White-Box Attack にロバストな Adversarial Detection の提案

山本 恭平 [1,*]　吉野雅之 [1]　冨樫 由美子 [1]

概要：深層学習が様々な分野で活用され，近い将来には自動車システムや医療システム等の人命に関わる分野への導入も期待される．一方，深層学習の普及に伴い、深層学習に対する攻撃の研究も進展を見せており，実システムへの被害が懸念されている．代表的な攻撃に，入力画像に作為的な微小ノイズを加えることで推論モデルの誤判断を引き起こす Adversarial Example Attack がある．他方、その対策手法も研究されており，特に Adversarial Example を検知する Adversarial Detection が知られているが，攻撃者が検知手法に関する情報を有する場合は，検知精度が低下する．本論文では，攻撃者が検知手法に関する情報を有する状況においても、高い検知精度が期待できる Adversarial Detection の実験結果を報告する．

キーワード： 深層学習，Adversarial Example，Adversarial Detection

Proposal of Robust Adversarial Detection for White-Box Attack

Kyohei Yamamoto [1,*]　Masayuki Yoshino[1]　Yumiko Togashi[1]

Abstract: Deep learning is utilized in various fields, and it is expected that it will be introduced into fields related to human life such as automobile systems and medical systems in the near future. On the other hand, with the spread of deep learning, research on attacks on deep learning is also progressing, and there is concern about damage to the system. A typical attack is the Adversarial Example Attack, which causes misjudgment of the inference model by adding artificial noise to the input. On the other hand, countermeasure methods are also being researched, and in particular, Adversarial Detection, which detects Adversarial Examples, is known, but if the attacker has information on the detection method, detection accuracy will decrease. In this paper, we report the experimental results of Adversarial Detection, which can be expected to have high detection accuracy even when the attacker has information on the detection method.

Keywords: Deep Learning，Adversarial Example，Adversarial Detection

1. はじめに

近年 AI 技術の進展に伴い，金融や製造など様々な分野で AI が活用されている．現状，人命等に影響が少ない分野を中心に AI が活用されているが，今後は自動車システムや医療システムなどの人命に深刻な影響を与えうるクリティカルな分野へも AI の導入が進むと期待されている．AI 技術の中でも深層学習は特に注目されており，画像分類等の分野では人を超える認識精度を達成している．しかしながら，深層学習を用いた AI に対する攻撃手法が複数報告されており[1-5]，セキュリティとプライバシ上のリスクが懸念されている．このような状況を受けて，EU の Ethics guidelines for trustworthy AI [6]をはじめとした多くの AI 関連ガイドラインにおいて，信頼できる AI を実現するため，アセスメントを実施した上でセキュリティを考慮することが要求されている．日本では総務省において AI 利活用ガイドライン[7]の策定が進んでおり、本ガイドラインでも AI のセキュリティを考慮することが求められている．

深層学習を用いた推論モデルに対する代表的な攻撃手法として，推論モデルを盗む Model Extraction Attack[1]，学習に利用されたデータを復元する Model Inversion Attack[2]，学習データ等に改変を加えることで推論モデルの精度劣化や推論モデルにバックドアを設置する Data Poisoning [3]，推論モデルが誤った出力をするような入力データを意図的に生成する Adversarial Example Attack [4,5] が知られている．本論文では，Adversarial Example Attack を取り上げる．Adversarial Example とは入力データにノイズを加えることで生成され，推論モデルの誤った出力を引き起こすデータを指す．[4]では，人が知覚できない程微小なノイズをデータに加えるため，人目で Adversarial Example と正常なデータを見分けることは難しい（図 1　Adversarial Example (FGSM Attack，論文[4])）．[5]では，一時停止の交通標識に物理的にシールを張り付けることで 45km/h 制限の標識と誤認識させることに成功しており（図 2 一時停止を 45km/h と誤認識させる Adversarial Example Attack.（論文[5])），この攻撃が悪用されれば，交通事故等の重大な事故を引き起こしかねない．このようなリスクへの懸念から，人命に影響を与えるクリティカルな分野への深層学習の導入は，運用も考慮しながら慎重に進める必要がある．以降，

1 株式会社日立製作所 研究開発グループ セキュリティトラスト研究部
Security & Trust Laboratories, Research & Development Group, Hitachi,Ltd.
* kyohei.yamamoto.sa@hitachi.com

Adversarial Example は画像分類を対象としている研究が多いため，本論文でも画像分類を対象とし，Adversarial Example Attack の中でも図 1 のような微小ノイズを加えて Adversarial Example を生成する攻撃を対象とする．

