# Adversarial noiseを用いたAdversarial Exampleの検知

高橋　知克† 　山田　真徳† 　山中　友貴† 　岩田　具治††

† NTT セキュアプラットフォーム研究所　〒 180–8585 東京都武蔵野市緑町 3–9–11
†† NTT コミュニケーション科学基礎研究所　〒 619–0237 京都府相楽郡精華町光台 2–4
E-mail:
†{tomokatsu.takahashi.wd,masanori.yamada.cm,yuuki.yamanaka.kb,tomoharu.iwata.gy}@hco.ntt.co.jp

**あらまし**　深層学習は入力に作為的に作られた微小のノイズを乗せて作られる Adversarial Example によって，攻撃者の思い通りに出力を操作される．これを防ぐためにランダムノイズを用いた検知手法が提案された．既存手法は Adversarial Example が深層学習モデルの決定境界の突出した部分に存在するので，ランダムノイズを加えた時に境界をこえやすい特性を利用して，高精度に検知を行う．しかしながら，一定数存在する境界の突出した部分以外に存在する Adversarial Example に対しては精度が低下する．そこで，本稿では Adversarial Example が非常に小さなノイズによって境界を超えるような変換を起こし誤分類を引き起こすため，通常のデータより決定境界の近くに存在すると仮定し，決定境界方向の変換を効率的に起こす Adversarial noise を用いた検知手法を提案した．画像データを用いた実験により，提案手法が既存手法では検知が難しい決定境界の突出部分以外に存在する Adversarial Example を検知できることを示す．さらに提案手法の性能解析を行い，その有用性について検討する．

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

## 1 はじめに

深層学習は人間の神経細胞の仕組みを模した機械学習の手法の一つであり，画像処理 [1] や音声認識 [2]、サイバーセキュリティ [3,4] などの幅広い領域で活用されている．また，近年では様々なサービスが登場しており，画像分類など分野によっては人間を超えた性能を達成していることから，深層学習は今後ますます重要となる技術である．

しかしながら，近年では深層学習のモデル自体を標的とした攻撃である Adversarial Example [5,6] の危険性が示唆されている．Adversarial Example とは入力データに対して作為的に作られた微小のノイズを加えて作られた悪性データであり，攻撃者の意図するように深層学習の出力を操作および撹乱する．特に画像分類の分野では人間には不可視のノイズを乗せることで，画像の見た目を変えることなくモデルの誤分類を引き起こすことから注目されており，Adversarial Example の研究の多くは画像分類を対象とする．したがって，本稿でも画像分類について扱う．Adversarial Example は深層学習を利用したサービス全ての脅威となるだけではなく，特に自動運転車やマルウェア検知などのモデルの出力の信頼性が求められる分野への深層学習の適用を困難にする．したがって，深層学習を Adversarial Example から守る技術が必要である．

Adversarial Example に対する対策技術は今までに様々なものが提案されているが，その中でも代表的な手法は Adversarial Training [7] と Adversarial Detection [8–15] の二つである．Adversarial Training は訓練データの中に Adversarial Example を混ぜて学習を行うことで，Adversarial Example にロバストなモデルを訓練する手法である．この手法は Adversarial Example の対策技術の中でも最も安定した性能を持ち，様々な研究がされている主流の手法である．しかしながら，学習によってモデルのパラメータが更新されるために訓練用の Adversarial Example を新たに生成しなくてはならないため，計算コストが通常の学習の数十倍になる問題がある．また，最近では Adversarial Example に対するロバストな分類性能と Adversarial Example でない普通のデータである Clean Sample に対する通常の分類性能の間にはトレードオフが存在し，本来の深層学習の分類性能が落ちること [16] や学習データに入っていないデータに対してロバスト性能が汎化しない [17] などの欠点が明らかになった．

一方で，Adversarial Detection は Adversarial Example 自体の特徴 [13] や深層学習に入力した時の特異な挙動 [11,12,14,15,18] を用いて，検知を行うことで Adversarial Example がモデルに入力されることを事前に防ぐ手法である．特定のパターンの Adversarial Example にしか精度が出なかったり，攻撃者の対策によって検知性能が落ちるといった欠点は存在するが，最近では高精度に Adversarial Example を検知する手法が提案されていることに加えて，Adversarial Training と比較して計算コストが低く，モデル自体に手を加えないため本来の性能を落とさないといった利点があるので注目されている．

Adversarial Detection の中でも，ランダムノイズを用いた手法 [11,12,15] は Adversarial Example にランダムノイズを乗せた時に深層学習の出力が大きく変化するという特徴を利用した手法である．この手法では攻撃者が Adversarial Example を生成する際，画像の見た目を変えずに誤分類を起こすため，微小のノイズによってモデルの決定境界を超えるような小さな変換が行われることで，Adversarial Example が決定境界の真のクラス側に突出している部分に生成されやすいと仮定し

た．そして，決定境界の突出した部分に生成された Adversarial Example はランダムノイズを加えた時に決定境界を超えるような出力変化が起こりやすいという特徴を利用して検知を行う。特に 2019 年に Roth らによって提案された手法は高い検知性能を達成し，Adversarial Training での対処が難しい高次元データに対しても高い精度で検知を行う．

