Vol.2019-SE-201 No.13
2019/3/8

深層学習のフィーチャに基づく学習モデル設計方法の提案と評価

太田 龍之介[†1]　玉置 悠斗[†1]　高井 直哉[†1]　青山 幹雄[†1]

概要：深層学習において要求する認識精度を達成する学習モデルを安定して生成するためには学習過程が分析可能な開発方法が必要である．しかし、データが学習に与える影響の分析が困難なため発見的な開発になっている．本稿では深層学習のフィーチャ（特徴量）に基づく学習モデル設計方法を提案する．学習過程における予測誤差とフィーチャの関係を分析することで段階的な開発を行う．訓練とテストの仮説検証型の二重反復開発プロセスによるそれぞれの誤差曲線の特性を活かした分析により、各反復での追加データを制御することで深層学習モデルを生成する．プロトタイプを実装し実データに適用することで提案方法の有効性と妥当性を評価する．

キーワード：深層学習，学習モデル生成，フィーチャ設計，特徴量設計，二重反復プロセス

A Feature-Based Design Method of Learning Model for Deep Learning and its Evaluation

RYUNOSUKE OTA[†1]　YUTO TAMAKI[†1]　NAOYA TAKAI[†1]　MIKIO AOYAMA[†1]

1. はじめに

近年，深層学習を用いて大量のデータから汎用的なルールを獲得させるシステム開発が盛んに行われている．深層学習の応用例は多種多様であり，特に自動運転車のような安全性が最優先されるべき分野では，学習モデルの認識精度は重要な要素である．また，十分な認識精度を達成する学習モデルの生成には大量のデータが必要とされ，データ収集には膨大な人的コストがかかる．一方で，学習にも膨大な時間的コストがかかるため，学習モデルの安定した精度の確保が必要である[3]．そこで，要求する認識精度を達成する学習モデルを安定して生成するためには学習過程が分析可能な開発方法が必要である．しかし，従来の開発では訓練データの選択におけるランダム性が強いため，データが学習に与える影響の分析が困難になり，発見的な開発になっている．

本稿では，データのフィーチャ（特徴量）に着目した段階的な深層学習モデル設計方法を提案する．提案方法を実現するプロトタイプを実装し,実データへ適用する.また，開発プロセスを実行し，従来の学習方法と比較することで提案方法の有効性と妥当性を評価する．

2. 研究課題

本稿では上記を踏まえ，以下の 2 点を研究課題とする．

(1) RQ1: フィーチャに基づき段階的に学習可能な深層学習モデル開発プロセスの提案.
(2) RQ2: 実際の画像データに提案方法を適用し，有効性と妥当性を評価.

3. 関連研究

3.1 深層学習のモデル生成[2]

深層学習では，データから機械に汎用的なルールを学習させ，学習済みモデルを生成する．モデルはデータのフィーチャをもとに学習を行い，生成されたモデルを利用して推論処理を行う．しかし，モデルの設計方法は体系化されておらず，試行錯誤に依っている状態である．特にデータのフィーチャに基づいた開発方法は確立されていない．

3.2 フィーチャ設計

機械学習モデルの予測精度を改善するために，適用対象となる問題の本質を表現したフィーチャに基づきデータを設計する技術体系である．フィーチャ設計では，主に探索的データ分析，データクレンジング，フィーチャの生成，変換，選択などの工程がある．特に，フィーチャ選択では訓練データの中から有用なフィーチャを選び出すことで，データセットの品質を向上させる．

遺伝的データ，構造化データ，異種データ，時系列データなどの多様なデータにおけるフィーチャ選択アルゴリズムの研究[5]や，教師あり学習，半教師あり学習，教師なし学習それぞれにおけるフィーチャ選択方法の提案[6]，また，画像データのフィーチャに関する研究[1]や，強化学習を用いて効率的にフィーチャ設計を行う提案[4]がある．

3.3 VGG16[8]

ILSVRC2014 のクラス分類の部門で高評価を得た畳み込み 13 層，全結合 3 層，計 16 層から成るニューラルネットワークである.100 万枚の画像データを学習しており,1,000 クラスを分類する．

†1 南山大学 理工学部 ソフトウェア工学科
Dep. of Software Engineering, Nanzan University

4. アプローチ

深層学習モデルの予測精度は，データ収集，データ生成，モデリングなどの各工程の内容によって変化し，生成された学習済みモデルを評価するまで測定することができない．そのため，学習モデルの生成を繰り返し実施する必要があり，目標の精度を達成するためには多大な工数を必要とする．しかし，従来の開発プロセスでは，学習モデルの生成が段階的に行われていないため，精度が達成できなかった場合の原因の分析が困難となっている．

本稿では，小量のデータを順次追加しながら学習を行い，フィーチャと訓練誤差，汎化誤差の関係を分析することで段階的に開発を行うアプローチをとる．さらにソフトウェア工学におけるインクリメンタルプロセスの考え方を応用し，訓練誤差と汎化誤差の性質の違いを考慮することで，訓練とテストを二重ループで実行する学習モデルの二重反復開発プロセスを提案する(図 1).