攻撃手法が研究される一方，対策手法も多く提案されている．Adversarial Example に対する代表的な対策手法に，Adversarial Training [4]と Adversarial Detection [8-19]がある．Adversarial Training は，学習時に Adversarial Example を学習させることで，Adversarial Example に対しロバストな推論モデルを生成する手法であり，対策技術の中で有効な手段の一つである． Adversarial Detection は Adversarial Example 特有の性質を捉えることで検知する手法である。Adversarial Detection は，入力画像や推論モデルの隠れ層の出力に対し主成分分析を行う手法[8,9,10]， Dropout Randomization により推論モデルをランダマイズした際の推論モデルの出力の変化量を検知に用いる手法[11]，入力画像に平滑化を施した際の推論モデルの出力の変化量を検知に用いる手法[12]，入力画像にランダムノイズや悪意のあるノイズを付加した際の推論モデルの出力の変化量を検知に用いる手法[13-18]等がある．Adversarial Detection は，推論モデル自体を改変しないため推論モデルの性能劣化を考慮する必要がないこと，Adversarial Training に比べ計算コストを抑えられることが長所としてあげられる．しかしながら，攻撃者が検知器に関する情報を持っている場合，攻撃者の対策次第で検知性能が落ちてしまうことが Adversarial Detection の課題としてある．本論文では，検知器に関する情報を攻撃者が持っている状況においても高い検知精度を達成する検知手法を提案する．

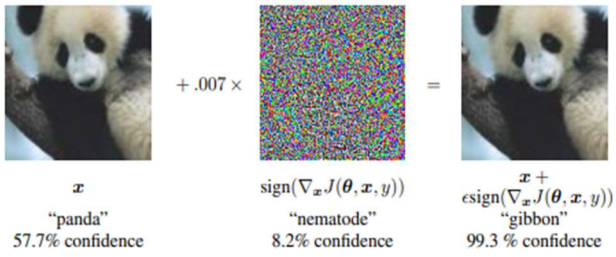

図 1　Adversarial Example (FGSM Attack，論文[4])
(左から正常データ，ノイズ，Adversarial Example)

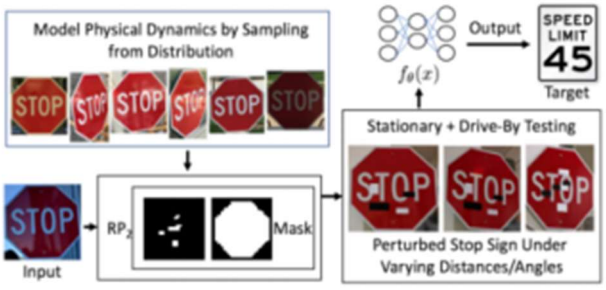

図2 一時停止を45km/h と誤認識させる Adversarial Example Attack.（論文[5]）

2. 関連研究

本章では関連研究について示す．はじめに 2.1 節で Adversarial Example Attack の関連研究について述べ，2.2 節で Adversarial Example Attack への対策技術に関する研究を示す．

2.1 Adversarial Example Attack

Adversarial Example Attack は，攻撃者が推論モデルの出力結果を誤った値へ誘導しようと意図的に作成したデータである Adversarial Example を用いた攻撃を指し, Adversarial Example は入力データに対し微小な摂動を加えることで生成される。有名な事例として，パンダの画像を他の動物に誤分類させる実験が報告されている[4](図 1). 以下, 攻撃者について説明する．入力データに対し確率ベクトルを出力する推論モデルをfとし，fの i 番目の要素をf_iとする．入力xに対し$F(x) = argmaxi\ f_i(x)$ と定め F(x)の値を予測値と呼ぶことにする．このときシンプルな攻撃者の目的は下式である．

$$Find\ \delta\ s.t.\ F(x) \neq F(x^*), x^* = x + \delta$$

Adversarial Example Attack は攻撃方針や状況によって細分化される．まず、攻撃方針により標的型攻撃と非標的型攻撃の 2 種類に分類できる．標的型攻撃は，攻撃者が誤分類を起こしたいクラス $y^* \neq y$ を定め，$F(x^*) = y^*$を満たす x^* を作成する攻撃である．これに対し非標的型攻撃は，入力データのクラスをy^*としたときに, $F(x^*) \neq y^*$ を満たす x*を作成する攻撃である．また、Adversarial Example Attack は、攻撃者の状況により White-Box Attack, Black-Box Attack, Gray-box Attack に分類される。White-box Attack は攻撃者が推論モデルに関する情報を全て保持している状況下で行う攻撃, Black-box Attack は攻撃者が推論モデルに関する情報を保持しておらず，入出力結果のみを元に行う攻撃，Gray-box Attack は攻撃者が推論モデルの情報を一部保持している状況下で行う攻撃である．

これまで Adversarial Example Attack は様々な手法が提案されている．以下では実験に使用する攻撃手法，Fast Gradient Sign Method(FGSM)[4]，C&W 攻撃[20]について説明する。

