Vol.2020-ICS-198 No.17
2020/3/10

継続学習によるRepresentational Forgettingの実験的解析

村田 健悟[1,a)]　豊田 哲也[2]　大原 剛三[2]

概要：ニューラルネットワークモデルを用い複数のタスクを継続的に学習した場合，新規タスクの学習により学習済みタスクを忘却してしまう，破滅的忘却と呼ばれる問題が生じる．近年の研究では，モデルの予測性能のみに基づいて忘却を評価しているため，モデルの内部特徴に生じる忘却についてほとんど理解されていない．そこで本研究では，内部特徴に生じる忘却（Representational Forgetting）に焦点を当て，継続学習の結果生じる忘却の度合い，および忘却によって内部特徴に生じるバイアスを実験的に解析する．

Experiments and Analysis on Representational Forgetting in Continual Learning

1. はじめに

汎用人工知能は，与えられるタスクが時間に従って変化したとしても，現在与えられているタスクのみでなく，過去に学習したタスクについても適切に動作する必要がある．これはすなわち，汎用人工知能には，過去に学習したタスクに対する知識を保持しつつ新しく与えられるタスクに対する知識を学習し続ける，継続学習への適応が必要であることを意味する．しかしながら，このような継続学習のフレームワークをニューラルネットワークモデルに適用した場合，破滅的忘却と呼ばれる，新規タスクの学習によって学習済みタスクに対する予測性能が著しく低下する現象が発生することが知られている [1].

この問題に対し，破滅的忘却を回避し，ニューラルネットワークモデルにおける継続学習を実現するため，これまでに多くの研究がなされている．しかし，これらの研究はモデルの予測性能のみに注目しているため，モデル内における知識の忘却に対する理解は未だに乏しい [2]．具体的には，ニューラルネットワークモデルの中間層の出力ベクトルが対象タスクの特徴を表し，モデルへの個々の入力に対する出力ベクトルがその特徴の具体例を表すと考え，その具体例を入力に対する表現（representation）とした場合，過去に学習したタスクに属する入力に対する表現がどのように忘却されるのかは明らかでない．このような特徴上に生じる忘却は Representational Forgetting [2] と呼ばれる．

本稿では，Representational Forgetting に注目し，実験を通したその解析結果を報告する．本実験では，3 種類のベンチマークデータを用い，複数の代表的な継続学習手法について，ニューラルネットワークモデルの特徴抽出モジュールの性能を評価することで，特徴に生じる Representational Forgetting の有無と大きさを調査する．加えて，当該モジュールの各タスクへの予測性能に基づき，特徴上に存在するタスクへのバイアスの強さおよび方向を明らかにする．本研究を通して得た主な知見は以下の通りである．

1. 層が深くなるにつれ，対応する特徴に生じる Representational Forgetting の大きさは大きくなるが，浅い層の特徴においても特徴に対する忘却は発生し得る．
2. 最も深い層の特徴上には，モデルの最終出力と同様，最後に学習したタスクへの強いバイアスが存在するが，他の特徴上には，最初に学習したタスクへのバイアスが存在する．

以下，2 節で関連研究を紹介し，3 節では Representational Forgetting の詳細な定義を与えたのち，本実験で用いる評価指標について説明する．4 節で実験の設定を述べ，5 節で実験結果について議論する．6 節でまとめを述べる．

[1] 青山学院大学大学院理工学研究科
Graduate School of Science and Engineering, Aoyama Gakuin University, Kanagawa, 252-5258 Japan
[2] 青山学院大学理工学部
College of Science and Engineering, Aoyama Gakuin University, Kanagawa, 252-5258 Japan
[a)] c5619156@aoyama.jp

2. 関連研究

2.1 継続学習手法

継続学習手法は，regularization-based approach，replay-based approach，そして parameter isolation-based approach の 3 種の方法に分類することができる [3]．以下では，それぞれの方法における代表的な手法について順に述べる．

