Vol.2019-SE-202 No.1
2019/7/12

モデルの歪みと機械学習プログラムの欠陥

中島 震[1,a)]

概要：機械学習ソフトウェアは，データの集まり（データセット）から有用な情報を導き出す方法に基づく．学習対象データセットは標本であり，標本の品質を客観的に考えることが難しい．多様なデータセットを用いるデータセット多様性が，ソフトウェア・テスティングによる品質検査の標準的な方法となる．本稿は，学習プログラムの欠陥がモデルの歪みとして現れるという仮説に基づき，データセット多様性を訓練済み学習モデルの歪みと関連して整理する．また，テスティングで用いるデータセットの自動生成法を提案し，手書き数字認識学習タスクの実験を通して提案方法の効果を考察する．

キーワード：ニューラル・ネットワーク，データセット多様性，ニューロン・カバレッジ，敵対学習

Distortion in Learning Models and Faults in Machine Learning Programs

Shin Nakajima[1,a)]

Abstract: Machine learning software derives useful information from a group of data, a dataset. However, a dataset is just a sample, and defining its absolute quality is almost impossible. Software testing with various datasets is a standard approach to the quality assurance. This paper introduces a hypothesis that faults in learning programs manifest themselves as distortion in the trained machine learning models. The paper also proposes a new dataset generation method, and illustrates its effectiveness with a case study of classifying hand-written roman numbers.

Keywords: Neural Networks, Dataset Diversity, Neuron Coverage, Adversarial Learning

1. はじめに

深層ニューラル・ネットワーク（DNN）に代表される機械学習の技術 [6][7] が発展し，極めて高度な信頼性（Extreme Reliability）達成が必須のアプリケーション・サービスにまで，その技術を利用する時代が近づいてきた．一方で，自動運転車に搭載される物体認識の DNN ソフトウェアに欠陥があり，それによる不具合が人身事故の原因となったかもしれない，ということが指摘される [23] など，品質問題への関心が大きくなっている．

機械学習の方法は，データの集まり（データセット）から帰納的に情報を導き出すという特徴がある．しかし，データセットは具体的なデータの集まりであり，標本選択バイアスの問題 [8] がつきまとう．また，データセットシフト [17] や敵対データ例 [18][19] といった機械学習が持つ特徴から，学習対象データセットに着目した絶対的な品質を考えることは難しい [15]．

機械学習ソフトウェアの場合，データセットの品質を間接的に確認するソフトウェア・テスティングの方法 [1]，特に，メタモルフィック・テスティング [3] が有力である [4]．さらに，テスト入力となるデータセットの自動生成法が技術的な鍵となる．従来の研究 [11][14][21][22][23] は，検査対象の特徴に着目する方法であるが，逆に一般性に欠ける．

本論文では，学習プログラムの欠陥が，モデルの歪みとして現れるという仮説に基づき，データセット多様性 [12][14] の考え方を訓練済み学習モデルの歪みと関連付けして整理する．また，さまざまな学習モデルに適用可能な一般的な

1 国立情報学研究所
NII, Chiyoda-Ku, Tokyo 101—8430, Japan
a) nkjm@nii.ac.jp

ⓒ 2019 Information Processing Society of Japan

自動生成方法を提案し，実験によって，提案方法の有効性を確認する．

2. 機械学習の方法

ニューラル・ネットワークによる教師あり学習を念頭に，機械学習ソフトウェアの概要を説明する．

多次元データ・ベクタ $\vec{x}^n$ と正解タグ・ベクタ $\vec{t}^n$ の組の集まりをデータセット DS とする．$DS = \{\langle \vec{x}^n, \vec{t}^n \rangle\}$ である．DS の経験分布を ρ^{ds} とすると，$d = \langle \vec{x}, \vec{t} \rangle$ として，$DS = \{\ d \mid d \sim_{i.i.d.} \rho^{ds}\ \}$ である[*1]．本稿では，$DS \sim_{i.i.d.} \rho^{ds}$ と略記する．

多次元データ $\vec{x}$ を入力とする非線形関数の集まりを $\{\vec{y}(W;\vec{x})\}_W$ と表す．この関数族のインデックス W は学習パラメータで，その値が確定すると入力データに対する計算結果（推論結果）を返す関数 $\vec{y}$ が決まる．この関数族自身，あるいは，その要素であるパラメータ値が未確定の $\vec{y}(W;\vec{x})$ を学習モデルと呼ぶ．

学習とは，学習モデルを決めて，訓練データセット LS=$\{\langle \vec{x}^n, \vec{t}^n \rangle\}$ を与えた時の誤差関数 $\mathcal{E}$ を最小化する W^* を求める数値最適化のことである．

$$W^* = arg\,\underset{W}{min}\ \mathcal{E}(W; \{\langle \vec{x}^n, \vec{t}^n \rangle\})$$

この最適化問題を解く学習プログラムを $\mathcal{F}_L(LS)$ と書く．今，$\ell(\vec{y}(W;\vec{x}^n), \vec{t}^n)$ を，$\vec{x}^n$ に対する出力値 $\vec{y}(W;\vec{x}^n)$ と正解タグ $\vec{t}^n$ の距離とする時，誤差関数 $\mathcal{E}$ を次のように定義する．

$$\mathcal{E}(W; \{\langle \vec{x}^n, \vec{t}^n \rangle\}) = \frac{1}{N}\sum_{n=1}^{N} \ell(\ \vec{y}(W;\ \vec{x}^n),\ \vec{t}^n\ )$$

誤差関数 $\mathcal{E}$ の最小化は LS の経験分布 ρ_{em} に関して，ℓ の期待値を最小化する W^* を求めることであり，

$$W^* \ = \ arg\ \underset{W}{min}\ E_{\langle \vec{x}, \vec{t} \rangle \sim \rho_{em}} [\![\ \ell(\vec{y}(W;\vec{x}), \vec{t})\ ]\!]$$