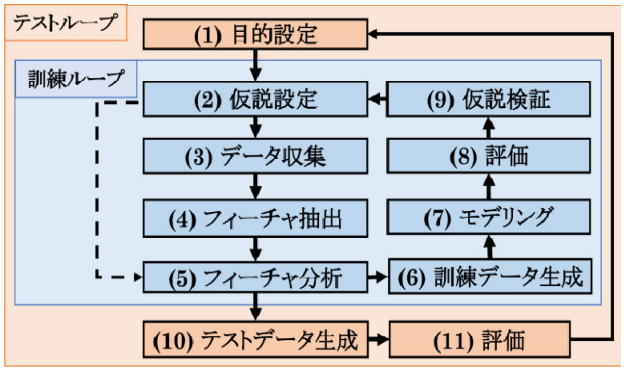


図 1 アプローチ
Figure 1 Approach

5. 提案方法

5.1 二重反復開発プロセス

本稿で提案する二重反復開発プロセスを図 2 に示す.

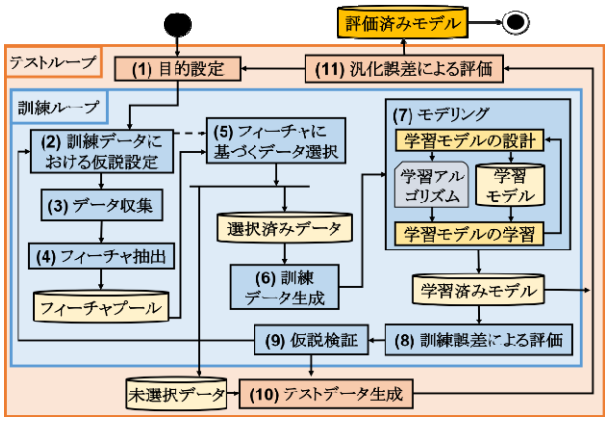


図 2 二重反復開発プロセス
Figure 2 Double Iterative Process

開発プロセスの詳細は以下の通りである.

(1) 目的設定

学習モデルにおいて，要求を満たすような目的を設定する．本稿では，解決する対象の問題と要求する学習モデルの認識精度を目的として設定する．プロセスを繰り返すにつれて目的設定は詳細化される.

(2) 訓練データにおける仮説設定

目的の達成に必要だと推定される訓練データについての仮説を設定する．本稿では，収集データの量や種類，ループごとの追加データ数，後に定義するSα，Fα(訓練誤差における評価指標の基準値)などの仮説を設定する．検証プロセスの結果に基づいて，必要があれば追加，変更する．また，後のデータ収集が不要な場合，フィーチャ分析プロセスに移る.

(3) データ収集

仮説設定で設定した仮説に基づき，分析対象のデータを取得する.

(4) フィーチャ抽出

データ収集で取得した全てのデータに対し，フィーチャを抽出する．取得したフィーチャはフィーチャプールに格納し，ループごとのフィーチャ分析の対象フィーチャとして扱う.

(5) フィーチャに基づくデータ選択

フィーチャプールに格納されたフィーチャから，後述のデータ選択プロセスに基づくデータ選択を行う．分析した結果，選択済みデータと未選択データを生成する.

(6) 訓練データ生成

生成された選択済みデータに対し，必要に応じてデータ整形，アノテーションを行い，訓練データを生成する.

(7) モデリング

学習モデルの設計と学習の 2 ステップから構成される．学習プロセスでは，生成した訓練データを学習し，結果として学習済みモデルが生成される.

(8) 訓練誤差による評価

生成された学習済みモデルに対し，訓練誤差による評価を行う．その結果，後で定義する S，F の値を算出する.

(9) 仮説検証

後述のデータ選択プロセスに基づき，仮説設定で設定した仮説が満たされているかどうか検証する．満たしていると判断した場合,テストループのテストデータ生成に移る.

(10) テストデータ生成

未選択データに対しアノテーションを行い，テストデータを生成する.

(11) 汎化誤差による評価

テストデータを用いて，学習済みモデルに対し汎化誤差による評価を行う．評価した結果，目的が達成されていればプロセス終了となる.

5.2 フィーチャに基づくデータ選択

5.2.1 データ選択プロセスの構成

本稿では，訓練誤差と汎化誤差をまとめて予測誤差と呼ぶことにする．段階的に分析しながら学習を行うために，データが持つフィーチャ数と学習済みモデルに対する予測誤差の関係に着目し，ループごとに追加する学習対象データを選択する．そこで，事前実験の結果からデータのフィーチャ数に応じて予測誤差の推移が変化すると仮定し，フィーチャ数をもとにデータをグループ化する．そして，フィーチャ数が最小のグループから学習を行うことで，学習プロセスを段階的に実行可能とする．さらに，訓練誤差が一定の基準値を満たすが，目標の精度に達していない場合はフィーチャ数が必要な数に達していないと判断し，訓練データ全体のフィーチャ数を増加するようにデータを追加する．これによって，その時点の学習に効果のあるフィーチャを特定し，学習を効率化する．ここで，フィーチャに基づいてデータを追加するプロセスをデータ選択プロセスと呼ぶことにする．