しかしながら，ランダムノイズを用いた手法は決定境界のより内側に存在するような，突出部分以外に生成された Adversarial Example に対してはランダムノイズを加えても決定境界を超えるような出力変化が起こりにくく，検知精度が下がるという問題が存在する．こういった突出部分にない Adversarial Example は一定数あると考えられる．なぜなら，Adversarial Example 生成時に加えるノイズの大きさをどこまで許すかは攻撃者次第であり，ノイズが大きいと当然決定境界を大きく超えた場所に Adversarial Example が生成される場合があるからである．また，攻撃者がランダムノイズを用いた手法を知っていた場合，意図的にノイズを大きくし，検知を妨害してくることも考えられる．したがって，突出部分以外に存在する Adversarial Example への精度改善は重要な課題である．

先に述べた課題を解決することを目指し，本稿では Adversarial noise を用いた検知手法を提案した．Adversarial Example が決定境界の突出部分に存在しない場合でも，画像の見た目を変更しないという制約上 Clean Sample と比べると決定境界に近い位置に Adversarial Example が存在するのではないかと考え，Adversarial noise をランダムノイズの代わりに用いた．Adversarial noise とは決定境界方向への変換を意図的に起こすノイズであり，元々Adversarial Example が生成される時に加えられるノイズと同様のものである．提案手法はランダムノイズの代わりに決定境界方向への変換を起こしやすい Adversarial noise を用いることで，既存手法では検知できない突出した部分にないような Adversarial Example でも決定境界に近ければ，その境界を超えることができる．したがって，既存手法では検知精度が落ちるような決定境界の内側に存在するパターンの Adversarial Example も高精度に検知することができると考えた．

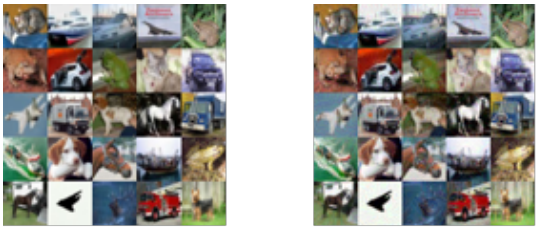
最後に本稿の構成をまとめる．まず，2 節において本稿の前提知識について説明する．続いて 3 節にて，提案手法である Adversarial noise を用いた検知手法について概説する．そして，実験による提案手法の評価と考察を 4 節で行う．最後に 6 にて結論を述べる．

## 2 前 提 知 識

本節では，前提知識となる事柄について概説する．最初に 2.1 節で Adversarial Example とその代表的な攻撃手法について述べた後，2.2 節において本稿における既存手法であるランダムノイズを用いた検知手法について説明する．

### 2.1 Adversarial Example

Adversarial Example [5] は攻撃者によって意図的に深層学習の誤分類を引き起こすように作られた悪性データであり，入力データに対して作為的に微小のノイズを乗せることで作成される．Adversarial Example の例を図 1 に示す．図 1 からわかるように人間の目では通常のデータである Clean Sample と Adversarial Example の見分けがつかないが，深層学習に入力すると Adversarial Example は本来のクラスとは違うクラスに誤分類される．



(a) Clean Sample　　(b) Adversarial Example

図 1　Adversarial Example の例

また，Adversarial Example は攻撃方針によって，target attack と non-target attack に分類される．target attack は攻撃者が誤分類したい特定のターゲットクラス $y^{target}$ に向けて誤分類するような攻撃を行う．一方で，non-target attack は入力データ $x$ の元のクラス $y^{true}$ から離れるように攻撃が行われる．本稿では，non-target attack を対象とする．

ここで，実際に Adversarial Example を作る方法について概説する．Adversarial Example の作成方法は今までに様々なものが提案されているが，最も代表的であり本稿でも利用した手法である FGSM と PGD について説明する．

**FGSM [6]:**FGSM はモデルの誤差関数に対する勾配を利用する代表的な攻撃手法の一つである．一般に深層学習の学習はデータ $x$ を入力した時の深層学習の出力 $F(x)$ が正解ラベル $y^{\mathrm{true}}$ とどれくらいズレているか計算する誤差関数 $L(F(x), y^{\mathrm{true}})$ を最小化するようにモデルのパラメータを変化させることだが，FGSM で Adversarial Example を作る際にはではパラメータを変更せず，代わりに入力データ $x$ 自身を変化させることで誤差関数の値を操作する．FGSM は以下の式 1 で表される

$$x^{\mathrm{adv}} = x + \epsilon * \mathrm{sign}(\nabla \mathrm{x}) \tag{1}$$

ここで，$\nabla x$ は以下である．

$$\begin{cases} \nabla x = \dfrac{\partial L(F(x), y^{\mathrm{true}})}{\partial x} & (\text{if non-target}) \\ \nabla x = -\dfrac{\partial L(F(x), y^{\mathrm{target}})}{\partial x} & (\text{if target}) \end{cases} \tag{2}$$

ここで，$x^{\mathrm{adv}}$ は $x$ を元にした Adversarial Example，$\epsilon$ は $x$ に乗せる作為的なノイズの大きさである．

**PGD [19]:**PGD は FGSM を拡張した非常に強力な攻撃手法である．FGSM では勾配方向に $\epsilon$ 分のノイズをまとめて乗せていたが，PGD ではステップサイズ $a$ ごとに，Adversarial Example を繰り返し更新する．PGD は以下の式 4 で表される．

$$x_0^{\mathrm{adv}} = x \tag{3}$$

$$x_{i+1}^{\mathrm{adv}} = Clip_{(x+\epsilon,x-\epsilon)}(x_i^{\mathrm{adv}} + a * \mathrm{sign}(\nabla x_i^{\mathrm{adv}})) \tag{4}$$