Fast Gradient Sign Method

FGSM は損失関数に対する勾配を利用する，シンプルかつ高速な攻撃手法である。損失関数Lは入力データ x に対する推論モデルの出力 $f(x)$とラベル y を引数として定義される関数であり，$f(x)$と y の誤差を測る．FGSM では，入力データ x に加えるノイズδは次のように計算される．

$$\delta = \varepsilon \cdot sign(\nabla_x L(f(x), y))$$

ここで, εはノイズの大きさを決定するハイパーパラメータであり，$sign$ 関数は以下で定義される．

$$sign(x) = \begin{cases} 1 & if\ x > 0 \\ 0 & if\ x = 0 \\ -1 & if\ x < 0 \end{cases}$$

図 1　Adversarial Example (FGSM Attack, 論文[4])は FGSM により生成された Adversarial Example である．

C&W 攻撃

C&W 攻撃は，Adversarial Example Attack の中で強力な攻撃の一つである．C&W 攻撃では，Adversarial Example の生成を，以下の最適化問題に帰着させる．

$$\underset{x,\delta}{\text{minimize}} \|\delta\|_p + c \cdot g(x+\delta, y)$$

ここで，c はハイパーパラメータ，$\|\cdot\|_p$はL_pノルム，g は

$$g(x,y) \leq 0 \; iff \; F(x) = y$$

を満たすように定めた関数である．関数 g として，例えば以下のものが使われている．

$$g(x,y) = max(0, \max_{i \neq y}(f_i(x)) - f_y(x))$$

上記最適化問題の近似解が Adversarial Example となる．

2.2 Adversarial Example への対策技術

Adversarial Example に対する対策として様々な手法が提案されている．対策手法は大きく 2 つに分類される．1 つは，Adversarial Example に騙されないように推論モデルのロバスト性を高める手法(Adversarial Training など)，もう 1 つは検知器を設けて，推論モデルへの入力が Adversarial Example であるか否か検知する手法(Adversarial Detection)である．推論モデルのロバスト性を高める代表的な手法として，推論モデルに Adversarial Example を学習させる Adversarial Training [4]や，推論モデルをより小さな推論モデルへ圧縮する distillation [21]等がある．Adversarial Training は，訓練時に Adversarial Example を生成することによる計算量の増大や，Adversarial Example でない通常のデータに対する分類精度の劣化が課題としてあげられる。また Adversarial Training は特定の攻撃に対するロバスト性を向上させるが、想定していない攻撃への耐性が十分でない．

次に Adversarial Detection について述べる．これまで提案されている Adversarial Detection の多くは，入力画像の特徴量や推論モデルの出力結果を用いて検知を行う．推論モデルには直接手を加えないため、Adversarial Training 等の防御手法に比べ、精度劣化を引き起こさない点が長所としてあげられる．しかしながら，検知器に関する情報を持っている攻撃者に対して，検知精度が低下してしまう点が Adversarial Detection の課題としてあげられる．これまで提案された Adversarial Detection には，入力データに平滑化を施し推論モデルの出力の変化から検知する手法[12]や，入力データにランダムノイズや作為的なノイズを付加した際の推論モデルの出力の変化から検知する手法[11]等があるが，いずれの手法も攻撃者が検知器に関する情報を持っているという状況下では検知精度が落ちてしまうことが知られている．

ここで，Adversarial Detection が搭載された推論モデルへの Adversarial Example Attack について考える．攻撃者の目的は，検知器をすり抜ける Adversarial Example を生成することである．つまり，入力データが正常なデータであると判定した場合には 0 を，Adversarial Example であると判定した場合には 1 を出力する検知器を D としたとき，攻撃者の目的は下式である．

$$Find \; \delta \; s.t. \; F(x) \neq F(x^*) \wedge D(x^*) = 0, x^* = x + \delta$$

推論モデルのみに対する Adversarial Example Attack の時と同様に，攻撃者の攻撃方針や状況に応じて，標的型攻撃や Black-Box Attack のように細分化できる．Adversarial Detection が搭載された場合に異なる点は，Adversarial Example Attack で述べた分類に加え，攻撃者が検知器の情報を保持しているか否かに基づいて分類できることである．攻撃者が検知器に関する情報を全て保持している状況下で行う攻撃を検知に関する White-Box Attack，攻撃者が検知器に関する情報を保持しておらず，入出力結果のみを元に行う攻撃を検知器に関する Black-Box Attack, 攻撃者が推論モデルの情報を一部保持している状況下で行う攻撃を検知器に関する Gray-Box Attack と呼ぶことにする．先ほど述べたように，これまで提案されている Adversarial Detection は検知器に関する Gray Attack，Black Attack に対し検知精度が低下することが知られている．本論文では以後，攻撃者は推論モデルに関する White-Box Attack を行うこととする．