と表せる．

この解 W^* を用いて，運用時の入力データ $\vec{x}$ を受け付ける推論プログラムを $\mathcal{F}_I(\vec{x}) = y(W^*;\ \vec{x})$ と表す．

機械学習ソフトウェアでは，W^* を介して係わり合う 2 つのプログラム $\mathcal{F}_L$ と $\mathcal{F}_I$ について，品質の問題を考えることになる．なお，より一般的には，機械学習ソフトウェアの品質問題は 3 つの観点から整理される [13][15]．本稿では，学習プログラム $\mathcal{F}_L$ のテスト法を考察する．

3. 学習プログラムのテスト法

3.1 ソフトウェア・テスティング

ソフトウェア・テスティングでは，何を入力とするか，どのようにして実行結果が正しいかを判定するか，という 2 つの観点を考える [1]．前者をテスト入力生成問題，後者をオラクル問題と呼ぶ．メタモルフィック・テスティング（Metamorphic Testing, MT）[3][4] は，テスト不可能プログラム [20] を含む幅広い検査対象に対して，この 2 つの問題を解決する方法論を与える．以下，テスト入力生成の問題に関連する事項を説明する．

第 1 に，検査対象が想定した機能振舞いを示すことを正常系テスティング（Positive Testing）によって確認する．本稿では，プログラムが，事前・事後条件を伴う場合を考える．プログラム本体 $Body$ と事前条件 Pre，事後条件 $Post$ の間には，「事前条件を満たす時，プログラム本体実行後の状態で事後条件を満たす」という関係が成り立つ．これを，$Pre \Rightarrow [\ Body\ ](Post)$ と表記する．Pre を満たすテスト入力データに対して，この関係式が成り立つ時，プログラム本体 $Body$ はテストに合格したとする．

第 2 に，想定外の状態での機能振舞いが不具合を生じないことを異常系テスティング（Negative Testing）によって確認する．プログラムは Pre を満たさない入力に対しても，それなりの振舞いを示すことが期待される．そこで $\neg Pre$ を満たすデータをテスト入力として検査すれば良いだろう．たとえば，ファズ・テスティング（Fuzz Testing）の方法を用いる．

一方，欠陥がありそうなプログラム箇所を実行経路に含むような入力をテスト・データとして選べばテストの作業効率が良い，このような状況のテストを，コーナーケース・テスティング（Corner-case Testing）という．一般にプログラムの制御フローへの欠陥混入が多い．プログラム中の条件分岐に着目することで，分岐条件を集めて，経路制約条件（Path Constraints）PC を求める．この PC を満たす入力データ a を与えることで，意図した実行経路のテストを実施することができる．$PC(a) \Rightarrow Pre(a)$ であれば正常系テスティングであるし，異常系テスティングを意図して $PC'(a) \Rightarrow \neg Pre(a)$ となるような PC' を PC から構成すればよい．この PC によるテスト生成法は，検査対象の内部構造を用いる「ホワイトボックス」手法である．

3.2 学習プログラムへのテスト入力

次に，学習プログラム $\mathcal{F}_L$ をテストする場合について，上に述べた 3 つの観点から，問題を考える．

数値計算の関数 $\mathcal{F}_L$ をは，訓練データセット LS を入力し，訓練済み学習パラメータ値 W^* を出力する．この時，正常系テスティング，異常系テスティング，コーナーケース・テスティング，各々の場合，どのような条件を満たすテスト入力を用いれば良いのだろうか．

文献 [8] の題名を逆読みすると，訓練データセット LS が仕様を表すとなる．LS の経験分布を ρ_{em} としよう．一方で，LS はテスト対象プログラム $\mathcal{F}_L$ への入力データでもある．今，仮に，LS を標本とする母確率分布 ρ が既知としよう．つまり，ρ_{em} の良い近似としての ρ が存在するこ

[*1] 独立同一分布（independent and identically distributed）．

とであり，LS は $LS \sim_{i.i.d.} \rho$ を満たす具体的な入力値である．

$\mathcal{F}_L$ への入力 LS が「仕様」という点に注目する．$\mathcal{F}_L$ は機能仕様（LS）から具体的な機能振舞いを確定する情報（W^*）への変換関数とみなせる．この時，ρ は入力データのドメインを定義する．つまり，ρ が $\mathcal{F}_L$ の入力仕様に関わる情報を規定し，LS は具体的な入力データである．そこで，$LS \sim_{i.i.d.} \rho$ が事前条件を満たすことを表すとすれば良い．標本選択バイアスがある場合，独立同一分布ではない．$LS \sim_{\neg i.i.d.} \rho$ と書くことにする．この「仕様の誤り（Specification Error）」[8] は事前条件を満たさない場合に相当する．

確率分布 ρ を入力データのドメインとするように対応付けると，正常系テスティングは $LS_P \sim_{i.i.d.} \rho$，異常系テスティングは $LS_N \sim_{\neg i.i.d.} \rho$ になる．一方，コーナーケース・テスティングについての PC による方法は検査対象の内部構造を用いた．$\mathcal{F}_L$ の場合，制御フローのようなプログラムの構造的な情報からコーナーケースを定義することはできない [11]．検査対象の学習タスクによって，コーナーケースの考え方が異なる [12]．学習タスクごとに，言い換えると ρ ごとに，何がコーナーケースとなるかを考察する必要がある．

また，LS_N は，母確率分布 ρ と関連することに注意して欲しい．通常のファズ・テスティングのように，ランダムに LS_N を選ぶとすると $\neg(LS_N \sim \rho)$ であり，全く異なる機能仕様あるいは学習タスクを扱うことになる．この点が，ランダム入力を考慮する通常のファズと異なる．さらに，ρ は既知ではないことから，具体的な議論を進めることが難しい．次節で具体的な議論の基礎となる考え方を導入する．