FGSM と PGD では，Adversarial Example の作成にモデルの勾配を利用するため，攻撃者が攻撃対象のモデルを持っている必要がある．そのため，モデルの情報を秘匿すれば攻撃が防げるように見えるが，それは難しい．なぜなら Adversarial Example にはあるモデルに対して作成した Adversarial Example が他のモデルに対しても Adversarial Example 成立する転移性があることが示されている [6]．そのため，モデルの情報を秘匿するだけでは防ぐことができない．攻撃者がモデルの情報を持っていることを White-Box 仮定，モデルの情報を持っていないことを Black-Box 仮定と呼称し，それぞれに対する研究が進められている．本稿では White-Box 仮定を前提とする．

### 2.2 ランダムノイズを用いた検知手法

既存手法であるランダムノイズを用いた検知 [11, 15] では，Adversarial Example と通常の入力データに対してランダムノイズを乗せた時の深層学習の出力変化が大きく違うという特性を利用して検知を行う．ランダムノイズを用いた手法は Roth ら [15] によって提案されたものと Huang ら [11] によって提案されたものが存在するが，これらは同時期に提案された手法であり，ほぼ同一のものである．本稿では，より性能の良い Roth らの手法を既存手法として扱い，評価実験等も Roth らの手法のみを対象とする．

ランダムノイズを用いた検知手法の検知の仕組みについて図 2(a) を用いて説明を行う．図 2(a) は class A と class B に属するデータが存在する空間を表現しており，深層学習モデルはこれらのデータを学習して，決定境界を引くことで分類を行う．このとき，攻撃者が class A のデータを class B に誤分類させるような攻撃を行なったとする．この攻撃は class A のデータに作為的なノイズを乗せることで，class B 方向に決定境界を超えるように画像を変換する．この操作により決定境界を超えたデータが Adversarial Example である．Adversarial Example の定義から，攻撃者は画像の見た目を class A から変更しないように最小の変換距離で誤分類を起こそうとする．その結果として，Adversarial Example は決定境界の近くの class A 側に突出している場所に生成されがちである．

こういった場所に存在する Adversarial Example に対してランダムノイズを加えると，図 2 に示すようにランダムな方向へ変換が起こる．このとき，突出した部分に存在する Adversarial Example は高確率で決定境界を超えるため出力の変化が起きやすく，一方で突出した部分に存在しない通常のデータは決定境界を超えないため出力の変化が起きにくい．既存手法ではこの違いを捉えることで Adversarial Example を検知できる．

しかしながら，既存手法にも検知が難しい Adversarial Example が存在すると考えられる．その例を図 2(b) に示す．

図 2(b) のような，大きな変換により，境界の突出した部分より内側に変換された場合や決定境界が凹んだ部分に変換された場合においてはランダムノイズによる決定境界を超えた出力変化が起きづらくなるため検知精度が落ちてしまう．攻撃側は決定境界の正確な構造はわからないので，偶発的に図 2(b) のような決定境界の内側や凹んだ位置に作成してしまうことは十分考えられる．また，攻撃者が既存手法の存在を知っている場合，変換距離を長くすることで意図的に作成することもできる．したがって，このような Adversarial Example のパターンを検知できないことは大きな問題である．実際に大きなノイズによって生成された Adversarial Example に対して検知性能が落ちることを 4.2 節の実験にて示す．

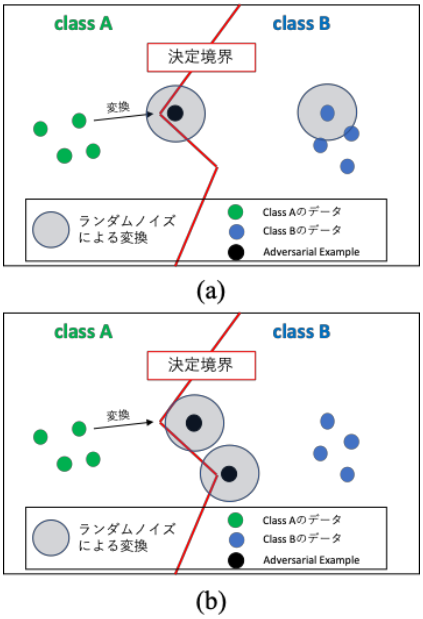



図 2　ランダムノイズを用いた Adversarial Detection

## 3 提 案 手 法

### 3.1 提案手法概要

ランダムノイズでは Adversarial Example が決定境界の突出した部分にない場合，境界を超えるような変換が起こりにくく，出力変化を捉えにくいという問題がある．その問題を解決するために，我々は多少ノイズが大きくなることで突出部分以外に生成されることはあっても，Clean Sample と比較すると決定境界に近い場所に Adversarial Example があるのではないかと仮定した．そして，その仮説にに基づきランダムノイズの代わりに決定境界の方向への変化を意図的に発生させることができる Adversarial noise を用いた検知を提案した．提案手法で用いた Adversarial noise を以下に示す．

$$\begin{aligned} &\xi_y = -\epsilon^{\mathrm{det}} * \mathrm{sign}(\nabla_x L(F(x), y)) \\ &s.t \quad y \neq y^{\mathrm{pred}} \end{aligned} \tag{5}$$

ここで，$\epsilon^{det}$ は検知時に加えるノイズの大きさ，$y^{\mathrm{pred}}$ はモデルによって出力された入力 $x$ の予測クラスである．上記の Adversarial noise は FGSM や PGD が Adversarial Example 生成時に用いるノイズと同種のものだが，本稿では Adversarial Example 生成時に用いるノイズと検知時に用いるノイズを区

別するために，検知時に用いるノイズを Adversarial noise $\xi_y$ と表記する．Adversarial noise を入力に加えることで $y$ クラスの決定境界を超える方向の変換を引き起こすことができる．