予測誤差とフィーチャ数の関係に関する仮説から，データ選択プロセスを提案する．データ選択プロセスは，フィーチャステージ作成と予測誤差分析プロセスの 2 ステップから構成される．データ選択プロセスの構成を図 3 に示す．

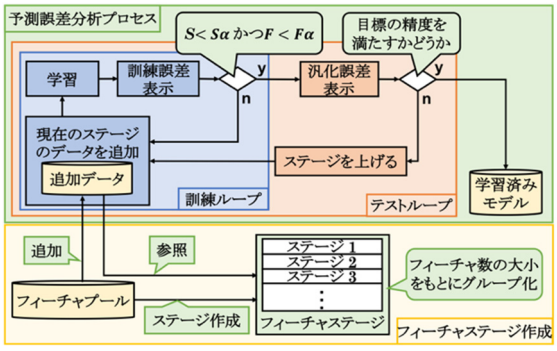


図 3 データ選択プロセス
Figure 3 Data Selection Process

それぞれのステップについての詳細を以下に示す．

(1) フィーチャステージ作成

予測誤差の値の推移を分析可能にするために，データが持つフィーチャ数の大小をもとにデータをグループ化する．フィーチャステージ作成では，以下の 3 つの手順を踏む．

1) フィーチャ数の算出

フィーチャプールに格納された各データのフィーチャから，各データにおけるフィーチャ数を獲得する．

2) ステージ設計

獲得したフィーチャ数から，フィーチャ数の大小をもとにグループ化するためのグループ数と各グループにおけるフィーチャ数の範囲を決定する．また，フィーチャ数の範囲は各グループで統一し，各グループに十分なデータ量を含むように設定する．

3) ステージ作成

2)をもとにデータをグループ化する．本稿では，フィーチャ数が最小のグループをステージ 1 とし，大きさの順に番号付ける．また，グループ化したフィーチャ全体をフィーチャステージと呼ぶ．

(2) 予測誤差分析プロセス

予測誤差を分析し，追加データを選択するプロセスを図 4 に示す．また，各プロセスの詳細を以下に示す．

1) 現在のステージのデータを追加

ループごとの追加データ数を設定し，フィーチャステージを参照する．次に現在設定されているステージのデータを訓練データとして追加する．また，最初のステージの設定はステージ 1 とする．

2) 訓練誤差を表示

現在の訓練データを学習させ，訓練誤差を表示する．そして，訓練誤差から $\boldsymbol{S}$，$\boldsymbol{F}$ を算出し，設定した基準値$\boldsymbol{S\alpha}$，$\boldsymbol{F\alpha}$と比較する．その結果，$\boldsymbol{S}$，$\boldsymbol{F}$ が条件を満たしていれば，次のプロセスに移り，満たさなければ 1)に戻る．

3) 汎化誤差を表示

テストデータを用いて，汎化誤差を表示する．また，学習モデルが目的設定で設定した目標の精度を満たしているか判断し，満たしていなければ 4)に移る．目標の精度が達成されればプロセス終了となる．

4) ステージを上げる

フィーチャステージに対する参照するステージを現在のステージから 1 段階上げる．

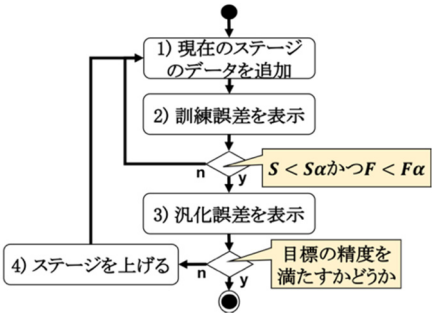


図 4 予測誤差分析プロセス
Figure 4 Prediction Error Analysis Process

5.2.2 予測誤差の評価指標

予測誤差はデータに対するエラー率を表すため，予測誤差の値の推移は，学習の収束や安定性を示唆する．そのため，フィーチャ分析プロセスにおいて，現在の学習状況から，次に追加して学習させるべきデータを決定するために，予測誤差の推移を分析する．そこで，予測誤差の推移を分析するための評価指標として，$\boldsymbol{S}$，$\boldsymbol{F}$ を定義する．

$$\boldsymbol{S} = \int_{0}^{n} \{\boldsymbol{f}(\boldsymbol{k}) - \boldsymbol{t}(\boldsymbol{k})\} \tag{1}$$

$$\boldsymbol{F} = \sum_{\boldsymbol{k}=1, \boldsymbol{a_k} > \boldsymbol{a_{k-1}}}^{\boldsymbol{n}} (\boldsymbol{a_k} - \boldsymbol{a_{k-1}}) \tag{2}$$

式 1，式 2 において，n は最終エポック数，$f(k)$は予測誤差の値の推移の近似曲線を表す関数，$t(k)$は期待する予測誤差の値の推移を表す関数，a_kは k エポックでの予測誤差の値である．また，S，F の値が十分に小さいかどうかを判断するための基準値として$S\alpha$，$F\alpha$を定義する．

(1) 式 1: 予測誤差は，学習が安定して収束するほど，0 へ早く収束する傾向にある．そこで，予測誤差の収束の速さを定量的に評価するために S を定義する．S は実際の予測誤差と期待する予測誤差との差を表し，予測誤差の推移が理想の曲線に近づくほど，値が小さくなる．