3.3 データセット多様性からのテスト

データセット多様性 [12] を定義しよう．$K+J$ 個の異なるデータセットの集まり $\{DS_i\}$ を，互いに素な K 個の $\{DS_k\}$ と J 個の $\{DS_j\}$ に分ける（$\{DS_i\} = \{DS_k\} \cup \{DS_j\}$，$\{DS_k\} \cap \{DS_j\} = \emptyset$）．ただし，$(\cup_{i=1}^{J+K} \rho_i) \sim_{i.i.d.} \rho$ として，$\rho_j \sim_{i.i.d.} \rho$ で，$\rho_k \sim_{\neg i.i.d.} \rho$ とする．つまり，DS_k は標本選択バイアスのことがある．また，DS_i の経験分布を ρ_i とすると，$i1 \neq i2$ に対して，$\neg(DS_{i1} \sim_{i.i.d.} \rho_{i2})$ とする．この時，$K+J$ 個のデータセットの集まり $\{DS_i\}$ は，データセット多様性を示すという．

今，データセット DS（$DS \sim_{i.i.d.} \rho$）が与えられたとする．$DS^{(0)} = DS$ として，$DS^{(0)}$ から異なるデータセットを生成していく方法を考えたい．ある変換関数 T を用いて，$DS^{(K)} = T(DS^{(K-1)})$ である．T の選び方によって，$DS^{(K)} \sim_{i.i.d.} \rho$ あるいは $DS^{(K)} \sim_{\neg i.i.d.} \rho$ とすれば，データセット多様性を示す集まり $\{DS^{(K)}\}$ を得ることができる．

学習プログラム $\mathcal{F}_L$ が訓練データセット LS と共に与えられた時，上記の手順で作成した $\{LS^{(K)}\}$ の要素となるデータセットをテスト入力として用いて検査すれば，ある $K1$ と $K2$ について，$LS^{(K1)} \sim_{i.i.d.} \rho$ ならびに $LS^{(K2)} \sim_{\neg i.i.d.} \rho$ を得ることができ，正常系テスティングと異常系テスティングを実施することができるだろう．

4. 歪みと欠陥

4.1 モデルの歪み

説明の都合上，いくつかの概念と記号を導入しよう．今，分類学習タスクを考え，データセット DS を $\{\langle \vec{x}^n, t^n \rangle\}$ とする．また，分類カテゴリのインデックスを c とすると，正解タグ t^n は c を値とする．

訓練済み学習パラメータ値の集まり W_e に対して，softmax 関数が $\vec{y}(W_e; \vec{x})$ の出力を規定する時，最も大きな softmax 値をとる分類カテゴリを求める関数 $topOne(_;_)$ は，$topOne(W_e; \vec{x}) = arg\ \underset{c}{max}\ \vec{y}(W_e; \vec{x})$ である．また，$\sharp(P; X)$ を有限集合 X の要素 x に対して述語 $P(x)$ が論理真となる割合とする時，DS の W_e に対する正解率 $acc(_,_)$ を $acc(W_e, DS) = \sharp(topOne(W_e; \vec{x}^n) \supseteq \{t^n\}; DS)$ と定義する．以下で添字 $j = 1\ or\ 2$ とする．

2 つのデータセットに対して，DS_1 を基準とする DS_2 の歪み度合いの関係を，$DS_1 \preceq DS_2$ と表す．また，同一の学習モデル $\vec{y}(W; _)$ に関する訓練済み学習パラメータの集まり 2 つに対して，W_1^* を基準とする W_2^* の歪み度合いの関係を $W_1^* \preceq W_2^*$ と表す．

この歪み度合いの関係を実数値をとる観測関数 obs を用いて定義する．微小値 ϵ_{obs} が与えられた時，$|\ _\ |$ が絶対値を表すとして，関係

$$\mathcal{R}_{obs}(W_1^*, W_2^*) \triangleq (|obs(W_1^*) - obs(W_2^*)| \leq \epsilon_{obs})$$

によって，$(W_1^* \preceq W_2^*) \Leftrightarrow \neg\mathcal{R}_{obs}(W_1^*, W_2^*)$ である．このような観測関数をみつけて歪み度合いを定義するといってよい．

4.2 仮説

次に 3 つの仮説を導入する．学習方式・アルゴリズムを上位仕様とする時，学習プログラム $\mathcal{F}_L$ の開発過程で混入する欠陥と歪み度合いの関連性を調べるたい．

「仮説 1」「良い」学習方式に基づくプログラムを考える．$W^* = \mathcal{F}_L(LS)$ の時，欠陥の有無に関わらず，$acc(W^*, LS) \approx acc(W^*, TS)$ である．

ここで，$LS \sim_{i.i.d.} \rho_{em}$ であって，LS と異なる TS は，$TS \sim_{i.i.d.} \rho_{em}$ である．また，繰り返し過程で得られるエポック e での $W^{(e)}$ に対する正解率グラフも「同等」になる．$acc(W^{(e)}, LS) \approx acc(W^{(e)}, TS)$．なお，汎化性能ならびに収束性が良い適切な学習方式を「良い」とした．

「仮説 2」 2 つの学習結果を $W_j^* = \mathcal{F}_L(LS_j)$ とする時，$LS_1 \preceq LS_2 \Rightarrow W_1^* \preceq W_2^*$ である．

つまり，歪んだ LS_j から歪んだ W_j^* が導かれる．

「仮説 3」適切な obs を選ぶと，$LS_1 \preceq LS_2$ の時，欠陥のない $\mathcal{F}_L$ は $W_1^* \approx W_2^*$，欠陥のある $\mathcal{F}'_L$ は $W_1^{*\prime} \prec W_2^{*\prime}$ である．