この提案手法の検知について図 3(a) を用いて説明する．図に示したように Adversarial noise $\xi_y$ によって引き起こされる真のクラス $y^{\text{true}}$ 方向の変換により，突出した部分の Adversarial Example は本来のクラス $y^{\text{true}}$ の決定境界に近いので，境界を超えるのに対して，class B に属する通常のデータは決定境界を超えづらいので，出力変化を捉えることができると考えられる．

また，提案手法は既存手法では精度が落ちる Adversarial Example も検知できる．これを図 3(b) に示す．図 3(b) から，提案手法は決定境界方向の変換を起こせるため決定境界の内側や凹んだ部分の Adversarial Example に対しても，決定境界に近ければ出力変化を捉えることができる．提案手法の元となる Adversarial Example が Clean Sample より決定境界に近い場所に存在するという仮説については 4.3 節にて検証を行う．

### 3.2 提案手法で用いる特徴量

本節では提案手法における検知時に用いる特徴量について概説する．提案手法は，Adversarial noise によって Adversarial Example にモデルの決定境界を超えるような出力変化を検知を行う．そのため，単純なものでは通常のデータと Adversarial Example に Adversarial noise を加えた時の出力変化の差分をとるなどが考えられるが，本稿においては，既存手法に用いられている特徴量を提案手法用に少し改変したものを用いた．本稿で用いた特徴量 AS を以下の式 6 に示す．

$$
\begin{aligned}
f_{y^{\text{pred}}}(x) &\equiv < w_{y^{\text{pred}}}, \phi(x) > \\
f_{y^{\text{pred}},z}(x) &\equiv f_z(x) - f_{y^{\text{pred}}}(x) = < w_z - w_y, \phi(x) > \\
g_{y^{\text{pred}},z}(x) &\equiv \frac{1}{K-1} \sum_{t \neq y^{\text{pred}}}^{K} (f_{y,z}(x+\xi_t) - f_{y,z}(x)) \\
\mu_{y^*,z} &\equiv E_{p(x^*|y^*)} E_\xi [g_{y^*,z}(x)] \\
\sigma^2_{y^*,z} &\equiv E_{p(x^*|y^*)} E_\xi [g_{y^*,z}(x) - \mu_{y^*,z}] \\
\bar{g}_{y^{\text{pred}},z}(x) &\equiv \frac{E_\xi [g_{y^{\text{pred}},z}(x) - \mu_{y^*,z}]}{\sigma_{y^*,z}} \\
\text{AS} &\equiv \max_z \bar{g}_{y^{\text{pred}},z}(x) \qquad (6)
\end{aligned}
$$

ここで，$y^{\text{pred}}$ は $x$ に対する予測クラスであり，$z$ は $y^{\text{pred}}$ 以外のクラスを示す．　$w_{y^{\text{pred}}}$ は深層学習モデルの最終層におけるクラス $y^{\text{pred}}$ を担当するユニットの重みである．また，$\phi(x)$ はモデルの最終層の一つ手前の出力を表す．<> は内積を表すので，$f_{y^{\text{pred}}}(x)$ はクラス $y^{\text{pred}}$ を担当するユニットの出力であるそして，$f_{y^{\text{pred}},z}(x)$ はクラス $y^{\text{pred}}$ とあるクラス $z$ の出力の差を見ている．$g_{y^{\text{pred}},z}(x)$ は，Adversarial noise を乗せた $f_{y^{\text{pred}},z}(x+\xi_t)$ と $f_{y^{\text{pred}},z}(x)$ の差であり，予測クラス以外のクラスの方向に変換を起こす Adversarial noise$\xi_t$ を用いた出力変化を表す．Adversarial noise$\xi_t$ は予測クラス $y^{\text{pred}}$ を除いた全てのクラス分だけ生成され，最終的には生成された $\xi_t$ によって引き起こされた出力変化の平均値が $g_{y^{\text{pred}},z}(x)$ となる．なお，ここで $K$ はクラスの総数である．また，$x^*$ は clean data であり，$y^*$ はその真のクラスである．$\mu_{y^*,z}$ と $\sigma^2_{y^*,z}$ は clean data の，$g_{y^{\text{pred}},z}(x^*)$ の平均と分散を表す．$\bar{g}_{y^{\text{pred}},z}(x)$ は $g_{y^{\text{pred}},z}(x)$ を $\mu_{y^*,z}$ と $\sigma^2_{y^*,z}$ にて正規化したものである．つまり，特徴量 AS は，入力データに対して Adversarial noise を加えた時の出力変化が通常のデータの時の出力変化がどれくらい離れているかどうかを見ている．

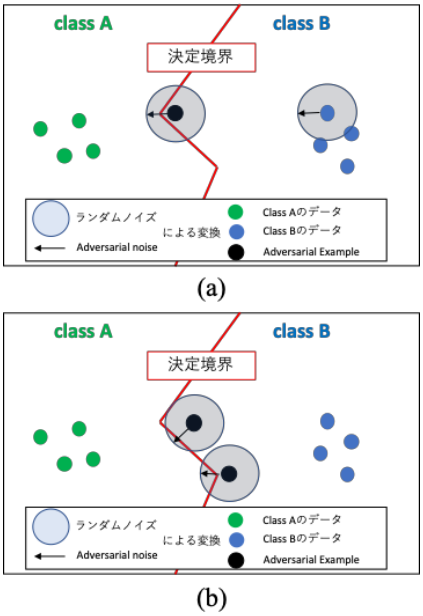


図 3　Adversarial noise による検知

## 4 評価実験

本節では，実験を通して，提案手法の性能と性質の評価を行う．4.1 節で実験の設定について述べ，4.2 で提案手法と既存手法の性能比較を行う．そして，4.3 にて特徴量の可視化などの実験により，提案手法の性質について解析を行う．

