来の防御手法との関係を図 4 に示す．本手法は「再攻撃による矯正」の処理を担う．すなわち，本手法は既存の AE 検出手法等により検出された AE に対して再度攻撃を行うことにより，原画像の正しいクラスを推定する．再攻撃に利用可能な手法には特に制限はなく一般的な手法が利用可能である．ただし，FGSM [6] や DeepFool [8] などのホワイトボックス攻撃は，矯正後に利用する DNN モデルの内部情報が利用可能な場合に限られる．言い換えると，商用クラウドなどの内部情報を利用できない DNN を利用する場合は，ブラックボックス攻撃手法を用いて再攻撃を行う必要がある．

3.3 再攻撃手法

提案手法では，任意の攻撃手法を用いて再攻撃を行うことができる．ここでは，先行研究 [24] に倣い，AE の矯正後に利用する DNN モデルの内部情報を利用できると仮定し，FGSM [6]，BIM [7]，DeepFool [8] の 3 手法を再攻撃に用いる場合について説明する．

Algorithm 1 FGSM を用いた再攻撃

Require: 検出された AE $\boldsymbol{x}_{adv}$，最大摂動量 ϵ_{max}，ステップ数 $steps$
Ensure: 再攻撃された AE $\boldsymbol{x}^{'}_{adv}$

```
ε ← 0
for steps do
    ε ← ε + (ε_max / steps)
    x'_adv ← x_adv + ε · sign(∇_{x_adv} L(θ, x_adv, y_adv))
    if C(x'_adv) ≠ C(x_adv) then
        return x'_adv
    end if
end for
```

3.3.1 FGSM を用いた再攻撃

検出された AE $\boldsymbol{x}_{adv}$ に対して，FGSM を用いて再攻撃を行う．

$$\boldsymbol{x}^{'}_{adv} = \boldsymbol{x}_{adv} + \epsilon \cdot sign(\nabla_{\boldsymbol{x}_{adv}} L(\theta, \boldsymbol{x}_{adv}, y_{adv}))$$

ここで，$\boldsymbol{x}^{'}_{adv}$ は再攻撃後の画像，y_{adv} は $\boldsymbol{x}_{adv}$ のラベル，ϵ は摂動量を制御するパラメータ，θ はモデルパラメータ，L は勾配損失，$sign$ は符号関数である．具体的なアルゴリズムを Algorithm 1 に示す．本実験では，敵対的攻撃ライブラリである foolbox [27] の実装にもとづき，入力が誤分類されるまで，ステップ数に応じて摂動量を徐々に増やすこととする．

3.3.2 BIM を用いた再攻撃

検出された AE $\boldsymbol{x}_{adv}$ に対して，BIM を用いて再攻撃を行う．

$$\boldsymbol{x}^{'}_{adv(0)} = \boldsymbol{x}_{adv}, \tag{1}$$

$$\boldsymbol{x}^{'}_{adv(n+1)} = Clip_{\boldsymbol{x}_{adv},\epsilon}\Big(\boldsymbol{x}^{'}_{adv(n)} + \alpha \cdot sign(\nabla_{\boldsymbol{x}_{adv}} L(\theta, \boldsymbol{x}^{'}_{adv(n)}, y_{adv}))\Big) \tag{2}$$

ここで，$\boldsymbol{x}^{'}_{adv(n)}$ は再攻撃後の画像，y_{adv} は $\boldsymbol{x}_{adv}$ のラベル，α はステップサイズ，θ はモデルパラメータ，L は勾配損失，$sign$ は符号関数である．$n = 0 \sim N$ の各反復後，$Clip$ 関数を用いて敵対的サンプルが常に元の入力の ϵ 領域内で最適化されるようにクリップする．