Regularization-based approach は，忘却の抑制を促進する正則化項をネットワークの学習に用いる損失関数に追加することで，新規タスクの学習による学習済みタスクの忘却を軽減する方法である．Elastic Weight Consolidation（EWC）[4] および Synaptic Intelligence（SI）[5] では，学習済みタスクを解くために重要なパラメータの変動に制限を加えるような正則化項が用いられる．Learning without Forgetting（LwF）[6] では，新規タスクの学習開始時点でのネットワークを別途保持し，現行ネットワークの新規タスクデータに対する出力が，保持したネットワークの同出力から離れることを抑制するような正則化項が利用される．

Replay-based approach は，学習済みタスクのデータの一部を保持し，新規タスクの学習時に用いることで忘却を軽減する方法である．Experience Replay（ER）[7] では，新規タスクの学習時に，新規タスクのデータに加え，保持データも学習データとして利用する．Gradient Episodic Memory（GEM）[8] では，保持データを用いた制約付き最適化を考えることで，学習済みタスクに対する予測精度の低下を回避する．これらの方法は，regularization-based approach と比べ，忘却を大きく緩和することができるが，実データの保持を必要とするため，データプライバシやメモリの制約といった実世界における問題への対処ができない．このような問題に対し，Deep Generative Replay（DGR）[9] は，学習済みタスクのデータを保持する代わりに，それらを学習した生成モデルを導入し，新規タスクの学習時にそのモデルを用いて生成したサンプルも学習データとして利用することで，前述のような実データ保持に関する問題に対処する手法である．

Parameter isolation-based approach は，ネットワークのパラメータ集合を分割した部分集合を各タスクに割り当てることで，学習済みタスクの忘却を完全に回避する方法である [10, 11]．これらの手法は，忘却を完全に阻止することができるが，モデルを利用した推定時に対象データの所属タスクの情報を必要とするため，incremental task learning scenario [7] のみにしか適用できない．

2.2 破滅的忘却の解析

これまでに，破滅的忘却自体に対する研究はあまりなされていない．最近の研究 [12, 13] では，モデルの最終出力上に，最後に学習したタスクへのバイアスが存在し，そのバイアスが破滅的忘却の一因であると述べられている．一方，Xiong ら [2] は，Representational Forgetting について初めて言及し，ネットワークの最も深い層の特徴における Representational Forgetting についても，破滅的忘却を引き起こす重要な要因であることを報告している．また，Nguyen ら [14] は，タスクの性質とモデルの予測性能の関係について調査し，タスクの複雑さと同性能の間には強い相関があることを示している．これらの研究は，継続学習および破滅的忘却の様々な性質を明らかにしているものの，Representational Forgetting に対する理解は未だ進んでいないといえる．

3. 評価指標

本節では，まず，本研究における問題設定について述べたのち，特徴，表現，および特徴上に生じるバイアスの点から Representational Forgetting を定義し，その後，本研究で行った実験および解析に用いた評価指標を説明する．

3.1 問題設定

本研究では，M 個のタスクからなるタスク列 $\mathcal{T} = (T_i)_{i=1}^{M}$ に対する継続学習について考える．各タスク T_i は，クラス集合 C_i を対象とするクラス分類タスクであると仮定する．ここで，任意の 2 タスク T_i，$T_j (i \neq j)$ に対し，クラス集合 C_i，C_j は互いに素であるとする．これは，タスクの増加とともに分類対象となるクラスが増加することを意味する．学習モデルは，これら M 個のタスクを 1 つずつ逐次的に学習する．なお，クラスの推定時に対象データの所属タスクは与えられないため，同モデルは学習時に観測したすべてのクラスから 1 つのクラスを予測する必要がある．このような問題設定は，incremental class learning scenario [7] と呼ばれ，継続学習において一般的な問題設定の 1 つである．一方，このようなオンラインでの学習に対し，オフライン学習では，全タスクのデータを一度に用いてモデルの学習が行われる．ここで，破滅的忘却とは，タスク T_j の学習によって学習済みのクラス集合 $C_i (i < j)$ に対するモデルの予測性能が著しく低下する現象として定義できる．