### 4.1 実験設定

**データセット:**本実験においてはデータセットとして，手書き数字画像のデータセット MNIST 及び 10 クラスの画像データセットである Cifar-10 を用いた．Cifar-10 は図 1 に示した Clean Sample である．それぞれのデータセットの内、5 万点の訓練データを分類モデルや提案手法および既存手法の検知器の学習に用い，そして 1 万点の評価データを用いて本実験の評価を行った．

**モデル:**モデルとして，MNIST では 5 層の CNN モデル，Cifar-10 では公開されている学習済みモデルを利用した [1]．

**攻撃手法:**検知するための Adversarial Example を生成するための攻撃として，先に述べた PGD を用いた．ノイズの大きさ $\epsilon$ は MNIST では 0.05〜0.3，Cifar-10 では 1/255〜32/255 の間でそれぞれ実験を行った．

**評価手法:**評価をする手法として，本稿で提案した提案手法と既存手法として Roth らによって提案されたランダムノイズを用いた検知手法を比較した．既存手法のパラメータは著者らが

1 : https://github.com/aaron-xichen/pytorch-playground

表 1　PGD に対する ROC-AUC(MNIST)

| | $\epsilon$ | | | |
|---|---|---|---|---|
| | 0.050 | 0.100 | 0.2 | 0.3 |
| 既存手法 | 0.999 | 0.999 | 0.999 | 0.999 |
| 提案手法 $\epsilon^{\mathrm{det}} = 1/255$ | 0.984 | 0.997 | 1.000 | 1.000 |
| 提案手法 $\epsilon^{\mathrm{det}} = 0.01$ | 0.830 | 0.992 | 1.000 | 1.000 |
| 提案手法 $\epsilon^{\mathrm{det}} = 0.02$ | 0.549 | 0.887 | 0.999 | 0.999 |

表 2　PGD に対する ROC-AUC(Cifar-10)

| | $\epsilon$ | | | |
|---|---|---|---|---|
| | 1/255 | 8/255 | 16/255 | 32/255 |
| 既存手法 | 0.697 | 0.989 | 0.983 | 0.965 |
| 提案手法 $\epsilon^{\mathrm{det}} = 1/255$ | 0.504 | 0.949 | 0.933 | 0.839 |
| 提案手法 $\epsilon^{\mathrm{det}} = 8/255$ | 0.519 | 0.993 | 0.999 | 1.000 |
| 提案手法 $\epsilon^{\mathrm{det}} = 16/255$ | 0.511 | 0.993 | 1.000 | 1.000 |
| 提案手法 $\epsilon^{\mathrm{det}} = 32/255$ | 0.506 | 0.992 | 1.000 | 1.000 |

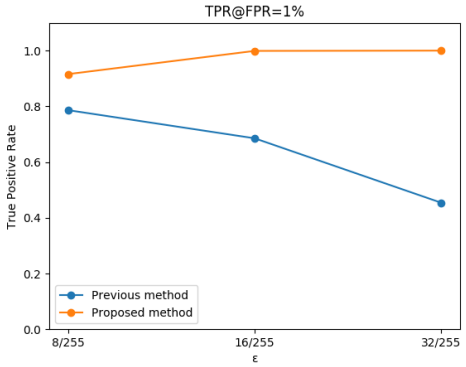



図 4　過検知率 1%の時の性能比較 (Cifar-10)

公開しているソースコードのものを用いた．

### 4.2　性 能 評 価

#### 4.2.1　ROC 曲線を用いた性能評価

各手法の ROC 曲線の AUC を表 1 と表 2 に示す．MNIST の実験結果である表 1 から，MNIST による実験では Adversarial Example 生成時のノイズの大きさ $\epsilon$ が小さい時は既存手法が優勢であり，それ以外の場合では既存手法と提案手法のうち最も性能の良いものがほぼ互角という結果になった．Cifar-10 の実験結果では MNIST と同様に $\epsilon$ の大きさが非常に小さい場合は既存手法の性能が高いが，$\epsilon$ が大きくなるほど性能が低下する傾向が見られた．一方で，提案手法は $\epsilon$ が 1/255 の時は既存手法に負けているが，$\epsilon$ が大きくなっても性能が低下せず，既存手法より高い性能を示した．

#### 4.2.2　ノイズの大きさに対する性能

先に述べたように提案手法は決定境界の突出した部分より内側に存在する Adversarial Example を検知することを目的とする．そこで，Adversarial Example 生成時のノイズの大きさに対する既存手法との性能比較を性能比較をを図 4 に示す．

グラフの縦軸は過検知 1 % に抑えた時の Adversarial Example の検知率を表す．グラフの横軸は検知対象の Adversarial Example 生成時に許容するノイズの大きさ $\epsilon$ をである．ノイズが大きいほど境界を大きく超える変換が起きやすくなるので，突出した部分ではなく決定境界の内側に存在する Adversarial Example が増えると考えられる．つまり，横軸の $\epsilon$ が大きいほど，既存手法で検知が難しいような Adversarial Example が生成されやすくなる．図 4 から $\epsilon$ が大きいほど既存手法は予想通り精度が 50 % ほどまで低下しているのに対して，提案手法は性能の低下が起こらず，高い精度を保っていることがわかる．このことから，提案手法は既存手法が苦手とする Adversarial Example に対しても検知することが可能である．