(2) 式 2: 予測誤差は，学習モデルのデータに対するロバスト性が高いほど，値の変化が小さくなる傾向にある．そこで，予測誤差の変化を定量的に評価するために F を定義する．F は予測誤差の総変化幅を表し，各エポックにおける a_k の変化が小さいほど，値が小さくなる．

5.2.3 予測誤差とフィーチャ数の関係

事前実験として，フィーチャ数ごとの予測誤差の推移を確認した．その結果から，学習モデルを評価した結果として出力される予測誤差の値の推移は，モデルが学習したデータのフィーチャ数に影響を受けるとの仮説を立てた．また，訓練誤差と汎化誤差では影響の受け方が異なると考える．仮定する訓練誤差，汎化誤差とフィーチャ数，及びデータ数との関係を以下に示す．

(1) 訓練誤差

定義した評価指標で訓練誤差を評価した結果とフィーチャ数，及びデータ数との関係を表 1 のように仮定する．S の値は学習データ全体のフィーチャ数が小さく，データ数が多いほど小さくなる．一方，F の値はフィーチャ数が小さいほど小さくなるが，データ数には影響を受けないと考える．

表 1 訓練誤差とフィーチャの関係
Table 1 Relationship between Training Errors and Features

訓練誤差	フィーチャ数	データ数
S	$\propto$	$\propto^{-1}$
F	$\propto$	

(2) 汎化誤差

定義した評価指標で汎化誤差を評価した結果とフィーチャ数，及びデータ数との関係を表 2 のように仮定する．S の値はフィーチャ数，データ数が大きいほど小さくなり，F はどちらにも影響されないと考える．

表 2 汎化誤差とフィーチャの関係
Table 2 Relationship between Generalization Errors and Features

汎化誤差	フィーチャ数	データ数
S	$\propto^{-1}$	$\propto^{-1}$
F		

6. プロトタイプの実装と実行

提案方法を支援し，実データ適用して有効性と妥当性を評価するために，プロトタイプを実装した．

6.1 実装環境

プロトタイプに使用したソフトウェアコンポーネントを表 3 に，ハードウェアコンポーネントを表 4 に示す．

表 3 ソフトウェアコンポーネント
Table 3 Software Components

コンポーネント	コンポーネント名	バージョン
OS	Ubunts	16.04
実装言語	Python	3.6.4
深層学習フレームワーク	Chainer	4.2.0
評価結果の可視化	ChainerUI	0.2.0
データ加工ツール	Pandas	0.20.3
可視化ツール	Matplotlib	2.0.2

表 4 ハードウェアコンポーネント
Table 4 Hardware Components

メモリ	CPU	GPU	コア数
64GB	i-Core i7	NVIDIA, GTX1080Ti	3,584

6.2 プロトタイプのアーキテクチャ

本研究のプロトタイプのアーキテクチャを図 5 に示す．

(1) データ分析プロセス

取得したデータのフィーチャを抽出し，フィーチャ，データ名，フィーチャ数をカラムとするテーブルデータをフィーチャプールに格納した．また，データの加工，分析には Pandas を用いた．

(2) 訓練プロセス

訓練データの生成，モデルの学習処理を Python と Chainer[7]で実装する．また，訓練誤差による評価結果の可視化を ChainerUI によって行う．