ここで，$W_1^* \approx W_2^* \Leftrightarrow \mathcal{R}_{obs}(W_1^*, W_2^*)$ である．

4.3 歪みデータ生成

「歪みデータ」の簡単な作成例として，敵対データ例生成の L-BFGS [18] を利用する方法を示す．

元データ $\vec{x}_S$ をラベル $\vec{t}_T$ に誤分類させる敵対データ例は制約条件付きの最適化問題の解 $\vec{x}^*$ として求めることができる．訓練データセット $LS = \{\langle \vec{x}^n, \vec{t}^n \rangle\}$ から得られた W^*（つまり $W^* = \mathcal{F}_L(LS)$）に対して，

$$A_\lambda(W^*; \vec{x}_S, \vec{t}_T, \vec{x}) = \ \ell(\vec{y}(W^*;\ \vec{x}),\ \vec{t}_T) + \lambda \cdot \ell(\vec{x},\ \vec{x}_S)$$

に対して，最適化問題，

$$\vec{x}^* = arg\ \underset{\vec{x}}{min}\ A_\lambda(W^*; \vec{x}_S, \vec{t}_T, \vec{x}), \qquad \text{s.t. } |x_j| \leq 1$$

によって求める．多次元データ $\vec{x}$ を $\langle \ldots, x_j, \ldots \rangle$ とした．上の関数 A_λ の定義式中，第 1 項は $\vec{x}$ の W^* に対する推論結果が与えられた教師タグ $\vec{t}_T$ を再現すること，第 2 項は $\vec{x}$ と元データ $\vec{x}_S$ との違いが小さいことを表す．

次に，上記の方法を利用して次のような変換関数を考える．入力データセットの要素である元データ $\vec{x}^n$ 全てにノイズを挿入したデータセットを求めるものである．

$$\vec{x}^{(n)*} = arg\ \underset{\vec{x}}{min}\ A_\lambda(W^*; \vec{x}^n, \vec{t}^n, \vec{x})$$

として，

$$T_A(LS, W^*) = \{\ \langle \vec{x}^{(n)*},\ \vec{t}^n \rangle\ \}$$

は，歪んだ入力ベクトル $\vec{x}^{(n)*}$ からなるデータセットとなる．λ が微小値の場合，歪みの度合いが大きい．λ が適切な大きさの時，$\vec{x}^{(n)*}$ はノイズの影響が小さく，$\vec{x}^{(n)}$ とほぼ一致する．

$T(LS^{(K)}) = T_A(LS^{(K)}, \mathcal{F}_L(LS^{(K)}))$ と定義すると，$LS^{(K+1)} = T(LS^{(K)})$ と表せる．あるいは，λ 値の選び方によって，さまざまなデータセット LS_λ を作ることもできる．以下，このようにして構成したデータセット（$LS^{(K)}$ と略記）が持つ性質をいくつか示す．

「性質 1」 $LS^{(K)}$ は LS（$LS^{(0)}$）と同じ学習タスクを指し示す．

$W^{(0)*} = \mathcal{F}_L(LS^{(0)})$ であり，$LS^{(1)}$ の要素である最適化問題の解 $\vec{x}^{(n)*}$ は $\vec{x}^{(n)}$ から大きく外れない．特別な場合に x^n になり得る．したがって，$LS^{(1)}$ は $LS^{(0)}$ と同じ学習タスクを指し示す．同様にして，$LS^{(K)}$ は $LS^{(K-1)}$ と同じ学習タスクを指し示す．その結果，$LS^{(K)}$ は $LS^{(0)}$ と同じ学習タスクを指し示す．

「性質 2」 $LS^{(K-1)} \preceq LS^{(K)}$

$LS^{(K)}$ の構成法から $LS^{(K-1)}$ を基準とした歪み度合いの関係が成り立つ．

「性質 3」 $LS^{(K)}$ は $W^{(K-1)*}$ に過適合する．

最適化問題の形より，適切に λ の値を決めることで，$LS^{(K)}$ は $LS^{(K-1)}$ よりも $W^{(K-1)*}$ に過適合することができる．

「性質 4」 ある K_c があって，$K \geq K_c$ を満たす K に対して，$acc(W^{(K)*}, TS) \approx acc(W^{(K_c)*}, TS)$ で，$acc(W^{(0)*}, TS) \geq acc(W^{(K_c)*}, TS)$ である．

ある K_c に対して，「性質 3」から $K = K_c + 1$ に対して，$LS^{(K)}$ は $W^{(K_c)*}$ に過適合する．$W^{(K)*} = \mathcal{F}_L(LS^{(K)})$ に対して，$W^{(K)*}$ は $LS^{(K)}$ から得られる解なので，$acc(W^{(K)*}, LS^{(K)}) \approx acc(W^{(K_c)*}, LS^{(K)})$ である．したがって，$acc(W^{(K)*}, TS) \approx acc(W^{(K_c)*}, TS)$ である．また，$LS^{(0)} \sim \rho_{em}$ かつ $TS \sim \rho_{em}$ と $LS^{(0)} \preceq LS^{(K_c)}$ から $acc(W^{(0)*}, TS) \geq acc(W^{(K_c)*}, TS)$ である．

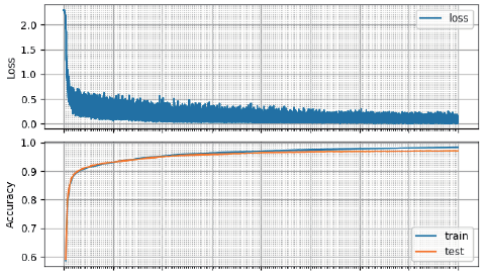
「性質 5」 繰り返しの結果，$LS^{(\infty)}$ と $W^{(\infty)*}$ に収束する．

「性質 4」の説明から，K_c より大きな K（$K > K_c$）に対して $acc(W^{(K)*}, LS^{(K)}) \approx acc(W^{(K_c)*}, LS^{(K)})$ なので，$acc(W^{(K)*}, LS^{(K)}) \approx acc(W^{(K+1)*}, LS^{(K+1)})$ として良い．「性質 3」から $LS^{(K+1)}$ は $W^{(K)*}$ に過適合するので，$acc(W^{(K)*}, LS^{(K)}) \approx acc(W^{(K)*}, LS^{(K+1)})$ であり，$LS^{(K)} \approx LS^{(K+1)}$ として良い．$LS^{(K)} \approx LS^{(\infty)}$ となる代表元 $LS^{(\infty)}$ を選ぶことができる．その結果，$W^{(\infty)*} = \mathcal{F}_L(LS^{(\infty)})$ に収束する．