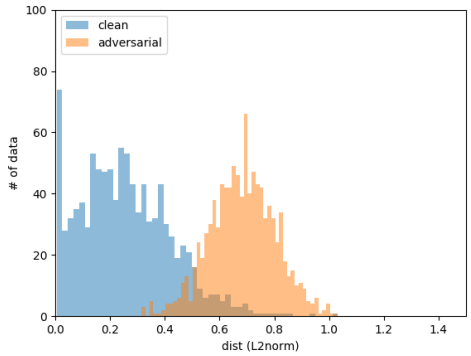



図 5　決定境界までの距離

### 4.3　提案手法についての解析

4.2 節における実験の結果，Adversarial Example に対して大きな $\epsilon$ のノイズが乗っている場合は，提案手法の方がよく検知できることがわかった．しかしながら，3 節で述べた Adversarial Example が Clean Sample より決定境界に近いため Adversarial noise での出力変化が起こりやすいという予測が正しければ $\epsilon$ が小さい場合でも既存手法と同程度には検知可能なはずである．したがって，この疑問を解決するために本節では提案手法について実験を通して解析する．また，以降の実験では Cifar-10 のみを対象とする．

#### 4.3.1　決定境界までの距離

まず，提案手法における Clean Sample より，Adversarial Example の方が決定境界に近いという予測を確認する．しかしながら，深層学習モデルの決定境界の形状や位置は不明である．そこで本実験では，決定境界とデータが近いならば，より少ないノイズで Adversarial Example が生成できると考え，PGD による各クラスへの Target attack を非常に小さなステップサイズを用いて行った時に，決定境界を超えた時点でのノイズの L2 ノルムを距離として扱う．実験結果を図 5 に示す．図 5 は各データの最も近い決定境界への距離のヒストグラムである．図から，最初の仮定とは異なり Adversarial Example が寧ろ Clean Sample より，決定境界から遠い位置にあることがわかった．

#### 4.3.2　特徴量の解析

仮定が間違っていた一方で，提案手法はノイズの大きさ $\epsilon$ が大きい時，既存手法より高い精度で検知を行なっている場合がある．その原因を示すために，提案手法の

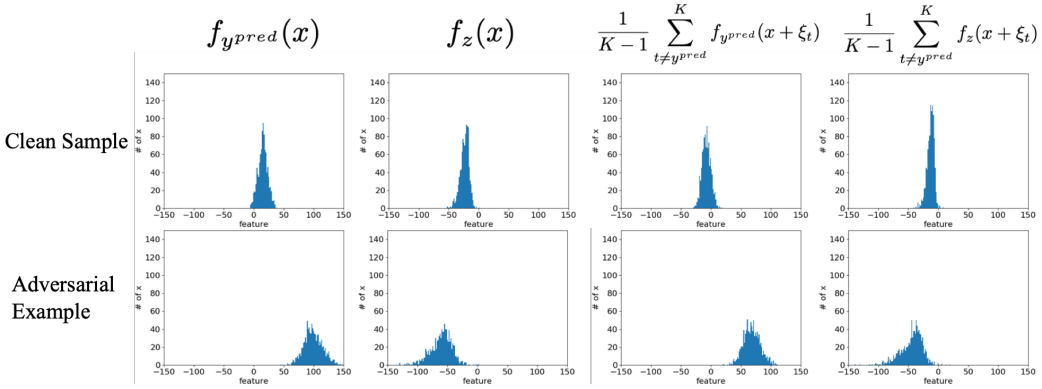



図 6　各特徴量の可視化 (Cifar-10,$\epsilon^{det} = 1/255$)

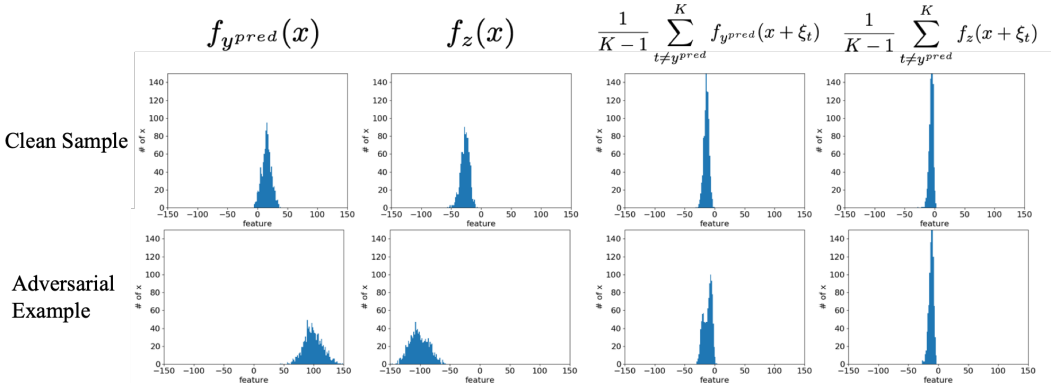



図 7　各特徴量の可視化 (Cifar-10,$\epsilon^{det} = 8/255$)