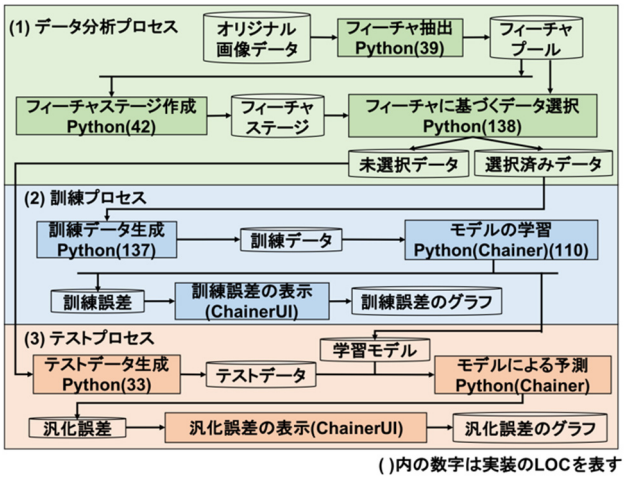


図 5 プロトタイプのアーキテクチャ
Figure 5 Architecture of Prototype

(3) テストプロセス

テストデータの生成，モデルによる予測処理を Python と Chainer で実装する．また，汎化誤差による評価結果の可視

化を ChainerUI によって行う．

6.3 データ分析プロセスの実行

図 5 に示すデータ分析プロセスに従って次のように実行される．

(1) フィーチャ抽出

プロトタイプでのフィーチャ抽出を図 6 に示す．

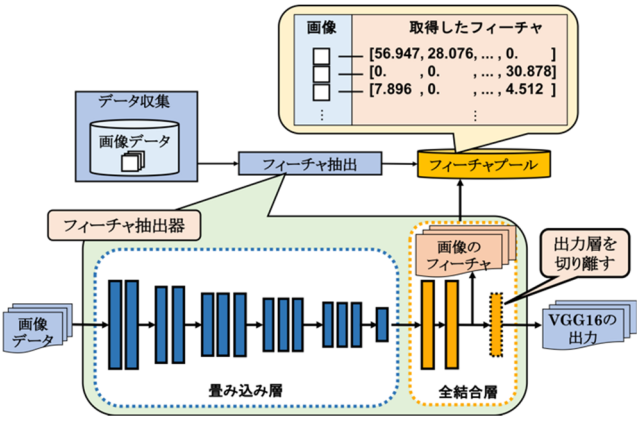


図 6 フィーチャ抽出
Figure 6 Feature Extraction

本稿では，適用する対象データを画像データとするため，学習済みモデルである VGG16 の出力層を切り離すことでフィーチャ抽出器を生成し，全ての画像データに対しフィーチャを抽出する．フィーチャプールには画像データとその画像のフィーチャを 1 対 1 で格納し，フィーチャから画像を参照可能にする．

(2) フィーチャステージ作成

プロトタイプでのフィーチャステージ作成を図 7 に示す．また，作成における手順を以下のように行う．

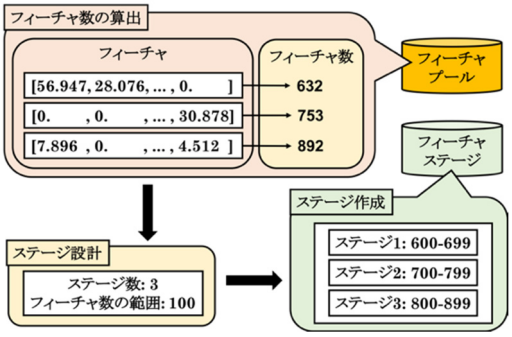


図 7 フィーチャステージ作成
Figure 7 Feature Stage Creation

1) フィーチャ数の算出

1 つのデータに対し，抽出したフィーチャは要素数 4,096 の配列で表される．配列の要素のうち，0 より大きい要素の個数をフィーチャ数として算出し，フィーチャプールに格納されている全データに対し行う．

2) ステージ設計とステージ作成

算出したフィーチャ数から，フィーチャステージの設計を行う．今回，収集した画像データのフィーチャ数が 600 から 899 となるデータが大半を占めた．そこで，各ステージが十分なデータ量となるように，ステージ数を 3，フィーチャ数の範囲を 100 とした．そして，ステージ設計をもとにフィーチャステージを作成する．

(3) フィーチャに基づくデータ選択

予測誤差分析プロセスに従って，現在のステージが遷移していく．現在のステージがステージ 1 の場合のデータ選択を図 8 に示す．まず，フィーチャステージを参照しステージ 1 のフィーチャ数の範囲を取得する．次に，フィーチャプールに格納されているデータのうち，フィーチャ数がステージ 1 の範囲内にあるデータを指定した数だけ選択する．そして，選択されたデータを選択済みデータ，選択されていないデータを未選択データとして生成する．

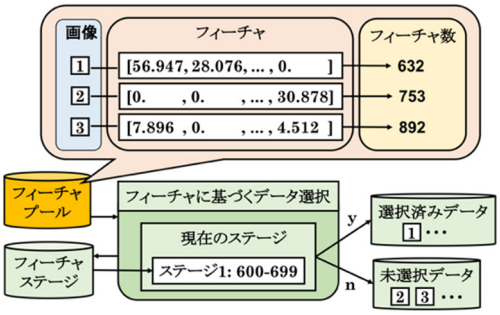


図 8 フィーチャに基づくデータ選択
Figure 8 Feature-Based Data Selection

7. 実データへの適用

7.1 適用目的

プロトタイプを実際の画像データに適用することによって，本稿での提案方法の有効性と妥当性を評価する．

7.2 適用対象

本稿では，画像認識問題である 3 クラス分類に適用する．3 種類のペットボトルの画像を用意し，ペットボトルのラベルをもとにクラスの特定を行う．画像データは Web カメラで取得し，1 クラス 4,000 枚の計 12,000 枚を対象データとする(図 9)．

図 9 対象データ
Figure 9 Target Data

7.3 適用方法

訓練データとテストデータに対し，バッチサイズ 128，エポック数 300 で学習し，テストした．本稿では，2 種類の適用方法で適用し評価する．それぞれの適用条件を表 5 に示す．適用 1 は基準値ごとに 3 回，適用 2 は 4 回実行した．また，ステージはフィーチャ数の範囲ごとにランク付けしたものであり，データ数は n ループ目のデータ数を表

す．また，訓練誤差の期待値を$t(k) = 0.5^k$とした．

表 5　提案方法の適用条件

Table 5 Application Conditions of Proposal Method

	基準値	ステージ	データ数	達成条件
適用 1	1:$S\alpha$=30,$F\alpha$=1.5 2:$S\alpha$=20,$F\alpha$=1.0 3:$S\alpha$=15,$F\alpha$=0.5	1:600-700 2:700-800 3:800-900	$30(n^2 - n + 5)$	精度 94%以上
適用 2	1:$S\alpha$=30,$F\alpha$=1.5	1:600-700 2:700-800 3:800-900	$150n$	6 ループ終了