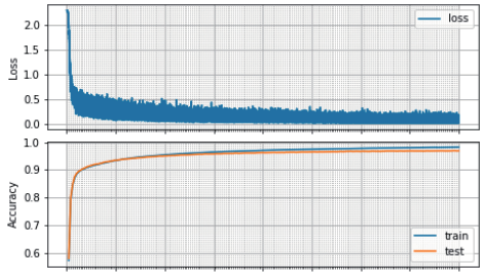
5. 実験

5.1 実験手順

手書きローマ数字分類学習の標準ベンチマーク問題である MNIST データセットを用いた実験の手順を示す．0 から 9 の手書き数字を 10 分類する問題である．訓練データセット LS が 60,000，試験データセット TS が 10,000 からなり，LS ならびに TS を 70,000 データのプールからランダムに選ぶと考える．したがって，LS と TS は同じ経験分

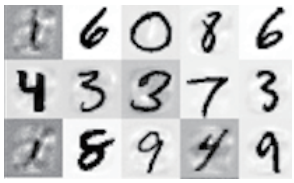


(a) Probably Correct (ProgPC)

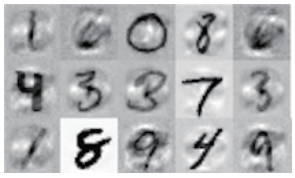


(b) Bug-Injected (ProgBI)

図 1 Loss and Accuracy: MNIST dataset

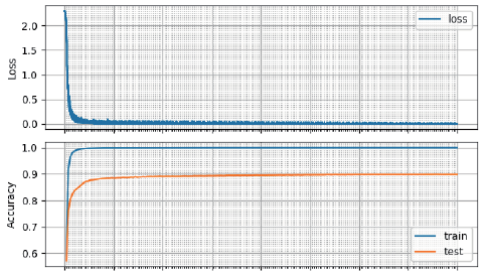


(a) Probably Correct (ProgPC)

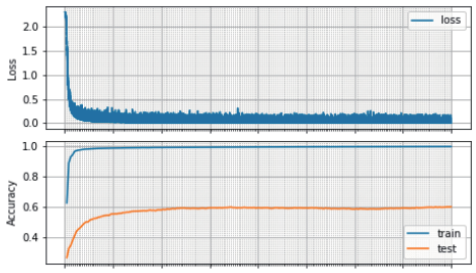


(b) Bug-Injected (ProgBI)

図 2 Distorted Data



(a) Probably Correct (ProgPC)



(b) Bug-Injected (ProgBI)

図 3 Loss and Accuracy: Follow-up Dataset

布を持つ．これらの MNIST データセットを $LS^{(0)}$ および $TS^{(0)}$ とする．

学習モデルとして古典的なニューラル・ネットワークを選択し，中間層と出力層の活性化関数を，各々，ReLU および softmax とした．ReLU は負の入力に対して定数 0 を，正の入力に対して線形応答する．softmax は分類結果の信頼度を与える連続関数である．推論対象データセットを DS とする時，関数 $topOne(W_e, DS)$ によって W_e に対する正解率を求めることができる．

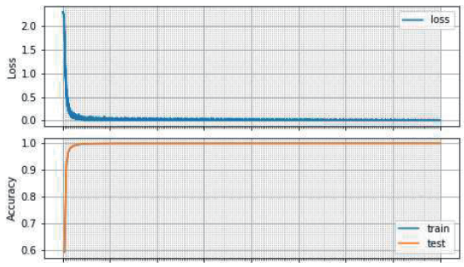
学習プログラムとして，欠陥がないと信じる $\mathcal{F}_L^{PC}$ (ProgPC) と，これに欠陥を挿入した $\mathcal{F}_L^{BI}$ (ProgBI) の 2 つを準備し，以下の実験を，各々の系列で実施して結果を比較した．なお，$MD = PC\ or\ BI$ として，$\mathcal{F}_L^{MD}$ 等と略記する．

(a) MNIST データセットの学習　$\mathcal{F}_L^{MD}(LS^{(0)})$ の解探索の過程で，繰り返しエポック e での誤差率 $\mathcal{E}$ と $LS^{(0)}$ ならびに $TS^{(0)}$ に対する正解率 $acc(W^{(e)}, LS^{(0)})$ および $acc(W^{(e)}, TS^{(0)})$ の変化を調べて $W^*(0)$ に至る収束傾向を監視する（図 1）．第 1 に，$LS^{(0)}$ と $TS^{(0)}$ が同じ経験分布にしたがうことから，$\mathcal{F}_L^{MD}$ の各々について，正解率が同等であることが確認できる．第 2 に，欠陥の有無によらずに，誤差率ならびに正解率グラフは，ほとんど区別できない．以上から，この実験結果は「仮説 1」と整合する．

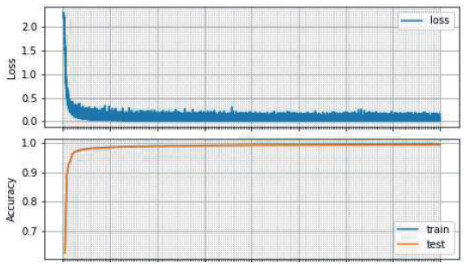
(b) 歪みデータセットの生成　ある λ を選んで，$T_A(LS^{(0)}, \mathcal{F}_L^{MD}(LS^{(0)}))$ によって，歪みデータセット $LS_{MD}^{(1)}$ を生成する．図 2 に，その一部を示した．目視する限り，両方とも $LS^{(0)}$ よりも「汚れて」おり，歪みの微小信号が加わっている．$LS^{(0)} \preceq LS_{MD}^{(1)}$ と考えてよさそうである．「性質 2」の具体例となっている．