3. 既存研究

本章では，本論文と特に関連の深い Adversarial Detection に関する研究について紹介する．

Using Random Noise

高い検知精度を達成している Adversarial Detection の一つにランダムノイズを利用した検知手法がある．この検知手法では，Adversarial Example にランダムノイズを付加した際の推論モデルの出力の変化量が，正常なデータにランダムノイズを付加した際の推論モデルの出力の変化量よりも大きいという性質を利用している．論文[13,14,15]は，いずれもランダムノイズ付加による推論モデルの出力の変化を捉える指標を定義しており，この指標と閾値を比較して Adversarial Example か否か判定している．いずれの手法も非常に高い精度で Adversarial Example を検知しており，また論文[13]では，想定していない様々な Adversarial Example Attack に対しても高精度で検知可能であることを示している．上記ランダムノイズを用いた検知手法に対して，検知器の情報を持つ攻撃者が，検知されない Adversarial Example を作成できることから，論文[16]では対策として Adversarial Example を生成する際に使用されるノイズ(Adversarial Noise)をランダムノイズの代わりに用いて検知を行っている．これにより，ランダムノイズでは検知できなかった Adversarial Example の検知が可能になる．また，論文[17]では，正常なデータはランダムノイズに対しロバ

スト性があるが Adversarial Noise で推論モデルの出力が変化しやすい．逆に Adversarial Example はランダムノイズで推論モデルの出力が変化しやすく, Adversarial Noise に一定のロバスト性があると実験で示しており，この排他的な性質を利用して，ランダムノイズと Adversarial Noise による 2 ステップの検知を行っている．この手法により，ランダムノイズ単体の検知手法に比べ，検知器に関する White-Box Attack への耐性が向上している．最後に論文[18]では，Adversarial Example にランダムノイズを加えると推論モデルの出力が変化しやすいという性質に加え，正常データとそれに類似したデータに同じランダムノイズを加えた時の推論モデルの出力の変化は正の相関が強く，Adversarial Example とそれに類似したデータに同じランダムノイズを加えた時の推論モデルの出力の変化は依存関係が弱いと予測して検知器を作成している．

4. 提案手法

3 章で紹介したように，これまでランダムノイズを用いた検知手法は多く提案されているが，検知手法に関する White-box Attack に対し脆弱である．そこで本章では，新たな仮説を元に設計したアルゴリズムを用いて，検知器に関する White-box 攻撃に対しロバストな検知手法を提案する．

本論文では，2 ステップにより構成される Adversarial Detection を提案する。ステップ 1 では，前章で述べたランダムノイズによる検知手法を単純化した「入力データにランダムノイズを加えた際の予測ラベルの変化率」を利用する検知手法である。ステップ 2 では,「予測ラベルが変化するまでノイズを乗せ，これにより生成されたデータの予測ラベルの偏り」を利用する検知手法である．ステップ 2 は「Adversarial Example の予測ラベルの変化先は偏りやすい」という仮説のもと設計している(図 3).

本手法の概要について説明する．図 4 は画像空間のイメージ図である．白い部分が正常データ，赤い部分が Adversarial Example に対応している．正常データと Adversarial Example の境界を決定境界と呼ぶことにする．3 章で紹介したように，正常データへのランダムノイズ付加による予測ラベルの変化は起こりにくいのに対し，Adversarial Example へのランダムノイズ付加により予測ラベルは変化しやすい．このことから，正常データの連結した空間は Adversarial Example の連結した空間より大きいとして図 1 に反映している．また，本手法で新たに提案する仮説は，Adversarial Example へのノイズ付加による予測ラベルの変化先は偏りやすい(Adversarial Example 生成前のラベルに戻りやすい)というものである．イメージを図 4 に示す．左側のネコの Adversarial Example にランダムノイズを加えると，ネコという予測結果に戻りやすいという仮説である．この仮説を，Adversarial Example は正常データの空間(図 3 の場合，ネコの画像の空間)に埋め込まれていると予測して図 4 に反映している．ステップ 1 では，図 4 の Adversarial Example に摂動を加えると決定境界を越えやすいことを検知に利用する．ステップ 2 では，Adversarial Example に摂動を加えると正常データの空間に戻りやすく，正常データに摂動を加えると異なる領域の Adversarial Example に変化しやすいことを検知に利用する．

提案手法について詳しく説明する．ステップ 1 のランダムノイズによる検知手法では，微小のランダムノイズにより入力データが決定境界を越えやすい場合に，入力データは Adversarial Example であると判定する．つまりランダムノイズによる検知手法は決定境界付近のデータを検知する．Adversarial Example は正常データに微小ノイズを加えて作成されるデータであることから決定境界に近いと考えられ，そのためランダムノイズによる検知手法によって効率的に検知できる．ただし，攻撃者が意図的に決定境界から遠い Adversarial Example を生成した場合，検知精度が低下することが問題として残る．