8. 評価

8.1 評価方法

提案方法と従来の学習方法(ランダムにデータを学習させた場合)による学習モデルの精度を比較する．モデルの評価関数は平均二乗誤差(式 3)を採用する．n は入力データ数，$\hat{y}_i$はi番目の入力x_iに対するモデルの出力，y_iはi番目の入力x_iに対する教師データの値を表す．また，適用 2 においてそれぞれのグラフにおける精度の収束の程度を評価するために，収束率(式 4)を定義する．M(n)は各データ数において精度が最大である点を結んだ折れ線グラフの 2 次の多項式近似曲線であり，m(n)は精度が最小である点の近似曲線である．

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \quad (3)$$

$$収束率 = \frac{M(900) - m(900)}{M(150) - m(150)} \quad (4)$$

8.2 評価結果

8.2.1 事前実験の評価

事前実験として，フィーチャ数の違いによる訓練誤差，汎化誤差の推移と S，F の値の関係を調べた．A.1，A.2 は画像データ数 150，300，1050 におけるフィーチャ数ごとの訓練誤差，汎化誤差を示す．A.3 はデータ数，フィーチャ数，及びそれぞれの S，F の値の関係を示す．

訓練誤差は学習データのフィーチャ数が小さいほど，MSE が早く収束した．一方，汎化誤差はフィーチャ数が大きいほど，早く収束した．そして，訓練誤差の S，F はどちらも同じデータ数においてフィーチャ数が小さいほど値が小さくなる傾向にあるが，汎化誤差の S はフィーチャ数が大きいほど値が小さくなる傾向が強かった．また，汎化誤差の F とフィーチャ数には相関が見られなかった．

8.2.2 適用 1

画像 750，1050，1410 枚での提案方法と従来の学習方法の精度を比較したグラフを図 8 に示す．

従来方法での精度はデータ数ごとの標準偏差はすべて 0.020 以上だったのに対し，提案方法ではすべて 0.020 を下回った．

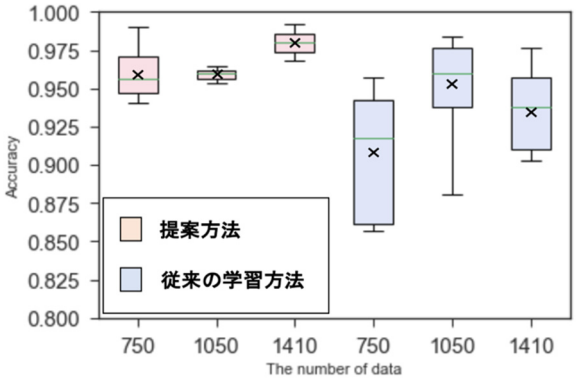


図 10　精度のばらつきの比較

Figure 10 Comparison of Variations in Accuracy

さらに，同じデータ数での精度の平均値に関しては，提案方法が従来の学習方法よりも上回っていることが確認でき，最大改善率はデータ数 750 で 5.4%だった．加えて，提案方法と従来の学習方法の精度のばらつきを比較すると，提案方法の方が精度のばらつきが小さいと言える．

8.2.3 適用 2

従来方法と提案方法でのデータ数の増加に伴う精度の推移(図 11,12)，及び各データ数における精度の平均値(表 6)を示す．

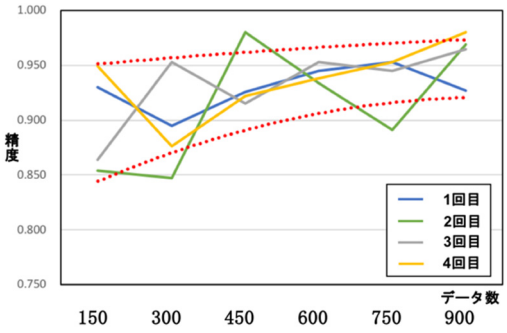


図 11　従来方法での精度の推移

Figure 11 Accuracy of Data Growth Conventional Method

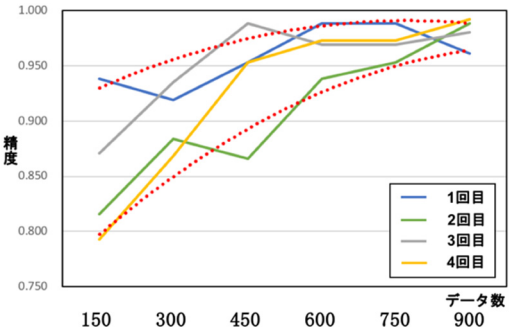


図 12　提案方法での精度の推移

Figure 12 Accuracy of Data Growth in Proposal Method

従来方法の収束率は 0.510 だったのに対し，提案方法の収束率は 0.197 だった．この結果から，従来方法での精度の収束は，データ数 150 から 900 において 2 分の 1 未満だ