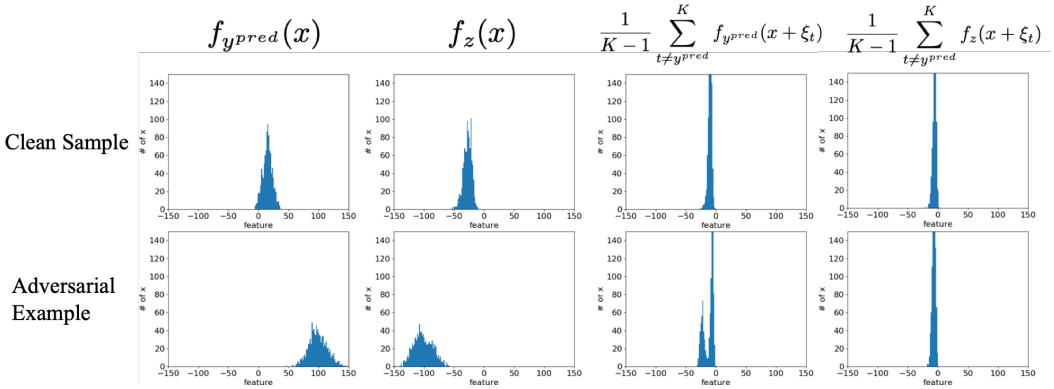



図 8　各特徴量の可視化 (Cifar-10,$\epsilon^{det} = 16/255$)

特徴量を可視化する．可視化した特徴量は式 6 で示した $g_{y^{\text{pred}},z}(x) = \frac{1}{K-1}\sum_{t\neq y^{\text{pred}}}^{K}(f_{y^{\text{pred}},z}(x+\xi_t)) - f_{y^{\text{pred}},z}(x)$ を分解した $\frac{1}{K-1}\sum_{t\neq y^{\text{pred}}}^{K} f_y(x+\xi_t) + \frac{1}{K-1}\sum_{t\neq y^{\text{pred}}}^{K} f_z(x+\xi_t) + f_y(x) + f_z(x)$ の各項を可視化する．その結果を各 $\epsilon^{\text{det}}$ ごとに図 6〜9 に示す．

図 7 から図 9，$\epsilon^{\text{det}}$ が 8/255 以上の場合，Cifar-10 の検知に関しては Clean Sample と Adversarial Example では $f_{y^{\text{pred}}}(x)$ と $f_z(x)$ のみに違いが存在することがわかった．このことから，$\epsilon^{\text{det}}$ が 8/255 以上の場合に提案手法は実質的にはモデルの出力の違いのみを見て Adversarial Example かどうかを検知している可能性がある．Adversarial Example を生成するときの $\epsilon$ を小さくした場合，出力が Clean Sample と同程度の Adversarial Example が生成され，そういった Adversarial Example を検知できないと考えられるので問題となる．これは表 2 において $\epsilon$ が小さい時に性能が出ないことからも見て取れる．

一方で，図 7 から $\epsilon^{\text{det}}$ が 1/255 の時には各項で違いが出ている．また，Clean Sample に Adversarial noise を乗せた時よりも，Adversarial Example に Adversarial noise を乗せた時のほうが値の変化が大きい．このことから，$\epsilon^{\text{det}}$ が 1/255 の時には Adversarial noise を乗せた時の出力変化を捕らえていると考えられる．しかしながら，$\epsilon^{\text{det}}$ が 1/255 は $\epsilon^{\text{det}}$ が 8/255 以上の場合ほどの性能が出ていない．

これらの解析から，提案手法は実験上 $\epsilon$ が大きい時に既存手法を上回る性能を持つが，その性能には疑問の余地が存在するため，更なる性能解析が必要である．このことについては今後の課題とする予定である．

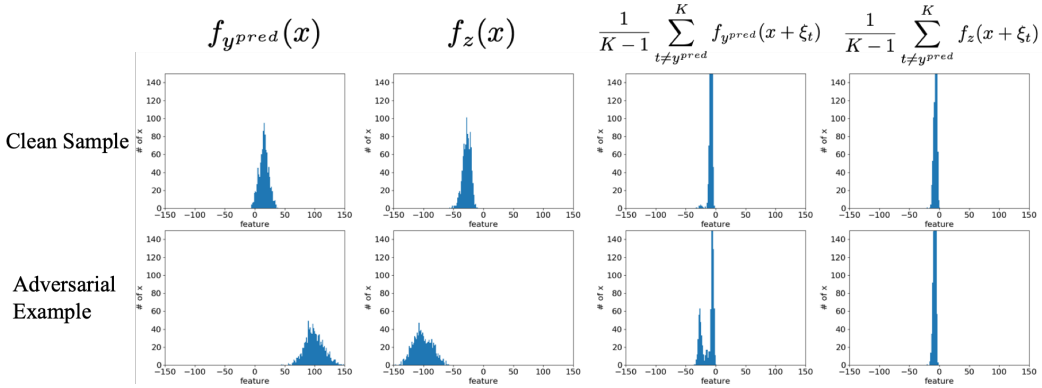



図 9　各特徴量の可視化 (Cifar-10,$\epsilon^{det} = 32/255$)

## 5　関 連 研 究

### 5.1　主な Adversarial Detection

先に述べたように Adversarial Example を検知することで，モデルへ入力されることを防ぐ Adversarial Detection は深層学習のモデルの性能低下を招かないことや，Adversarial Training では対処できない高次元のデータに対しても高い精度で検知ができるため，様々な手法が提案されている．

Grosse ら [8] は，Adversarial Example が通常のデータと違う確率分布に従うと仮定し，統計的検定により検知する手法を提案した．また，Metzen ら [9] は，分類モデルの各中間層の出力ごとに Adversarial Example か通常のデータかを識別するようなバイナリ検知器を訓練することで検知を行った．Feinman ら [10] は入力データの中間層の出力が，通常のデータ集団とどれくらい距離にあるのか，カーネル密度推定を用いて検知する手法を提案した．Xu らは，入力画像に平滑化フィルタやピクセルの値域を狭くなどの操作である Feature squeezing を行った時の出力が、Feature squeezing を行わない時の出力とどれくらい離れているかを評価して検知を行った．通常の画像は Feature squeezing によって出力は変化しないが，一方で Adversarial Example は微小のノイズを画像全体に乗せている影響か出力が著しく変化するため検知ができる．