具体的なアルゴリズムを Algorithm 2 に示す．

3.3.3 DeepFool を用いた再攻撃

DeepFool は，非常に小さい摂動において決定境界を線形と見なすことで，直交ベクトルを求め，直交ベクトルに沿って敵対的サンプルを探索する．

分類器 $f(\cdot)$ を線形とみなすと，入力 $\boldsymbol{x}$ について，

Algorithm 2 BIM を用いた再攻撃

Require: 検出された AE $\boldsymbol{x}_{adv}$，摂動サイズ ϵ，ステップサイズ α，反復回数 N
Ensure: 再攻撃された AE $\boldsymbol{x}^{'}_{adv}$

$\boldsymbol{x}^{'}_{adv(0)} = \boldsymbol{x}_{adv}$
for N **do**
  $\boldsymbol{x}^{'}_{adv(n+1)} \leftarrow \boldsymbol{x}^{'}_{adv(n)} + \alpha \cdot sign(\nabla_{\boldsymbol{x}_{adv}} L(\theta, \boldsymbol{x}_{adv(n)}, y_{adv}))$
  $\boldsymbol{x}^{'}_{adv(n+1)} \leftarrow Clip_{\boldsymbol{x}_{adv},\epsilon}(\boldsymbol{x}^{'}_{adv(n+1)})$
end for
return $\boldsymbol{x}^{'}_{adv(N)}$

$$f(\boldsymbol{x}) = \boldsymbol{W}^{\top}\boldsymbol{x} + \boldsymbol{b}$$

となる．ここで，$\boldsymbol{W}$ は重み，$\boldsymbol{b}$ はバイアスである．また，入力 $\boldsymbol{x}$ において，ラベル k に対する分類器 $f(\cdot)$ の出力を $f_k(\boldsymbol{x})$ とすると，予測ラベル $\hat{k}(\boldsymbol{x})$ は，

$$\hat{k}(\boldsymbol{x}) = \arg\max_{k} f_k(\boldsymbol{x})$$

である．

検出された AE $\boldsymbol{x}_{adv}$ に対して，DeepFool を用いて再攻撃を行う．$f(\cdot)$ がラベル $\hat{k}(\boldsymbol{x}_{adv})$ を出力する空間領域の境界のうち，$\boldsymbol{x}_{adv}$ に最も近い超平面を $\hat{l}(\boldsymbol{x}_{adv})$ とすると，

$$\hat{l}(\boldsymbol{x}_{adv}) = \arg\min_{k \neq \hat{k}(\boldsymbol{x}_{adv})} \frac{|f_k(\boldsymbol{x}_{adv}) - f_{\hat{k}(\boldsymbol{x}_{adv})}(\boldsymbol{x}_{adv})|}{\| w_k - w_{\hat{k}(\boldsymbol{x}_{adv})} \|_2}$$

ここで，w_k は重み $\boldsymbol{W}$ の k 番目の要素である．

これにより，$\boldsymbol{x}_{adv}$ を超平面 $\hat{l}(\boldsymbol{x}_{adv})$ に射影するベクトルが最少摂動 $\boldsymbol{r}_*(\boldsymbol{x}_{adv})$ となる．

$$\boldsymbol{r}_*(\boldsymbol{x}_{adv}) = \frac{|f_{\hat{l}(\boldsymbol{x}_{adv})}(\boldsymbol{x}_{adv}) - f_{\hat{k}(\boldsymbol{x}_{adv})}(\boldsymbol{x}_{adv})|}{\| w_{\hat{l}(\boldsymbol{x}_{adv})} - w_{\hat{k}(\boldsymbol{x}_{adv})} \|_2^2}(w_{\hat{l}(\boldsymbol{x}_{adv})} - w_{\hat{k}(\boldsymbol{x}_{adv})})$$

ここで，$f_{\hat{l}(\boldsymbol{x}_{adv})}(\boldsymbol{x}_{adv})$ は $\hat{l}(\boldsymbol{x}_{adv})$ での分類器 $f(\cdot)$ の出力結果である．

具体的なアルゴリズムを Algorithm 3 に示す．

Algorithm 3 DeepFool を用いた再攻撃

Require: 検出された AE $\boldsymbol{x}_{adv}$，反復回数 N
Ensure: 再攻撃された AE $\boldsymbol{x}^{'}_{adv}$