ったが, 提案方法では5分の1以上収束したことがわかる．そのため，提案方法の方が，従来の学習方法よりも精度の制御が容易である．さらに，表 6 からデータ数 450，600，750，900 での精度が全体的に上回っているため，比較的高い精度で学習が早く収束していることがわかる．

表 6 各データ数における精度の平均値

Table 6 Average of Accuracy for Each Set of Data

	150	300	450	600	750	900
提案方法	0.839	0.891	0.940	0.954	0.952	0.981
従来方法	0.895	0.897	0.928	0.945	0.941	0.964

9. 考察

9.1 事前実験についての考察

訓練誤差では, 訓練データ全体のフィーチャ数が小さい，すなわち学習コストが小さいほど収束が早いと言える．また，同じデータ数でもフィーチャ数が大きいほど F の値が大きくなる傾向があるため，学習コストが大きいほど局所的にエラー率が大きくなる可能性が高いと言える．一方，汎化誤差ではフィーチャ数が大きいほど収束が早いため，訓練データのフィーチャのパターンが多いほどエラー率が小さくなると言える．

このことから，訓練データのフィーチャ数を制御することで, 予測誤差それぞれの推移を制御できると考えられる．また，予測誤差を制御出来れば，学習過程で未然に過学習を防ぐことが可能であると考えられる．

9.2 RQ1 についての考察

実データへの適用から，フィーチャに基づいたデータ選択によって学習対象のデータを順次追加していくことで，予測誤差を分析しながら学習することが可能であることを明らかにした．また，事前実験と評価の結果から追加データのフィーチャ数を制御することで，予測誤差の推移を一定の範囲で制御可能なことが確認できた．

本稿では，フィーチャ数の範囲を 100 に設定したが，データ数が十分な限りで，より範囲を絞ることで予測誤差の分析をさらに容易にできると考えられる．また，$S\alpha$，$F\alpha$の値はループ遷移の起こりやすさと関係しているため，取得可能なデータ数に応じて値を変更する必要があると考えられる．特に，取得したデータ数が少ない場合は，予測誤差が収束しないため，$S\alpha$，$F\alpha$ともに大きめに設定しループ遷移を積極的に行う必要があると考えられる．

9.3 RQ2 についての考察

9.3.1 精度の安定性

適用 1 より，従来方法では訓練データのランダム性が大きいため，同じデータ数でも学習ごとの精度のばらつきが大きい．また適用 2 より，データ数が増加している一方で精度が減少している箇所が複数存在する．そのため，必ずしもデータ数を増やすことで精度が改善されるとは限らないと考えられる．それに対し，提案方法ではフィーチャに基づいたデータ選択により，訓練データのランダム性が軽減されるため，同じデータ数での精度のばらつきが比較的小さくなり学習が早く収束すると考えられる．しかし，提案方法のそれぞれのループにおいて，データ数の増加に伴って精度が下がる箇所もいくつか観測されたため，フィーチャ数以外の要素でも追加データを制御し，予測誤差の分析を容易にする必要があると考えられる．

一方，提案方法の収束率の結果から，従来方法と比べてデータ数が十分な場合において一定の精度の確保が容易であることを確認した．これより，提案方法は目標の精度を達成しやすいため, より安定した開発であることが言える．

9.3.2 精度の改善率

表 4 より，データ数 450，600，750，900 の場合の精度は改善されたのに対し，データ数 150，300 での精度は従来方法を下回った．この結果から，提案方法ではフィーチャ数の少ないデータから学習していくため，データ数が少ない場合は要求する認識精度に必要なフィーチャ数を獲得することが困難であると考えられる．

10. 今後の課題

今後の課題は以下の点である．

(1) 他のデータ，深層学習モデルへ提案方法を適用

本稿で適用した画像データや深層学習モデルとは異なるデータやモデルに対し, 提案方法を適用し評価することが課題である. 例として, 文書データや RNN への適用が挙げられる.

(2) 異なる予測誤差分析プロセスの検討

本稿で提案した分析プロセスとは異なる分析プロセスを検討し, 本稿での分析プロセスとの比較を行うことが課題である. 特に, フィーチャと予測誤差の新たな関係を模索し, フィーチャ数以外の要素から予測誤差の制御を検討する.

11. まとめ

学習モデルの設計方法は体系化されておらず，試行錯誤に依っている状態である．本稿では，フィーチャに基づくデータ選択によって，段階的に学習可能な深層学習モデル開発プロセスを提案した．提案方法のプロトタイプを実装し，実際の画像データに適用することで，学習モデルの精度の安定性と改善率を評価した．また，従来方法と比較することで，提案方法の有効性と妥当性を示した．

参考文献

[1] G. Dong, et al., Feature Engineering for Machine Learning and Data Analysis, CRC Press, 2018, pp. 55-79.

[2] I. Goodfellow, et al., Deep Learning, MIT Press, 2016.

[3] 丸山 宏, 城戸 隆, 機械学習工学へのいざない, 人工知能, Vol. 33, No. 2, 2018 年 3 月, pp. 124-131.