ランダムノイズを用いた手法 [11,15] は先に述べた手法と比較して，非常に高精度に Adversarial Example を検知できる．本稿で既存手法とした Roth ら [15] による手法に加えて，Huang ら [11] による手法も存在するが，これらの手法は同時期に提案されたものでほぼ同じものである．しかしながら，2 で示したように Adversarial Example に乗せるノイズが大きくなると，決定境界の突出した部分より内側に Adversarial Example が生成されるようになるため，性能が低下する．

また，ランダムノイズを用いる手法は，Hu ら [12] によっても提案されている．この手法はランダムノイズに対して出力が変化しないような正則化項を加えて生成された Adversarial Example に対して，検出する方法を提案したものである．この手法は，ランダムノイズを用いた検知手法に対するピンポイントに対策した攻撃に対するさらなる防御手法であり，本稿とは趣旨が異なる．

### 5.2　Detection 以外の防御手法

Adversarial Example に対する対策手法は Adversarial Detection 以外にも様々なものが提案されている．本節では，その中でも代表的なものに絞って述べる．まず，1 節でも述べた Adversarial Training [7] は訓練データに Adversarial Example を混ぜて学習することで，頑強なモデルを生成する手法であり，最も代表的な防御方法である．長所は攻撃者の情報に寄らず安定した性能をだせることである．一方で短所は計算コストの大きさ，ロバスト性能と分類制度のトレードオフ [16]，学習データに入っていないデータに対してロバスト性能が著しく落ちる [17] ことである．Gradient Masking [20, 21] は微分不可能な計算を深層学習モデルに組み込むことで勾配を利用した Adversarial Example の作成を防ぐ手法である．しかしながら，Adversarial Example はあるモデルに対して作成された Adversarial Example が他のモデルでも Adversarial Example として成立するという特性があるので，その特性を攻撃者が利用することで突破できることが判明している [22]．また，正常なデータの特徴を学習することで，Adversarial Example が入力された際に，類似の正常データを出力する Input Reconstruction [23, 24] という手法も存在するが，先に述べた防御手法と比較すると性能が出ていない．

## 6　ま　と　め

本稿では既存手法であるランダムノイズを用いた検知手法が決定境界のより内側に存在する Adversarial Example に対しては検知精度が低下する問題を解決するために決定境界方向の変換を意図的に起こす Adversarial noise を用いた検知手法を提案した．提案手法は既存手法が苦手とする境界の突出部分に存在しない Adversarial Example を検知することに成功した．

しかしながら，$\epsilon$ が小さい Adversarial Example に対して性能が出ないことや，性能解析の結果，出力の差のみ見ている可能性があったりと性能には疑問の余地が大いに残った結果となった．今後の課題として，様々なデータセットを用いて解析を行い，提案手法の有用性について検討していく．

## 文　献


[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[2] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdelrahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Brian Kingsbury, et al. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal processing magazine*, 29, 2012.

[3] Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, et al. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 187–196. International World Wide Web Conferences Steering Committee, 2018.

[4] Omid E David and Nathan S Netanyahu. Deepsign: Deep learning for automatic malware signature generation and classification. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2015.

[5] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[6] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[7] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[8] Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick McDaniel. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280*, 2017.

[9] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. *arXiv preprint arXiv:1702.04267*, 2017.

[10] Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017.

[11] Bo Huang, Yi Wang, and Wei Wang. Model-agnostic adversarial detection by random perturbations. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 4689–4696. AAAI Press, 2019.

[12] Shengyuan Hu, Tao Yu, Chuan Guo, Wei-Lun Chao, and Kilian Q Weinberger. A new defense against adversarial images: Turning a weakness into a strength. In *Advances in Neural Information Processing Systems*, pages 1633–1644, 2019.

[13] Jiayang Liu, Weiming Zhang, Yiwei Zhang, Dongdong Hou, Yujia Liu, Hongyue Zha, and Nenghai Yu. Detection based defense against adversarial examples from the steganalysis point of view. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4825–4834, 2019.

[14] Yuxian Qiu, Jingwen Leng, Cong Guo, Quan Chen, Chao Li, Minyi Guo, and Yuhao Zhu. Adversarial defense through network profiling based path extraction. *arXiv preprint arXiv:1904.08089*, 2019.

[15] Kevin Roth, Yannic Kilcher, and Thomas Hofmann. The odds are odd: A statistical test for detecting adversarial examples. *arXiv preprint arXiv:1902.04818*, 2019.

[16] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

[17] Huan Zhang, Hongge Chen, Zhao Song, Duane Boning, Inderjit S Dhillon, and Cho-Jui Hsieh. The limitations of adversarial training and the blind-spot attack. *arXiv preprint arXiv:1901.04684*, 2019.

[18] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017.

[19] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[20] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2017.

[21] Guneet S Dhillon, Kamyar Azizzadenesheli, Jeremy Lipton, Zachary C anfd Bernstein, Jean Kossaifi, Aran Khanna, and Anima Anandkumar. Stochastic activation pruning for robust adversarial defense. *arXiv preprint arXiv:1803.01442*, 2018.

[22] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

[23] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*, 2018.

[24] Uiwon Hwang, Jaewoo Park, Hyemi Jang, Sungroh Yoon, and Nam Ik Cho. Puvae: A variational autoencoder to purify adversarial examples. *IEEE Access*, 7:126582–126593, 2019.