次に，ステップ 2 の予測ラベルの偏りを用いた検知手法では，予測ラベルの偏りが大きい場合に入力データは Adversarial Example であると判定する．Adversarial Example は，正常データに作為的な微小ノイズを加えて作成されるデータであるため，ランダムノイズの付加により予測ラベルが元の正常データのラベルに戻りやすい傾向があると期待できる．この性質を利用することで，予測ラベルの偏りを用いた検知手法により Adversarial Example を効率的に検知できると考えた．ただし，攻撃者が正常データのラベルと異なるラベルに変化しやすい Adversarial Example を作成できた場合，検知精度が低下する恐れがある．このようなデータの単純な構成方法は，Adversarial Example 同士の決定境界に近いデータを作成することである(例えば, ネコの画像から作成された，ネコ 10%，イヌ 45%，ウマ 40%と予測を出す Adversarial Example)．決定境界から遠く，かつステップ 2 の検知器をすり抜ける Adversarial Example を作成する事は困難であると予想する．以上，述べたことを表 1 ○は検知されやすいデータ，×は検知されづらいデータにまとめている．2 つの検知器を使うことで，決定境界に近い Adversarial Example, 決定境界から遠い Adversarial Example 両方の検知が可能となり, 検知に関する White-box Attack に対し高い検知精度を維持可能であると期待される．

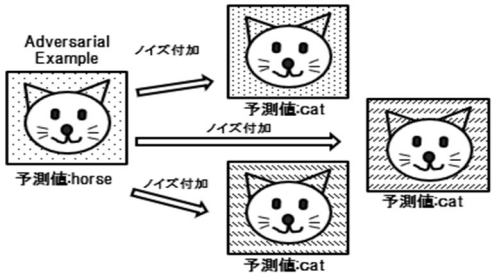

図 3 Adversarial Example の予測の変化先が偏るイメージ図

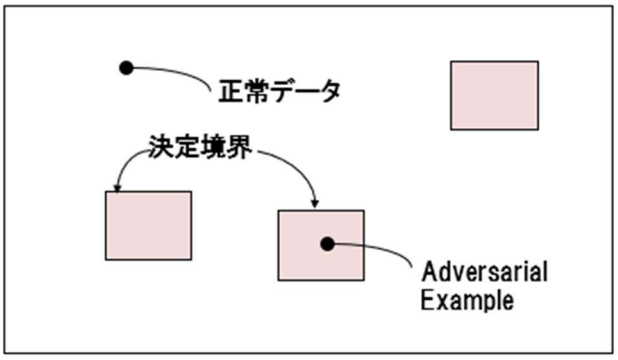

図 4　画像空間のイメージ図

	決定境界から遠い 正常データ	決定境界に近い 正常データ	決定境界から遠い Adversarial Example	決定境界に近い Adversarial Example
ランダムノイズによる検知	×	△	×	○
予測ラベルの偏りによる検知	×	△	○	△

表 1 ○は検知されやすいデータ, ×は検知されづらいデータ，△は中間的な性質のデータ

5. 実験・評価

本章では提案手法の検証結果について示す. 5.1 節で実験に使用するデータセット，推論モデル等の前提条件について，5.2 節で検証結果を記す.

5.1 前提条件

データセット

検証に用いたデータセットは，手書き数字の画像データセット MNIST と，犬や車など 10 種類の画像からなるデータセット CIFAR10 である．MNIST には 60,000 枚の学習用データと，10,000 枚のテスト用データが含まれており，CIFAR10 には 50,000 枚の学習用データと, 10,000 枚のテスト用データが含まれている.

推論モデル

推論モデルは公開されているプログラムを使用した(https://github.com/aaron-xichen/pytorch-playground)．MNIST では 3 層の全結合モデルを使用し，2 層目と 3 層目の入力前に Dropout を行う．Dropout 率は 0.2 とした．CIFAR10 では 7 層の畳み込み層と最終層の全結合層からなるモデルを使用し，各畳み込み層の後に Batch Normalization を行う．いずれのモデルも活性化関数として ReLU 関数を使用している．実験で使用した推論モデルのテストデータに対する認識精度は，MNIST では 98.310%，CIFAR10 では 91.470%である.

攻撃手法

Adversarial Example の生成手法として, FGSM と C&W 攻撃を用いた．MNIST に対しては FGSM（$\varepsilon = 0.01, 0.05$）と C&W 攻撃の 3 通りの方法で，CIFAR10 に対しては FGSM（$\varepsilon = 0.01$）と C&W 攻撃の 2 通りの方法で Adversarial Example を生成する. C&W 攻撃は公開されているプログラムを使用した(https://github.com/kkew3/pytorch-cw2) .

検知手法

4 章で述べたように，ステップ 1「入力データにランダムノイズを加えた際の予測ラベルの変化率」，ステップ 2「予測ラベルが変化するまでノイズを乗せ，これにより生成されたデータの予測ラベルの偏り」という，2 ステップによる検知を行う．以降，アルゴリズムの表に沿って提案手法を説明していく.

初期設定

N:入力画像のピクセル数.

F:入力データに対し予測ラベルを返す実数値関数(2.1 節).

n:Step1 におけるノイズ付加による画像生成枚数.

τ_i:Step i で Adversarial Example の判定に使用する閾値.

max:Step2 におけるノイズ付加回数の上限.