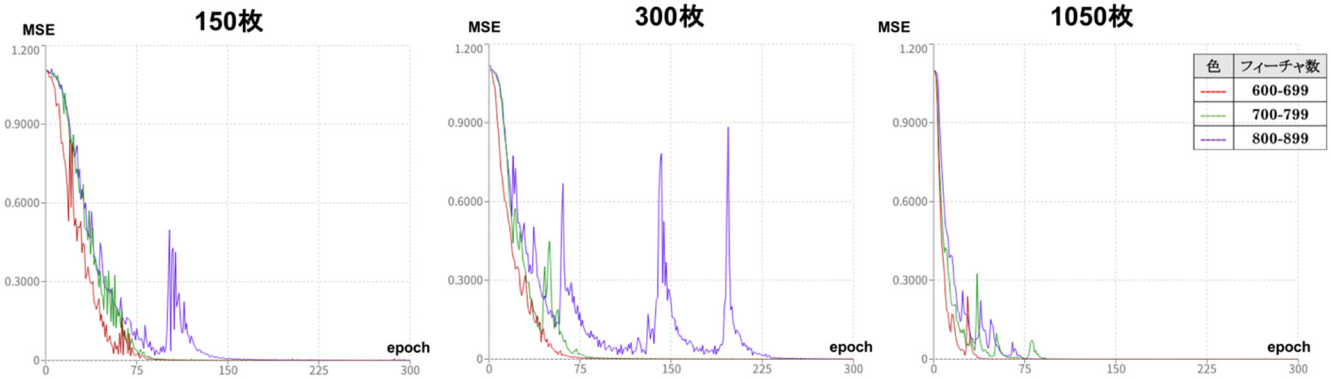
[4] U. Khurana, et al., Feature Engineering for Predictive Modeling Using Reinforcement Learning, AAAI-18, 2018.

[5] J. Li, et al., Feature Selection: A Data Perspective, ACM Computing Surveys, Vol. 50, No. 6, Dec. 2017, 45 pages.
[6] S. Ozdemir, et al., Feature Engineering Made Easy, Packt, 2018.
[7] Preferred Networks, Chainer, https://chainer.org/.
[8] K. Simonyan, et al., Very Deep Convolutional Networks for Large-Scale Image Recognition, arXiv:1409.1556, Apr. 2015, 14 pages.

付録 A

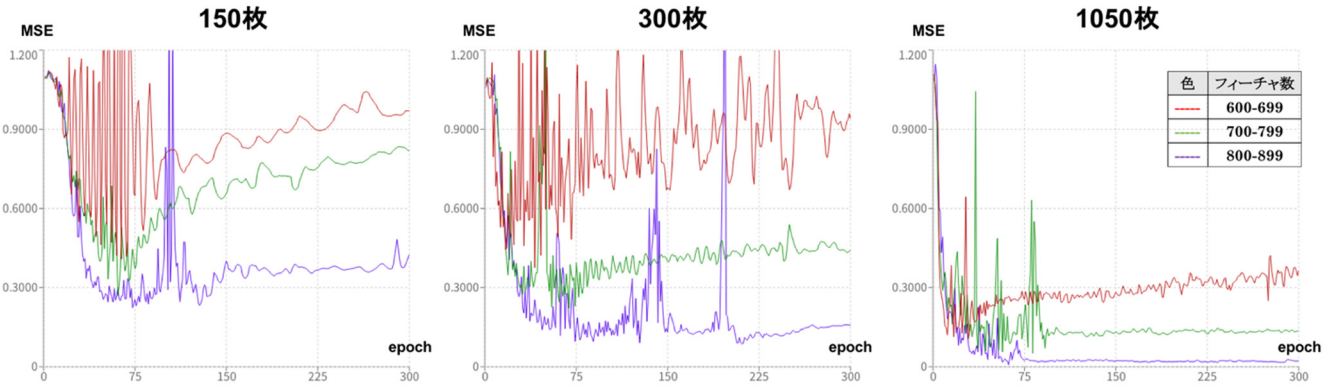
A.1 各フィーチャ数での訓練誤差

A.1 Training Errors in The Number of Each Feature



A.2 各フィーチャ数での汎化誤差

A.2 Generalization Errors in The Number of Each Feature



A.3 各フィーチャ数での S, F の値

A.3 S and F Values in The Number of Each Feature

訓練誤差

S	訓練誤差: S	600-700	700-800	800-900
	150	28.49	37.82	46.07
	300	19.51	27.57	49.75
	600	13.87	15.71	23.95
	1050	6.63	10.57	14.13
	1650	6.39	23.32	11.19

F	訓練誤差: F	600-700	700-800	800-900
	150	1.45	1.59	2.25
	300	0.31	0.84	4.17
	600	0.10	0.66	1.05
	1050	0.30	0.59	0.48
	1650	0.45	4.35	0.08

汎化誤差

汎化誤差: S	600-700	700-800	800-900
150	267.36	206.05	121.34
300	254.23	130.78	68.24
600	61.88	70.94	83.20
1050	86.61	48.67	15.48
1650	73.71	49.29	21.14

汎化誤差: F	600-700	700-800	800-900
150	12.38	2.74	4.94
300	16.07	4.60	7.11
600	3.07	4.81	5.60
1050	3.05	4.68	1.46
1650	3.24	7.10	3.81