(c) 歪みデータセットの学習　$\mathcal{F}_L^{MD}(LS_{MD}^{(1)})$ について，(a) と同様のグラフを監視した．ただし，正解率については，$acc(W_{MD}^{(e)}, LS_{MD}^{(1)})$ および $acc(W_{MD}^{(e)}, TS^{(0)})$ を調べた．図 3 に，その結果を示す．

ⓒ 2019 Information Processing Society of Japan



(a) Probably Correct (ProgPC)



(b) Bug-Injected (ProgBI)

図 4 Loss and Accuracy: Distorted Testing Dataset

ある程度，学習が進行したエポック以降，次のような 3 点を観察することができよう．第 1 に，$acc(W^{(e)}_{MD}, LS^{(1)}_{MD})\approx 1.0$ であり，$W^{(e)}_{MD}$ が $LS^{(1)}_{MD}$ に過適合している．「性質 3」と整合する．

第 2 に，$W^{*(1)}_{MD} = \mathcal{F}^{MD}_{L}(LS^{(1)}_{MD})$ に対して，$acc(W^{*(1)}_{MD}, TS^{(0)})$ は $acc(W^{*(0)}_{MD}, TS^{(0)})$ よりも低下している．したがって，$W^{*(0)}_{MD} \preceq W^{*(1)}_{MD}$ と考えられる．(b) より，$LS^{(0)} \preceq LS^{(1)}_{MD}$ だったことから，「仮説 2」と整合する．

第 3 に，観測関数 $obs(W^*)$ として正解率 $acc(W^*, TS^{(0)})$ を選んだと考えて，R_{acc} の ϵ_{obs} として 0.1 程度を選ぶと，$\neg R_{acc}(W^{*(1)}_{BI}, W^{*(0)}_{BI})$ で，$R_{acc}(W^{*(1)}_{PC}, W^{*(0)}_{PC})$ である．「仮説 3」と整合する．

(d) 歪みデータセットの正解率　(b) の方法で試験データセットを歪める．つまり，$TS^{(1)}_{MD} = T_A(TS^{(0)}, \mathcal{F}^{MD}_{L}(LS^{(0)}))$ とし，$\mathcal{F}^{MD}_{L}(LS^{(0)}_{MD})$ の過程で，$acc(W^{(e)}_{MD}, LS^{(1)}_{MD})$ および $acc(W^{(e)}_{MD}, TS^{(1)}_{MD})$ を監視した（図 4）．「仮説 1」ならびに「性質 3」と整合する．

5.2 不活性ニューロン

第 4.1 節では，訓練済み学習モデルの歪み度合いを天下り的に導入し，適当な観測関数の計算結果から歪み度合いの関係を定義した．以下，ニューロン・カバレッジ [16] と歪み度合いの関係を考察する．

一般に，個々のニューロンの入出力は，活性化関数を σ として，$out = \sigma(\sum w_j \times in_j)$ と表現できる．入力信号 in_j が確定した時，決められた閾値よりも大きな出力信号 out を持つニューロンを「活性」状態にあると云う．AN を活性状態ニューロンの集まり，TN を全ニューロンとして，ニューロン・カバレッジを $NC = |AN|/|TN|$ と定義する．本稿では，NC 計算の対象を中間層に限定する．また，ReLU を σ としたので，活性状態を定義する閾値を 0 とする．

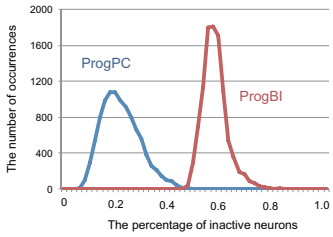


図 5 Inactivated Neurons

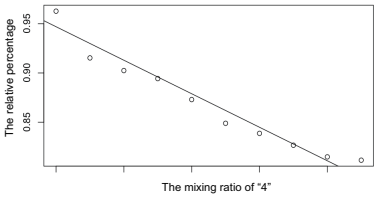
図 5 に試験データセット $TS^{(0)}$（10,000 個）を与えた時の不活性ニューロンの割合に関する測定結果を示した．図 1 の収束したエポックでのスナップショットに相当する．横軸は不活性ニューロン率 $(1 - NC)$ であり縦軸に対応する値を得た試験データの度数を表す頻度グラフである．図 5 から欠陥混入のない $\mathcal{F}^{PC}_{L}$ の学習結果 W^*_{PC} は 20%近傍にピークがある一方，欠陥混入した $\mathcal{F}^{BI}_{L}$ が導いたモデル（W^*_{BI}）は 60%近傍にシャープなピークを示すことがわかる．つまり，W^*_{BI} のほうが不活性ニューロンが多く，活性状態にあるニューロンの偏りが大きい．W^*_{BI} は W^*_{PC} よりも歪みの度合いが大きいと考えることができるだろう．

MNIST 訓練データセットを $LS^{(0)}$ とする時，ProgPC の実験で生成した歪みデータセットを $LS^{(1)}_{PC}$，ProgBI の実験で生成した歪みデータセットを $LS^{(1)}_{BI}$ とする．「性質 2」から，$LS^{(0)} \preceq LS^{(1)}_{PC}$ かつ $LS^{(0)} \preceq LS^{(1)}_{BI}$ となる．また，図 2 に示した画像の目視から，$LS^{(1)}_{PC} \preceq LS^{(1)}_{BI}$ と考えられる．この時，「仮説 2」から $W^{(1)*}_{PC} \preceq W^{(1)*}_{BI}$ と考えられる．つまり，図 5 はこの状況を示す．