Step1

入力データ x を受け取り，x を Adversarial Example と判定した場合は 1 を，Adversarial Example ではないと判定した場合は 0 を出力するアルゴリズムである．はじめにノイズを n 個生成し，入力データ x にそれぞれ加算することで n 枚のデータ x_i を生成する(line3,4)．次に各データの予測値 $F(x_i)$ を求め，ラベルの変化した枚数を数える(line5,6). 最後にラベルの変化した割合 $\frac{count}{n}$ と閾値 τ_1 を比較して，x が Adversarial Example，であるか判定する.

Step2

Step1 で生成された n 枚の入力データを受け取り，Step1 の入力データ x を Adversarial Example と判定した場合は 1 を，Adversarial Example ではないと判定した場合は 0 を出力するアルゴリズムである．はじめに，各データ x_i に予測値が変化するまでノイズを加える(line5-8)．ただしノイズ付加回数が *max* に到達した場合は打ち止めとする．次に予測値が変化したデータの変化先を集計し，最も多い変化先へ移動したデータの割合 $\frac{\max(List)}{count\_1}$ と閾値 τ_2 を比べて，x が Adversarial Example であるか判定する(line12)．ここで *max(・)*は配列中の最大値を抜き出す関数である.

Main

Step 1 ,Step2 を用いた提案手法のアルゴリズムである．入力データ x が Adversarial Example であるか判定結果を出力する．はじめに入力データ x を Step1 に入力し，出力 $result, x_1, .., x_n$ を受け取る．$result = 1$ であれば $result = 1$ を Main の出力としアルゴリズムを終了する．$result = 0$ であれば, $x_1, \ldots, x_n$ を Step2 に入力して出力 $result$ を受け取り，result を Main の出力としてアルゴリズムを終了する．result=1 は入力データが Adversarial Example と判定したこと，result=0 は入力データが正常データと判定したことを指す.

Main

Input: $x \in \mathbb{R}^N$
Output: $result \in \{0, 1\}$
1. $result, x_1, \ldots, x_n = Step1(x)$
2. *if* $result == 0$:
3. $\quad result = Step2(x_1, \ldots, x_n)$;
4. *return* $result$;

Step1

Input: $x \in \mathbb{R}^N$
Output: $result \in \{0, 1\}, x_1, \ldots, x_n \in \mathbb{R}^N$
Parameter: τ_1
1. $count = 0; result = 0;$
2. *for* i *in* $\{1, \ldots, n\}$:
3. $\quad$ *Generate noise* $\delta \in \mathbb{R}^N$;
4. $\quad x_i = x + \delta$;
5. $\quad$ *if* $F(x_i) \neq F(x)$:
6. $\quad\quad count \mathrel{+}= 1$;
7. *if* $\frac{count}{n} > \tau_1$:
8. $\quad result = 1$;
9. *return* $result, x_1, \ldots, x_n$

Step2

Input: $x_1, \ldots, x_n \in \mathbb{R}^N$
Output: $result \in \{0, 1\}$
Parameter: τ_2
1. *List=[0,…,0]; (Listの長さ)=(label数)*
2. *count_1=0;*
3. *for* i *in* $\{1, \ldots, n\}$:
4. $\quad count\_1 = 0;\ x_i^* = x_i$;
5. $\quad$ *While* $F(x_i^*) \neq F(x_i)$ *and* $count\_2 < max$:
6. $\quad\quad$ *Generate noise* $\delta \in \mathbb{R}^N$;
7. $\quad\quad x_i^* = x_i^* + \delta$;
8. $\quad\quad count\_2 \mathrel{+}= 1$;
9. $\quad$ *if* $F(x_i^*) \neq F(x_i)$:
10. $\quad\quad List(F(x_i^*)) \mathrel{+}= 1$;
11. $\quad\quad count\_1 \mathrel{+}= 1$;
12. *if* $\frac{\max(List)}{count\_1} > \tau_2$:
13. $\quad result = 1$;
14. *return result;*

5.2 実験

ステップ 2 における仮説「Adversarial Example の予測値の変化先に偏りがある」の検証と，Main アルゴリズムの検知精度評価を行う．

ノイズの生成

検知に使用するノイズとして，ガウス分布や一様分布等の確率分布に従う生成方法や，論文[14,16]で記されている Adversarial Noise を用いる方法が考えられる．本実験ではガウス分布に従いノイズを生成する．

閾値τ_1, τ_2の設定

実験の中で，高い検知精度を達成する閾値をτ_1, τ_2として設定する．Adversarial Example の検知精度と，正常データを Adversarial Example と判定する誤検知率の間にはトレードオフがあり，閾値はこれらの値を調整する役割を持つ．