$\boldsymbol{x}^{'}_{adv(0)} \leftarrow \boldsymbol{x}_{adv}$
$i \leftarrow 0$
while $\hat{k}(\boldsymbol{x}^{'}_{adv(i)}) = \boldsymbol{x}_{adv}$ **or** $i < N$ **do**
  for $k \neq \hat{k}(\boldsymbol{x}_{adv})$ **do**
    $w^{'}_k \leftarrow \nabla f_k(\boldsymbol{x}^{'}_{adv(i)}) - \nabla f_{\hat{k}(\boldsymbol{x}_{adv})}(\boldsymbol{x}^{'}_{adv(i)})$
    $f^{'}_k \leftarrow f_k(\boldsymbol{x}^{'}_{adv(i)}) - f_{\hat{k}(\boldsymbol{x}_{adv})}(\boldsymbol{x}^{'}_{adv(i)})$
  end for
  $\hat{l} \leftarrow \arg\min_{k \neq \hat{k}(\boldsymbol{x}_{adv})} \frac{|f^{'}_k|}{\|w^{'}_k\|_2}$
  $\boldsymbol{r}_i \leftarrow \frac{|f^{'}_{\hat{l}}|}{\|w^{'}_{\hat{l}}\|_2^2} w^{'}_{\hat{l}}$
  $\boldsymbol{x}^{'}_{adv(i+1)} \leftarrow \boldsymbol{x}^{'}_{adv(i)} + \boldsymbol{r}_i$
  $i \leftarrow i + 1$
end while
return $\boldsymbol{x}^{'}_{adv(N)}$

4. 評価実験

提案する手法の有効性を検証するために，多様なデータセット，攻撃手法間での検証（実験 1），ならびに先行研究 [24] との比較（実験 2）を行った．

4.1 実験 1：データセット・攻撃手法間の比較

実験 1 では，多様なデータセットと攻撃手法を組み合わせて提案手法の矯正性能を検証する．なお，防御手法により AE が検出されたという前提のもとで，AE の矯正を試みる実験を行った．

本実験では先行研究 [24] に倣い，DNN モデルの内部情報を利用できると仮定し，AE を生成する際の攻撃の種類は，勾配ベースの攻撃から，FGSM [6]，BIM [7]，DeepFool（DF） [8]，CW [9]，JSMA [10] を，スコアベースの攻撃からは LocalSearch（LS） [11]，決定ベースの攻撃からは HopSkipJumpAttack（HSJA） [13] を採用した．これらの攻撃手法は防御評価のガイドライン [14] を基に，類似した手法の採用を避け，広範で代表的な攻撃を選定している．FGSM は，最も強力な攻撃条件である勾配ベースにおける単純な攻撃手法であり，ワンステップで AE を生成できる．一方，BIM は FGSM をより小さなステップで反復探索することで，より強力な AE を生成する．DF は短時間で AE を生成するための最小摂動を探索する．CW は勾配ベースの中でも強力な攻撃であり，高い攻撃成功率を誇り，非常に小さな摂動を実現している．JSMA は他の勾配ベース攻撃手法と異なり，画素単位で摂動を加える．また，勾配ベースとは異なる実験条件を考慮して，スコアベース，決定ベースからそれぞれ LS，HSJA を採用した．

提案手法における再攻撃および，評価実験における AE 生成の攻撃は，foolbox [27] を用いて実装した．foolbox に準じた各攻撃/再攻撃手法のパラメータを表 1 に示す．

入力データは，MNIST [28]，CIFAR-10 [29]，ImageNet（ILSVRC2012） [30] の 3 種類の画像データセットを使用することとし，各データセットにおいて，分類モデルが原画像を正しく識別でき，かつ，敵対的攻撃が成功した 1,000 サンプルを使用した．また，矯正後の AE を識別した結果が原画像と同じになる，すなわち矯正が成功した割合を評価指標とした．分類モデルは，MNIST と CIFAR-10 を対象とした分類器については，先行研究 [24] を基に実装した．モデルの構成と学習パラメータについては，表 2，3，4 に示す． ImageNet を対象とした分類器は PyTorch で提供されている事前学習済のバッチ正則化を含む VGG-19 [31] を用いた．