3.2 Representational Forgetting とバイアス

本稿では，学習モデルとしてディープニューラルネットワークモデルを考え，その特定のモジュールの出力を**特徴**と呼ぶ．また，モデルに与えられる個々の入力ベクトルに対するモジュールの出力を，同入力の**表現**と呼ぶ．すなわち，表現とは特徴のインスタンスであり，タスクから学習した知識を表すものとして考えられる．加えて，任意の特徴において，ある特定のクラス群に対する表現のみが正しく学習されている場合，その特徴に**バイアス**が存在すると定義する．このとき，モデル中のある特徴においてバイアスが生じ，学習済みタスクの入力に対する表現が忘却さ

れる現象が *Representational Forgetting* であると定義できる．このような Representational Forgetting とバイアスの関係性から，Representational Forgetting および破滅的忘却をより深く理解するためには，特徴に生じるバイアスの方向，すなわち，どのタスクに対する表現が正しく学習されているのかを特定し，その強さを定量的に評価することが重要であると考えられる．

3.3 Partial Retrain Accuracy

ここでは，まず，解析対象の特徴に対応する特徴抽出モジュールの性能評価指標である Partial Retrain Accuracy（PRA）について説明する．ここで，任意の特徴抽出モジュール M_k に対する PRA は，以下の手順で計算する．

1. 全タスクについて，逐次的にモデルを訓練する．
2. M_k より深い層に存在する全パラメータを初期化する．
3. 入力層から M_k までの層に対する全パラメータの値を固定したうえで，全タスクのデータを使用し，オフライン学習によってモデル全体を再訓練する．
4. 再訓練後モデルの accuracy をテストデータを用いて計算し，その値を PRA とする．

PRA の値が，始めからオフライン学習により訓練を行ったモデルの accuracy と比べて低かった場合，M_k の出力である特徴において Representational Forgetting が発生していると考えられる．

PRA の計算は，パラメータの初期化フェーズ（Step 2）を除き，Xiong らが提案した指標 [2] と同等である．この初期化フェーズは，計算対象のモジュールより深い層におけるパラメータから，Step 1 での学習結果の影響を除去するために追加している．実際，初期化を行わない場合，これらのパラメータが再学習（Step 3）を阻害し，PRA の値を不当に低くすることがある．

3.4 バイアス解析のための指標

前項で述べたように，PRA を計算することで，任意の特徴における Representational Forgetting の大きさを定量的に測定することが可能である．しかしながら，PRA は特徴上のバイアスについて，その方向と強さといった情報を示すことはできない．そのため，それらの情報を知るための指標として，特徴抽出モジュール M_k を対象とした再訓練後モデルの micro-F1（F1）を用いる．ここで，各タスク T_j に対し，F1 は以下のように定義される．なお，F1 は PRA の計算時の予測結果に基づいて計算するため，厳密には特定の特徴抽出モジュール M_k が仮定されるが，以下の定義では簡単のため，M_k についての記述は省略している．

$$\mathrm{F1}(T_j) = \frac{2\mathrm{Recall}(T_j)\cdot\mathrm{Precision}(T_j)}{\mathrm{Recall}(T_j)+\mathrm{Precision}(T_j)} \tag{1}$$

$$\mathrm{Recall}(T_j) = \frac{\sum_{c\in C_j}\mathrm{TP}_c}{\left(\sum_{c'\in C_j}\mathrm{TP}_{c'}\right)+\left(\sum_{c'\in C_j}\mathrm{FN}_{c'}\right)} \tag{2}$$

$$\mathrm{Precision}(T_j) = \frac{\sum_{c\in C_j}\mathrm{TP}_c}{\left(\sum_{c'\in C_j}\mathrm{TP}_{c'}\right)+\left(\sum_{c'\in C_j}\mathrm{FP}_{c'}\right)} \tag{3}$$

ここで，TP_c, FP_c, FN_c はそれぞれ，クラス c に属するテストデータに対する，真陽性，偽陽性，偽陰性の数である．

F1 は任意のタスクに対する分類性能を評価するため，その値が低いことは，対象タスクに対する表現が，同タスク以降に与えられたタスクの学習によって忘却されたか，学習されていないことを示している．また，ある 2 タスク間で F1 の値が大きく異なっている場合，高い F1 の値を示したタスクへのバイアスが解析対象の特徴に存在すると考えられる．そこで，特徴上に生じたバイアスの強さを定量的に評価する指標として，F1Ratio を以下のように定義する．

$$\mathrm{F1Ratio} = \frac{\min_{T_i\in\mathcal{T}}\mathrm{F1}(T_i)}{\max_{T_i\in\mathcal{T}}\mathrm{F1}(T_i)}. \tag{4}$$