5.3 評価

仮説検証

ステップ 2 における仮説「Adversarial Example の予測ラベルの変化先は正常データと比べ偏りがある」について検証する．図 5 は MNIST の推論モデルに対し，各攻撃手法で Adversarial Example を 5000 枚ずつ生成した時に，正常データとそれぞれの Adversarial Example に対し，Step2, line12 の値$\frac{\max(List)}{count_1}$に偏りが出るか図示したものである．Step1，Step2 における付加ノイズは，いずれも平均 0，分散 0.4 のガウス分布に従い生成し，Step1 における画像生成枚数 *n=100*, Step2 におけるノイズ付加回数の上限 *max=50* とする．図 5 の横軸は$\frac{\max(List)}{count_1}$，縦軸は横軸の値をとる画像の枚数（を正規化したもの）であり，$\frac{\max(List)}{count_1}$が大きいほど予測ラベルの変化先に偏りがあることになる．図 5 では，正常データ(赤)が Adversarial Example に比べ左側に寄っており，このことから仮説通り，Adversarial Example の予測ラベルの変化先は正常データに比べ偏りやすいことが分かる．

CIFAR10 の推論モデルに対する実験を図 6 に示す. Step1 における付加ノイズは，平均 0，分散 0.05 のガウス分布，Step2 における付加ノイズは，平均 0，分散 0.5 のガウス分布に従い生成する．Step1 における画像生成枚数 *n=100*, Step2 におけるノイズ付加回数の上限 *max=50* とする．図 6 より, CIFAR10 においても正常データが左側に寄っており，仮説通りの傾向が表れていることが分かる．特に CIFAR10 の場合は，多くの Adversarial Example の予測ラベルの変化先は，全て元のラベルに戻っており(横軸 1.0 の部分)．仮説と同じ傾向が強く現れている．

検知精度

上記で検証した性質を用いた提案検知手法の検知精度を評価する．MNIST に対する検知精度の結果は表 2 である．パラメータは *n=50,max=50* と定め，付加ノイズは図 5 を作成した時と同様に設定している．閾値τ_1, τ_2は，MNIST に対する Adversarial Example(FGSM0.01: 717 枚 FGSM0.05:1000 枚 C&W 攻撃:1000 枚)を用いて実験を行い，高い検知精度を達成するように設定した．表 2 の実験では，閾値の設定のときと異なるデータから生成した Adversarial Example(FGSM0.01:717 枚　FGSM0.05:1000 枚 C&W 攻撃:1000 枚)を使用している．表 2 より，ステップ

2 を組み合わせることでより多くの Adversarial Example を検知できていることが分かった．特にステップ 1 ではほとんど検知できなかった C&W 攻撃による生成データを，ステップ 2 により検知可能となった．CIFAR10 に対する検知精度は表 3 である．図 6 から Step2 のみで検知精度が出ると見込めたため，Step2 のみで検知を行った．パラメータ *n=50,max=50* は事前に定め，付加ノイズは図 6 を作成した時と同様に設定している．閾値の設定は MNIST のときと同じように，高い検知精度となるように設定し，τ_2=0.9 とした．表 3 より，C&W 攻撃に対し一定の検知精度はあるものの，FGSM 攻撃に対する検知精度が低くなっていることが分かった．また，MNIST，CIFAR10 いずれの検知においても，正常データを Adversarial Example と誤検知する確率が 20～30%であるため，誤検知率の高さが課題として残る．

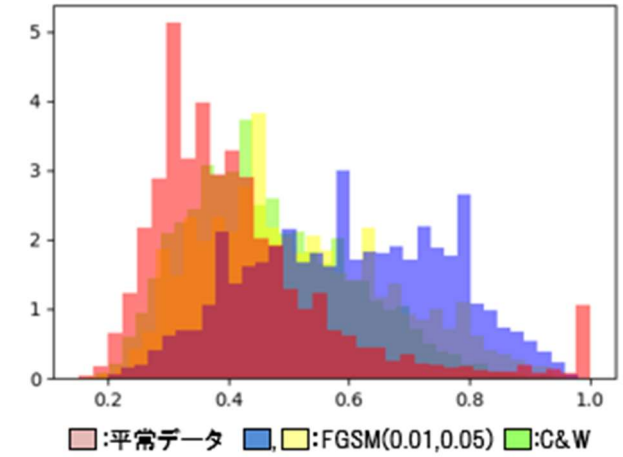

図 5 予測ラベルの変化先の偏り(MNIST)

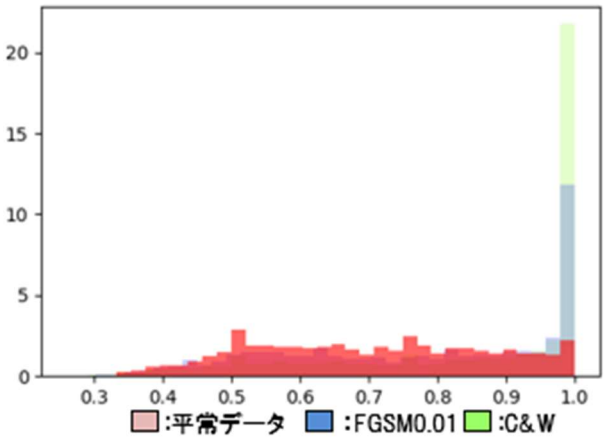

図 6 予測ラベルの変化先の偏り(CIFAR10)