ここで，予備実験として各攻撃の基本的な性能を検証す

表 1: 攻撃/再攻撃手法のパラメータ

攻撃手法	パラメータ
FGSM	$steps = 1,000$, $\epsilon_{max} = 1.0$
BIM	$\epsilon = 0.3$, $\alpha = 0.05$, $N = 10$
DF	$N = 100$, subsample=10
CW	binary_search_steps=5, max_iterations=1000, confidence=0, learning_rate=0.005, initial_const=0.01
JSMA	max_iter=2000, num_random_targets=0, theta=0.1, max_perturbations_per_pixel=7
LS	r=1.5, p=10.0, d=5, t=5, R=150
HSJA	iterations=64, initial_num_evals=100, max_num_evals=10000, gamma=1.0

表 2: MNIST 分類モデルの構成

Architecture	Output shape
Conv.ReLU	24 × 24 × 16
Conv.ReLU	20 × 20 × 32
Max Pooling	10 × 10 × 32
Conv.ReLU	6 × 6 × 64
Dropout(0.25)	6 × 6 × 64
Linear.ReLU	128
Dropout(0.25)	128
Linear	10

表 3: CIFAR-10 分類モデルの構成

Architecture	Output shape
Conv.ELU	30 × 30 × 64
Conv.BatchNorm.ELU	28 × 28 × 64
MaxPooling	14 × 14 × 64
Dropout(0.25)	14 × 14 × 64
Conv.ELU	12 × 12 × 128
Conv.BatchNorm.ELU	10 × 10 × 128
MaxPooling	5 × 5 × 128
Dropout(0.25)	5 × 5 × 128
Conv.ELU	3 × 3 × 256
Conv.BatchNorm.ELU	1 × 1 × 256
Dropout(0.25)	1 × 1 × 256
Linear.BatchNorm.ELU	1024
Dropout(0.25)	1024
Linear	10

る．各データセットにおいて，敵対的サンプルを 1,000 サンプル生成するまでの攻撃成功率および，生成した 1,000 サンプルの原画像と敵対的サンプル間の平均摂動量について表 5， 6， 7 に示す．このときの摂動量は，L_2 ノルムによって計算される．各攻撃の攻撃成功率から，WB 条件の攻撃は勾配情報を用いて摂動を探索できるため成功率が

表 4: 分類モデルの学習パラメータ

	MNIST	CIFAR-10
Optimizer	Adam	Adam
Learning rate	10^{-4}	10^{-2}
Batch size	128	128
Epochs	99	99
Test accuracy	99.50%	82.60%

表 5: 攻撃手法に関する予備実験（MNIST）

評価値	AE 生成時の攻撃手法						
	FGSM	BIM	DF	CW	JSMA	LS	HSJA
攻撃成功率	0.992	1.000	1.000	1.000	0.898	0.597	0.515
平均摂動量	4.344	2.487	1.801	1.398	2.964	7.147	1.591

表 6: 攻撃手法に関する予備実験（CIFAR-10）

評価値	AE 生成時の攻撃手法						
	FGSM	BIM	DF	CW	JSMA	LS	HSJA
攻撃成功率	0.980	1.000	1.000	1.000	1.000	0.243	0.821
平均摂動量	1.139	0.270	0.196	0.157	0.724	5.756	0.468

表 7: 攻撃手法に関する予備実験（ImageNet）

評価値	AE 生成時の攻撃手法						
	FGSM	BIM	DF	CW	JSMA	LS	HSJA
攻撃成功率	1.000	1.000	0.999	1.000	0.990	0.031	0.814
平均摂動量	1.184	0.235	0.145	0.154	1.243	6.369	23.074