定義から明らかなように，F1Ratio が小さいことは，解析対象の特徴上に強いバイアスが存在することを意味する．

4. 実験設定

本研究では，各特徴における Representational Forgetting，および，特徴上に生じるバイアスを解析するため，複数の代表的な継続学習手法について，3 種類のデータセットを利用した実験を行った．本節では，その実験における設定について述べる．

4.1 データセットとタスクの構成

本実験では，MNIST [15]，SVHN [16]，および CIFAR10 [17] の 3 種類のデータセットを用いた．これらは，継続学習研究において広く用いられている画像データセットであり，いずれも 10 種類のクラスから構成される．ここでは，先行研究 [5,7] に従い，連続する 2 つのクラスを対象とする分類タスクを 1 タスクとして定義した．MNIST を例にとると，1 番目のタスクはクラス ‘0’ とクラス ‘1’ のデータによって構成され，2 番目のタスクはクラス ‘2’ とクラス ‘3’ のデータによって構成される．このようにして，各データセットについて 5 つのタスクを定義した．

多くの先行研究では，ある 1 つのタスク実行順序のみを用いて実験を行っている [7,18]．しかしながら，このような実験設定では，タスク間に生じた F1 の値の差が，タスクの実行順序ではなく，個々のタスクの分類の容易さの違いによって生じる可能性がある．そこで，そのようなタスクの複雑さによる影響を除去するため，**表 1** に示すように，タスクの実行順序を巡回させることで 5 種類のタスク

表 1　各試行におけるタスク実行順序
Table 1　Task execution sequence in each trial.

Trial number	T^{1st}	T^{2nd}	T^{3rd}	T^{4th}	T^{5th}
Trial #1	T_1	T_2	T_3	T_4	T_5
Trial #2	T_2	T_3	T_4	T_5	T_1
Trial #3	T_3	T_4	T_5	T_1	T_2
Trial #4	T_4	T_5	T_1	T_2	T_3
Trial #5	T_5	T_1	T_2	T_3	T_4

順序を作成し，各タスク順序を用いて 5 度の試行を行った．なお，表 1 において，記号 $T_i(i \in \{1,2,3,4,5\})$ は定義した i 番目のタスクを表し，$T^{j\text{-th}}(j \in \{1,2,3,4,5\})$ はタスク実行順序において j 番目に実行されるタスクを表す．加えて，各試行を異なる乱数シードを用いて 5 度行い，これら計 25 試行から得られる評価値の平均値を最終的な結果として用いた．

4.2　ネットワーク構造と特徴

本項では，実験に使用したネットワークの構造について述べたのち，ネットワークにおける特徴の定義について説明する．MNIST に対する実験では，3 層の畳み込み層に全結合層を重ねた Convolutional Neural Network（CNN）を用いた．なお，畳み込み層のチャネル数はそれぞれ 16，32，64 とした．また，カーネルサイズはすべて 3 とし，2 層目，3 層目のストライドを 2 とすることでダウンサンプリングを適用した．一方，SVHN および CIFAR10 に対する実験では，reduced ResNet18 [8]（以下，ResNet と呼ぶ）を用いた．ここで，**表 2** に ResNet の構造を示す．同表において，3 列目は対応するモジュールの持つ畳み込み層を示している．また，畳み込み層のパラメータは，“conv⟨ カーネルサイズ ⟩-⟨ チャネル数 ⟩” として記載している．加えて，module3，module4，module5 の最初の畳み込み層において CNN と同様の方法でダウンサンプリングを適用した．なお，batch normalization layer，global average pooling layer，およびクラス分類層については省略している．

次に，各ネットワークにおける特徴の定義について説明する．CNN では，各畳み込み層の活性化後出力を特徴として定義し，ResNet では各モジュールの出力を特徴として定義した．すなわち，CNN はモデルの最終出力を含めた 4 つの特徴を持ち，ResNet は同様に 6 つの特徴を持つ．なお，各特徴に対し，対応する層の深さが浅い順に番号付けを行った．