MNIST	Params:$\tau_1 = 0.7, \tau_2 = 0.45, n = 50, max = 50$		
Adversarial Example	Step1の検知枚数	Step2の検知枚数	検知精度
CW(1000枚)	103	421	52.4%
CW+FGSM0.05+FGSM0.01 (999+1000+671=2670枚)	1357	624	74.2%

表 2 MNIST に対する検知精度

CIFAR10	Params:$\tau_2 = 0.9, n = 50, max = 50$	
Adversarial Example	Step2の検知枚数	検知精度
CW(1000枚)	585	58.5%
CW+FGSM0.01 (1000+1000=2000枚)	1084	50.4%

表 3 CIFAR10 に対する検知精度

6. まとめ

ランダムノイズによる検知と予測ラベルの偏りによる検知，2 種類の検知手法を組み合わせることで，検知器に関する White-box Attack に対しロバストな Adversarial Detection 手法を提案した．仮説「Adversarial Example の予測ラベルの変化先に偏りがある」の正当性を実験により確認し，この性質を利用した検知手法とランダムノイズによる検知手法を組み合わせることで Adversarial Example の検知器を作成した．しかしながら，正常データの誤検知率が上がることや計算量が増大することが難点としてある．今後は，検知精度の高さと誤検知率の低さを両立と計算量の削減に向けて，ノイズ生成方法や，アルゴリズムの改良，Step1 に用いる検知手法の検討を行う必要がある．提案検知手法が，検知器に関する White-box Attack に対しロバストであるかの評価は今後の課題とする．

参考文献

[1]. F. Tram`er, F. Zhang, et al. Stealing Machine Learning Models via Prediction APIs. In USENIX Security Symposium, 2016.

[2]. M. Fredrikson, S. Jha, and T. Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In CCS, 2015.

[3]. T. Gu, B. Dolan-Gavitt, and S. Garg. BadNets: Identifying vulnerabilities in the machine learning model supply chain. arXiv preprint arXiv:1708.06733, 2017.

[4]. I. Goodfellow, J.Shlens and C. Szegedy. Explaining and Harnessing Adversarial Examples. arXiv preprint arXiv:1412.6572, 2014.

[5]. K. Eykholt, I. Evtimov, E. Fernandes, et al. Robust physical-world attacks on deep learning visual classification. In IEEE, 2018

[6]. European Commission， Ethics guidelines for trustworthy AI, https://www.aepd.es/sites/default/files/2019-12/ai-ethics-guidelines.pdf ， 2019.

[7]. 総務省，AI 利活用ガイドライン，https://www.soumu.go.jp/main_content/000624438.pdf，2019.

[8]. D. Hendrycks and K. Gimpel. Early Methods for Detecting Adversarial Images. In ICLR 2017

[9]. A. N. Bhagoji, D. Cullina, and P. Mittal. Dimensionality Reduction as a Defense against Evasion Attacks on Machine Learning Classifiers. arXiv preprint arXiv:1704.02654, 2017.

[10]. X. Li and F. Li, Adversarial Examples Detection in Deep Networks with Convolutional Filter Statistics. arXiv preprint

arXiv:1612.07767, 2016.

[11]. R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner.Detecting Adversarial Samples from Artifacts. arXiv preprint arXiv:1703.00410, 2017.

[12]. W. Xu, D. Evans, Y. Qi Feature squeezing: Detecting adversarial examples in deep neural networks. In NDSS, 2018.

[13]. Bo Huang, Yi Wang and Wei Wang. Model-Agnostic Adversarial Detection by Random Perturbations. In International Joint Conference on Artificial Intelligence, 2019.

[14]. K. Roth, Y. Kilcher, and T.Hofmann. The Odds are Odd: A Statistical Test for Detecting Adversarial Examples. In ICML 2019.

[15]. J. Wang, J. Sun, P. Zhang, and X. Wang. Detecting Adversarial Samples for Deep Neural Networks through Mutation Testing. arXiv preprint arXiv:1805.05010, 2018.

[16]. 高橋知克, 山田真徳, 山中友貴, 岩田具治. Adversarial noise を用いた Adversarial Example の検知. In DEIM Forum 2020.

[17]. S. Hu, T. Yu, C. Guo, et al. New Defense Against Adversarial Images: Turning a Weakness into a Strength. In Advances in Neural Information Processing Systems, pp. 1635–1646, 2019.

[18]. Y. Wu, SS. Arora, et al. Beating Attackers At Their Own Games: Adversarial Example Detection Using Adversarial Gradient Directions. arXiv preprint arXiv:2012.15386, 2020.

[19]. N. Carlini, D. Wagner. Adversarial Examples Are Not Easily Detected: By passing Ten Detection Methods. arXiv preprint arXiv:1705.07263, 2017.

[20]. N. Carlini, D. Wagner. Towards evaluating the robustness of neural networks. IEEE, 2017.

[21]. G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In NIPS, 2014.