高く，BB 条件の攻撃より強力であることが分かる．また，FGSM の派生である BIM は FGSM よりも平均摂動量が小さい．DF や CW は摂動量をさらに小さくすることが可能であり，画素単位で摂動を与える JSMA と LS は，摂動量が比較的大きくなることが確認できる．

提案手法により再攻撃を行った実験の結果を表 8，9，10 に示す．これらの結果から，いずれのデータセット，攻撃手法においても 90%以上の矯正に成功しており，広範なデータセット，攻撃手法に対して適用できることが示唆された．また，すべてのデータセットにおいて LS に対する矯正成功率が比較的低い値を示した．予備実験より，LS における原画像と AE の摂動量が大きいことから，AE が原画像から離れていることが一つの要因として考察される．このことは本手法の直感的なアイデアである，AE と決定境界との距離が近いことで矯正が可能だということが示唆されている．一方で，MNIST における FGSM や ImgeNet における HSJA の摂動量は大きいものの，矯正成功率は高いため，予備実験で評価している平均摂動量は，特定の敵対的サンプルと決定境界との距離を測る十分な指標とはいえない．矯正に失敗したサンプルに対して個々に摂動量を比較する，画素単位で最大距離を検討できる L_∞ で摂動量を計算するといった，さらなる検証が必要である．

表 8: 実験 1：MNIST を対象とした矯正成功率

矯正時の再攻撃手法	AE 生成時の攻撃手法						
	FGSM	BIM	DF	CW	JSMA	LS	HSJA
FGSM	0.999	0.999	0.978	1.000	0.993	0.938	1.000
BIM	0.998	0.999	0.978	1.000	0.995	0.937	1.000
DF	0.993	0.998	0.944	1.000	0.987	0.939	1.000

表 9: 実験 1：CIFAR-10 を対象とした矯正成功率

矯正時の再攻撃手法	AE 生成時の攻撃手法						
	FGSM	BIM	DF	CW	JSMA	LS	HSJA
FGSM	0.992	1.000	1.000	1.000	0.994	0.911	1.000
BIM	0.992	1.000	1.000	1.000	0.993	0.911	1.000
DF	0.991	0.997	0.999	0.998	0.994	0.913	0.991

表 10: 実験 1：ImageNet を対象とした矯正成功率

矯正時の再攻撃手法	AE 生成時の攻撃手法						
	FGSM	BIM	DF	CW	JSMA	LS	HSJA
FGSM	0.926	0.991	0.999	0.994	0.999	0.981	0.997
BIM	0.919	0.999	1.000	0.992	0.998	0.989	1.000
DF	0.923	0.997	0.997	0.993	0.998	0.987	0.997

4.2 実験 2：先行研究との比較

実験 2 では，XAI を用いた矯正手法 [24] と提案手法の比較を行った．AE を生成する際の攻撃の種類は，FGSM，BIM（L_2, L_∞），CW を採用しデータセットは，MNIST，CIFAR-10 を使用した．再攻撃手法，分類モデル，評価指標は実験 1 と同様である．

結果を表 11 に示す．比較手法の結果は，論文に示された結果のうち条件に合致する結果のなかで最良の結果を引用した．提案手法とは実験に利用したサンプルおよび分類モデルが異なる可能性があることから，厳密な比較ではない点に留意されたい．

Kao らの手法 [24] では，CIFAR-10 において，CW により生成された AE と比較して，FGSM や BIM により生成された AE の矯正の成功率が低下する傾向がみられた．これに対して提案手法では，データセットや攻撃手法の組み合わせによって成功率が大きく変化することなく，すべての攻撃に対して高い矯正性能を示した．先行研究 [24] では，特に CIFAR-10 において，説明手法が誤った解釈をする確率が高く，性能の低下がみられたと考察されている．一方，本手法では AE に共通する脆弱性に着目して矯正を行ったため，データセットや AE を生成する攻撃手法に依存せず，安定した矯正成功率を示したと考える．