4.3　学習方法

本実験では，以下の継続学習手法について比較を行った．**Fine-tuning** は，モデルを通常の方法で各タスクについて逐次的に訓練する手法である．**SI** および **LwF** は，正則化項を利用した regularization-based method である．**GEM** および **ER** は，保持データを利用した replay-based method である．また，これらの手法に加え，全タスクのデータについてオフライン学習でモデルの訓練を行う **Offline** についても実験を行った．Offline は，評価指標における上限値を与える手法として考えることができる．また，これらの手法のハイパーパラメータとして，SI，LwF における正則化係数を 1.0 とし，GEM，ER における各クラスごとの保持データ数を，MNIST，SVHN，CIFAR10 のそれぞれについて，128，128，512 とした．

表 2　Reduced ResNet18 の構造
Table 2　Structure of reduced ResNet18.

Module name	Output dimension (C, H, W)	Block
module1	$20 \times 32 \times 32$	[conv3-20]
module2	$20 \times 32 \times 32$	$\begin{bmatrix}\text{conv3-20}\\ \text{conv3-20}\end{bmatrix} \times 2$
module3	$40 \times 16 \times 16$	$\begin{bmatrix}\text{conv3-40}\\ \text{conv3-40}\end{bmatrix} \times 2$
module4	$80 \times 8 \times 8$	$\begin{bmatrix}\text{conv3-80}\\ \text{conv3-80}\end{bmatrix} \times 2$
module5	$160 \times 4 \times 4$	$\begin{bmatrix}\text{conv3-160}\\ \text{conv3-160}\end{bmatrix} \times 2$

本実験では，PRA，F1，F1Ratio を計算するにあたり，継続学習によりモデルを訓練した後，オフライン学習によりモデルの再訓練を行う必要がある．これら両訓練において，最適化手法として Adam [19] を使用し，学習率を 0.001 に設定した．また，バッチサイズについては 64 とし，ER における保持データの学習についても同じバッチサイズを使用した．加えて，継続学習時には，MNIST，SVHN，CIFAR10 のそれぞれについて，エポック数を 5，10，10 に設定した．同様に，全データを用いる再訓練時にも，同じエポック数を使用した．さらに，両学習フェーズの開始時には，畳み込み層のパラメータを He の初期値 [20] によって初期化し，全結合層のパラメータを平均が 0，標準偏差が 0.01 の正規分布を用いて初期化した．

5.　実験結果

本節では，PRA による評価を用いた Representational Forgetting の解析，および F1Ratio と F1 による評価を用いた特徴上のバイアスに対する解析の結果を述べる．

5.1　PRA による評価

すべてのデータセットおよび手法における，PRA による評価結果を**図 1** に示す．ここで，feature index が最大値をとる場合，PRA はモデルの出力層の特徴に対する accuracy，すなわち，継続学習のみを適用した場合のモデルの accuracy を示している．また，Offline ではモデルの再訓練を行わないため，すべての feature index において

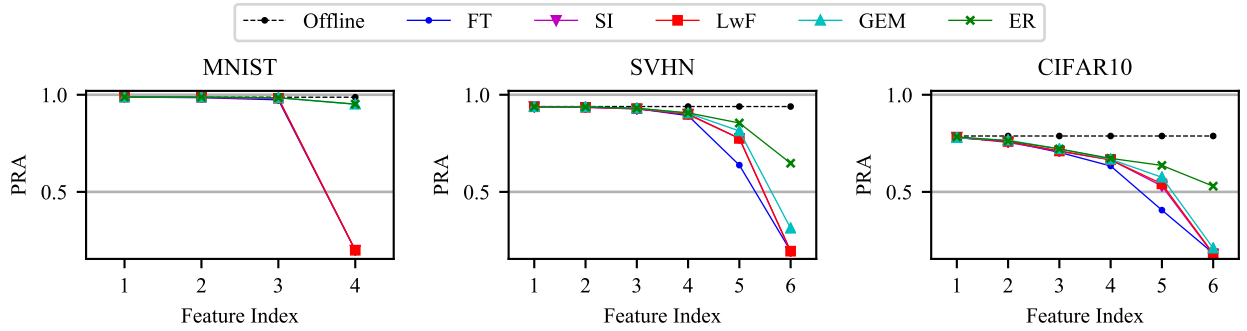

図 1　各データセットにおける各特徴抽出モジュールに対する PRA
（左：MNIST，中央：SVHN，右：CIFAR10）
Fig. 1　Scores of PRA for each feature extraction module on the datasets (left: MNIST, middle: SVHN, right: CIFAR10).