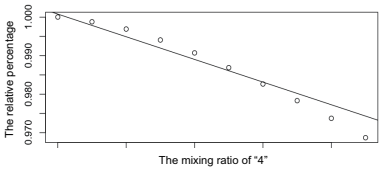
一方，図 3 は次のように理解できる．$TS^{(0)}$ の経験分布は MNIST 訓練データセット $LS^{(0)}$ と同じと考えられるが，歪み度合いの関係から $LS^{(1)}_{PC}$ や $LS^{(1)}_{BI}$ の経験分布とは異なる．$LS^{(1)}_{PC} \preceq LS^{(1)}_{BI}$ が成り立つ（図 2）とすると，「仮説 2」から $W^{(1)*}_{PC} \preceq W^{(1)*}_{BI}$ である．この歪み度合いが $TS^{(0)}$ に対する正解率の違い（図 3）として現れ，この違いは不活性ニューロン率の違い（図 5）と整合する．

不活性ニューロン率が大きいことは，推論結果に影響するニューロン数が少ないことであり，学習モデル全般を有効に使っているとは云い難い．図 5 は，不活性ニューロン率の頻度を示す．この時，2 つの可能性があるだろう．入力データによらず特定のニューロンが活性状態になる「活性の局所化」と入力データによって活性状態のニューロンが異なる「均一化」である．図 5 を得た際の測定によって，前者の「活性の局所化」つまり偏りが生じていることがわ

図 6 Mixing Two Numbers



(a) Inactive Neurons



(b) Softmax Values

図 7 Metrics for Mixed Numbers

かった．以上から，不活性ニューロン率と歪み度合いに相関があると考えることは直感的に妥当と云えよう．

上記の議論は，ひとつの実験結果からの推測である．不活性ニューロン率が歪み度合いの指標となるかもしれない．

5.3 歪み応用による画像の混合

第 4.3 節の歪みデータ生成方法を応用として，2 つの画像を混合した新しい画像を得る実験を行った．この実験では ProgPC だけを使って，$W_{PC}^{(0)*} = \mathcal{F}_L^{PC}(LS^{(0)})$ として求めた訓練済み学習モデルを用いた．

図 6 は，MNIST データセットの訓練データセット $LS^{(0)}$ 中の「6」と「4」の画層を各々 1 つ選び，混合比率を変化させて得た画像である．上段は，正解タグを 6 として，最左は「6」のみとし，左から右へ「6」の比率を低下させていった．最右は「4」のみからなり，したがって，推論結果が 6 となる敵対データ例である．下段は，同様にして，正解タグを 4 として，「4」から出発し「6」の比率を増加させた．

図 6 の 2 つの画像列は，第 4.3 節の歪みデータ生成方法によって，目視上，連続的に変化する歪みデータを作成できることを示している．

このようにして作成した混合画像列を $\mathcal{F}_I^{PC}(_)$ に入力し，softmax 値ならびに不活性ニューロン率を測定した．図 7 に示した測定結果は，横軸に図 6 の上段の画像列を配置する．図 7(a) は最左の画像を基準とする相対的な不活性ニューロンの比率を，図 7(b) は同様にして求めた相対的な softmax 値の比率を表すグラフである．画像列の左から右へ，正解タグの softmax 値の減少幅は両端で約 3%になる．混合データの作成法から良好な推論性能を示すことが期待され，これを確認することができた．また．不活性ニューロン率の減少幅は両端で約 15%であり，活性状態にあるニューロン数の増加は有意といえる．つまり，入力データの歪み度合いが大きくなると，活性状態になるニューロンが広がりをもち，非局所化する．

一般に，学習の汎化性能を向上させようとすると，学習モデルの自由度は大きいことが好ましい．一方，自由度が大きいと，想定範囲のデータに対しては活性状態ニューロンが局所化し，つまり，冗長なニューロンが増加すると考えられる．汎化性能が良いと，歪み度合いの大きいデータに対応できる．また，図 7(a) から，活性状態ニューロンが増加し，十分な自由度を持っていた．なお，混合したデータ列は敵対データ例を含むことから，汎化性能を目指すことと敵対データ例が存在することは表裏一体であることがわかる．

6. 関連研究

深層ニューラル・ネットワーク（DNN）のソフトウェア・テスティングでは，データセット多様性を示すテストケースの自動生成が鍵となる．

DeepRoad [22] は，に GAN を拡張した UNIT に基づく方法によって新たなデータを生成する．ここで，敵対生成ネットワーク（Generative Adversarial Networks, GAN）[19] は，DNN 表現による生成モデルを合成する方法であって，零和ゲーム*2 として定式化される．与えられたデータセットの経験分布 ρ_{em} にしたがう新しいデータを出力する [5]．UNIT[10] は GAN を拡張した方法で，2 つのデータセットに属する画像を合成した画像データからなるデータセットを生成する．元の試験データセットと同様な分布にしたがうことから，「想定範囲での網羅性向上」であり，正常系テスティングに使えるといえる

機械教師（Machine Teaching）[24] は，訓練済み学習モデル $\vec{y}(W^*; _)$ を導く「最適な」訓練データセットを求める問題の枠組みで，2 段階最適化問題として定式化される．「想定範囲外のデータセット」生成に関心があり，例外系テスティングへの応用が可能な枠組みといえる．

本稿の方法は，歪みデータを単目的最適化問題によって生成するもので，初期データセットの経験分布から外れるようなデータセット多様性を実現する．「想定から少し離れたデータセット」生成であり，正常系に対するコーナーケース・テスティングに使えるといえる．

機械学習の研究では，標本選択バイアス [8] などの理由によってデータセットの変化が生じた場合であっても，妥当な推論性能を示すロバストな学習方法の確立が大きな課題となっている [17]．また，訓練データセットが小さい時にバイアスを避けることを目的としたデータセット拡