表 11: 先行研究 [24] との矯正成功率の比較

データセット	矯正手法	AE 生成時の攻撃手法			
		FGSM (L_∞)	BIM (L_2)	BIM (L_∞)	CW (L_2)
MNIST	先行研究 [24]	0.889	0.949	0.905	0.972
	提案手法（FGSM）	0.999	0.996	0.999	1.000
	提案手法（BIM）	0.998	0.996	0.999	1.000
	提案手法（DF）	0.993	0.992	0.998	1.000
CIFAR-10	先行研究 [24]	0.581	0.616	0.729	0.936
	提案手法（FGSM）	0.992	0.997	1.000	1.000
	提案手法（BIM）	0.992	0.997	1.000	1.000
	提案手法（DF）	0.991	0.995	0.997	0.998

5. 結論

本研究では，検出された AE に対して再度攻撃を行うことで AE を矯正し，攻撃前の原画像の正しい分類結果を得る手法を提案した．実験結果から，提案手法は従来手法と比較して，多様な攻撃方法によって生成された AE をより安定的に矯正できることが示された．今後は，検出器による本手法への影響や，矯正に失敗するサンプルの特徴を調査する．

参考文献

[1] Madry, et al. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.
[2] Xie, et al. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*, 2017.
[3] Dziugaite, et al. A study of the effect of jpg compression on adversarial images. *arXiv preprint arXiv:1608.00853*, 2016.
[4] Zhao, et al. Attack as defense: Characterizing adversarial examples using robustness. In *Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis*, pp. 42–55, 2021.
[5] Szegedy, et al. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.
[6] Goodfellow, et al. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.
[7] Kurakin, et al. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pp. 99–112. Chapman and Hall/CRC, 2018.
[8] Moosavi-Dezfooli, et al. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2574–2582, 2016.
[9] Carlini, et al. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pp. 39–57. Ieee, 2017.
[10] Papernot, et al. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pp. 372–387. IEEE, 2016.
[11] Narodytska, et al. Simple black-box adversarial perturbations for deep networks. *arXiv preprint arXiv:1612.06299*, 2016.
[12] Brendel, et al. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017.
[13] Chen, et al. Hopskipjumpattack: A query-efficient decision-based attack. In *2020 ieee symposium on se-*

curity and privacy (sp), pp. 1277–1294. IEEE, 2020.

[14] Carlini, et al. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

[15] Shafahi, et al. Adversarial training for free! *Advances in Neural Information Processing Systems*, Vol. 32, , 2019.

[16] Meng, et al. Athena: A framework based on diverse weak defenses for building adversarial defense. *arXiv preprint arXiv:2001.00308*, 2020.

[17] Guo, et al. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2017.

[18] Buckman, et al. Thermometer encoding: One hot way to resist adversarial examples. In *International conference on learning representations*, 2018.

[19] Feinman, et al. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017.

[20] Ma, et al. Characterizing adversarial subspaces using local intrinsic dimensionality. *arXiv preprint arXiv:1801.02613*, 2018.

[21] Wang, et al. Dissector: Input validation for deep learning applications by crossing-layer dissection. In *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*, pp. 727–738. IEEE, 2020.

[22] Wang, et al. Adversarial sample detection for deep neural network through model mutation testing. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, pp. 1245–1256. IEEE, 2019.

[23] Cireşan, et al. Multi-column deep neural network for traffic sign classification. *Neural networks*, Vol. 32, pp. 333–338, 2012.

[24] Kao, et al. Rectifying adversarial inputs using xai techniques. In *2022 30th European Signal Processing Conference (EUSIPCO)*, pp. 573–577. IEEE, 2022.

[25] Selvaraju, et al. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

[26] Zhou, et al. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921–2929, 2016.

[27] Rauber, et al. Foolbox: A python toolbox to benchmark the robustness of machine learning models. *arXiv preprint arXiv:1707.04131*, 2017.

[28] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, Vol. 29, No. 6, pp. 141–142, 2012.

[29] Krizhevsky, et al. Learning multiple layers of features from tiny images. 2009.

[30] Olga Russakovsky, et al. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, Vol. 115, No. 3, pp. 211–252, 2015.

[31] Simonyan, et al. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.