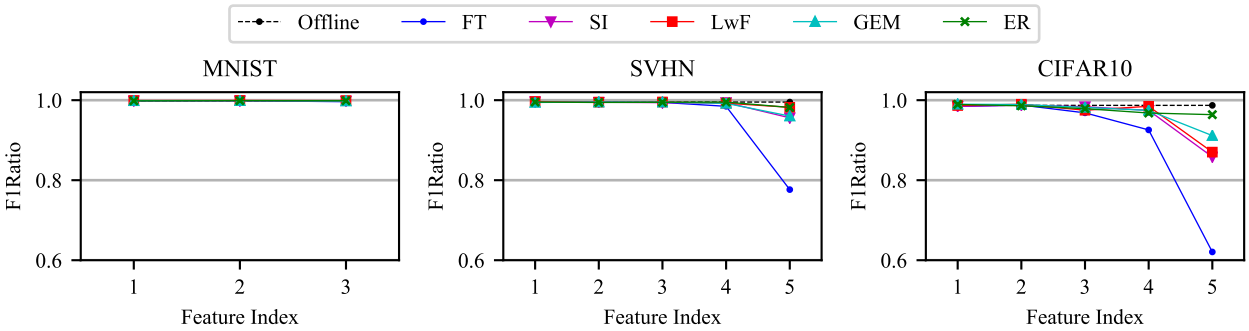

図 2　各データセットにおける各特徴に対する F1Ratio
（左：MNIST，中央：SVHN，右：CIFAR10）
Fig. 2　Scores of F1Ratio for each feature on the datasets (left: MNIST, middle: SVHN, right: CIFAR10).

同じ値をとる．

まず，すべての継続学習手法において，PRA が単調減少傾向にあり，feature index が大きくなるにつれ，Offline とのスコア差が大きくなっていることが図 1 からわかる．このことは，層が深くなるにつれ，対応する特徴に生じた Representational Forgetting の大きさが大きくなっていることを示している．言い換えると，浅い層では，クラス間に共通する汎用的な表現が学習されていると考えられる．

次に，各データセットの結果を比較すると，各継続学習手法と Offline とのスコア差が，異なる feature index から表れ始めていることがわかる．実際，CIFAR10 では 2 番目の特徴からスコアに差が生じているが，SVHN では 4 番目の特徴から差が生じている．さらに，MNIST では最後の特徴においてのみスコア差が表れており，このことは，MNIST に対しては，任意の継続学習手法において Representational Forgetting が起きないことを意味している．加えて，スコア差の大きさを比較すると，SVHN を用いた場合より，CIFAR10 を用いた場合の方が最後の特徴を除いてスコア差が大きい．このことは，CIFAR10 に対する Representational Forgetting の大きさがより大きくなることを意味している．これらの結果は，Representational Forgetting の大きさが学習対象としている一連のタスク全体の複雑さに依存しており，また，複雑であるほどより浅い層の特徴でより大きく忘却が発生することを示している．

さらに，各継続学習手法の結果を比較すると，任意の feature index において，PRA の値の大小関係が同一であることがわかる．実際，任意の feature index において，ER が他の継続学習手法より大きな PRA の値をとっており，GEM が次に大きな値をとっている．しかしながら，feature index が小さくなるにつれ，継続学習手法間の PRA の値の差は小さくなっている．これらの結果から，モデルの出力層をはじめとする深い層の特徴において，ER などの洗練された継続学習手法が忘却の緩和に大きく寄与する一方，浅い層において生じる軽微な忘却に対しては，どの手法もその効果に大きな差はないことがわかる．