*2 「れいわ」と読む，Zero-Sum のこと．

表 1 Dataset Generation Methods

	GAN [5]	UNIT [10]	Machine Teaching [24]	本研究
問題定式化	零和ゲーム	min-max ゲーム	2 段階最適化問題	単目的最適化問題（[18]）
$\sim_{i.i.d.}\ \rho_{em}$ のデータ生成	◎	◎	○	△
$\sim_{\neg i.i.d.}\ \rho_{em}$ のデータ生成	-	-	○	○
データの混合	-	データセット間	-	データセット内

張（Data Augmentation, DA）の研究（e.g. [9]）を含む．DeepRoad[22] は GAN を DA の方法として用いた．

一方，本稿が対象とする品質保証では，データセット多様性の考え方に基づき，偏りのあるデータセットを生成する方法が関心となる．使い方によっては，データセット拡張にも利用可能かもしれない．表 1 は，上に紹介した方法の比較を整理したものである．

7. おわりに

深層ニューラル・ネットワーク（DNN）に代表される機械学習ソフトウェアの検査では，欠陥の検出効率の良い多様なデータセットを用いる必要がある．本論文では，学習プログラムの欠陥が，モデルの歪みとして現れるという仮説に基づき，データセット多様性の考え方を訓練済み学習モデルの歪みと関連付けして整理し，データセットの自動生成方法を提案した．また，学習モデルの歪みとニューロン・カバレッジの関連性を実験によって考察した．

今後，CNN や RNN といった DNN 法に，本稿で提案したデータセット自動生成法を適用すること，また，学習モデルの歪みの厳密な定義法を考察すること，などの研究を進めていきたい．

謝辞 本研究の一部は JSPS 科研費 JP18H03224 の助成および NEDO の委託業務[*3]によって実施した．実験に協力して下さった今井克則氏（(株) グラッツ）に感謝する．

参考文献

[1] P. Ammann and J. Offutt : *Introduction to Software Testing*, Cambridge University Press 2008.
[2] L. Breiman : Statistical Modeling: The Two Cultures, *Statistical Science*, 16(3), pp.199-231, 2001.
[3] T.Y. Chen, S.C. Chung, and S.M. Yiu : Metamorphic Testing - A New Approach for Generating Next Test Cases, HKUST-CS98-01, The Hong Kong University of Science and Technology, 1998.
[4] T.Y. Chen, F.-C. Kuo, H. Liu, P.-L. Poon, D. Towey, T.H. Tse, and Z.Q. Zhou : Metamorphic Testing: A Review of Challenges and Opportunities, *ACM Computing Surveys* 51(1), Article No.4, pp. 1-27, 2018.
[5] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, anY. d Bengio : Generative Adversarial Nets, In *Adv. NIPS 2014*, pp.2672-2680, 2014.
[6] I. Goodfellow, Y. Bengio, and A. Courville : *Deep Learning*, The MIT Press 2016.
[7] S. Haykin : *Neural Networks and Learning Machines (3ed.)*, Pearson India 2016.
[8] J.J. Heckman: Selection Bias as a Specification Error, *Econometrica*, vol.47, no.1, pp.153-161, 1979.
[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton: Imagenet classification with deep convolutional neural networks. In *Adv. NIPS 2012*, pp. 1097—1105, 2012.
[10] M.Y. Liu, T. Breuel, and J. Kautz: Unsupervised Image-to-image Translation Networks, In *Adv. NIPS*, pp.700-708, 2017.
[11] S. Nakajima and H.N. Bui : Dataset Coverage for Testing Machine Learning Computer Programs, In *Proc. 23rd APSEC*, pp.297-304, 2016.
[12] 中島震：データセット多様性のソフトウェア・テスティング，コンピュータ・ソフトウェア，35(2), pp.26-32, 2018.
[13] S. Nakajima : Quality Assurance of Machine Learning Software, In *Proc. GCCE 2018*, pp.601-604, 2018.
[14] S. Nakajima : Dataset Diversity for Metamorphic Testing of Machine Learning Software, In *Proc. 8th SOFL+MSVL*, pp.21-38, 2019.
[15] 中島震，妹尾義樹，大岩寛，磯部祥尚：機械学習ソフトウェアの品質評価保証レベル，電子情報通信学会ソフトウェアサイエンス研究会，那覇，2019.
[16] K. Pei, Y. Cao, J. Yang, and S. Jana : DeepXplore: Automated Whitebox Testing of Deep Learning Systems, In *Proc. 26th SOSP*, pp.1-18, 2017.
[17] J. Quinonero-Candela, M. Sugiyama, A. Schwaighofer, and N.D. Lawrence (eds.) : *Dataset Shift in Machine Learning*, The MIT Press 2009.
[18] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus: Intriguing properties of neural networks, In *Proc. ICLR 2018*, 2014.
[19] D. Warde-Farley and I. Goodfellow: Adversarial Perturbations of Deep Neural Networks, in *Perturbation, Optimization and Statistics*, The MIT Press 2016.
[20] E.J. Weyuker : On Testing Non-testable Programs, *Computer Journal*, 25 (4), pp.465-470, 1982.
[21] X. Xie, J.W.K. Ho, C. Murphy, G. Kaiser, B. Xu, and T.Y. Chen : Testing and Validating Machine Learning Classifiers by Metamorphic Testing, *J. Syst. Softw.*, 84(4), pp.544-558, 2011.
[22] M. Zhang, Y. Zhang, L. Zhang, C. Liu, and S. Khurshid: DeepRoad: GAN-Based Metamorphic Testing and Input Validation Framework for Autonomous Driving Systems, In *Proc. 33rd ASE*, pp.132-142, 2018.
[23] Z.Q. Zhou and L. Sun: Metamorphic Testing of Driverless Cars, *Comm. ACM*, vol.62, no.3, pp.61-67, 2019.
[24] X. Zhu : Machine Teaching: An Inverse Problem to Machine Learning and an Approach Toward Optimal Education, In *Proc. 29th AAAI*, pp.4083-4087, 2015.

*3 NII へは再委託

ⓒ 2019 Information Processing Society of Japan