5.2　F1Ratio および F1 による評価

F1Ratio を用いた，各特徴上でのバイアスの強さに関する実験結果を図 **2** に示す．なお，モデルの出力層の特徴については，最後に学習したタスクに大きなバイアスがかか

ることが知られているため同図から除いた．まず，Offline の結果に注目すると，全データセットにおいて 1 に近い値をとっていることから，バイアスがほとんど存在しないことがわかる．このことは，タスクの順序を巡回させる本実験設定が，タスク実行順位におけるタスクの複雑さの影響を除去できていることを意味する．次に，MNIST の結果に注目すると，すべての手法において，どの特徴上にもバイアスがほとんど存在しないことがわかる．一方，SVHN および CIFAR10 を用いた場合，深い層において F1Ratio の値が減少していることから，それらの層の特徴上にバイアスが存在することがわかる．実際，SVHN では 4 番目の特徴からバイアスが生じており，CIFAR10 では 3 番目の特徴からバイアスが生じている．加えて，PRA の場合と同様に，層が深くなるにつれ，対応する特徴上でのバイアスの強さがより強くなることがわかる．

以上の結果から，SVHN および CIFAR10 を用いた場合，深い層の特徴上にバイアスが生じることがわかった．そこで，バイアスの方向を調べるため，バイアスが生じた特徴について，各タスクに対する F1 の値を分析した．まず，SVHN を用いた実験における，4 番目と 5 番目の特徴での各タスクに対する F1 の値を**図 3** に示す．4 番目の特徴については，F1Ratio の結果から明らかなように，FT を除くすべての手法においてバイアスがほとんど存在しない．一方，FT については，最後に学習したタスクへの強いバイアスが存在する．また，5 番目の特徴では，ほとんどの手法において F1 の値が単調増加傾向を示している．このことは，5 番目の特徴上に，$T^{4\text{th}}$，$T^{5\text{th}}$ のようなタスク系列終盤で学習したタスク（以下，終盤タスクと呼ぶ）へのバイアスが存在することを示唆している．

次に，CIFAR10 を用いた実験における，3，4，5 番目の特徴での各タスクに対する F1 の値を**図 4** に示す．まず，5 番目の特徴に注目すると，SVHN の場合と同様に，ほとんどの手法において F1 の値が単調増加傾向を示している．一方，3 番目の特徴では，FT を除き谷状のグラフ形状を示している．この形状は，3 番目の特徴上に，最後に学習したタスクと最初に学習したタスクへのバイアスが存在することを示唆している．特に ER では，最後に学習したタスクに対する F1 の値と比べ，最初に学習したタスクに対する F1 の値が明らかに大きい．このことは，ER を適用した場合，最初に学習したタスクへの非常に強いバイアスが 3 番目の特徴上に生じることを意味している．さらに，4 番目の特徴に注目すると，GEM については，最初に学習したタスクへのバイアスが消失し，5 番目の特徴と同様に，F1 の値が単調増加傾向を示している．しかしながら，SI と ER については，依然として谷状のグラフ形状をとっている．なお，FT については，すべての特徴上で終盤タスクへのバイアスが存在する．

CIFAR10 を用いた実験の結果は，バイアスが生じた特徴において，深い層の特徴上には終盤タスクへのバイアスが存在する一方，その他の特徴上には初めに学習したタスクへのバイアスが存在することを明らかにしている．これらの発見は，新規タスクの学習によって，深い層の特徴は学習済みタスクに対する表現を単純に忘却していくのに対し，3 番目の特徴といったモデル全体の中間付近に位置する層の特徴（以下，中間に位置する特徴と呼ぶ）は，学習済みタスクに対する表現を一部保持しながら学習を進めていることを示唆している．加えて，FT において，バイアスが生じたすべての特徴上に初めに学習したタスクへのバイアスが存在しなかったことから，中間に位置する特徴における表現保持は，各継続学習モデルが行う忘却阻止の機能によるものと考えられる．すなわち，それら忘却阻止の機能が，中間に位置する特徴における忘却の軽減を促進していることを示唆している．一方，それらの特徴において，終盤タスクに対する F1 の値が比較的低いことから，表現保持が新規タスクの学習を阻害していることがわかる．すなわち，本実験で用いた継続学習手法では，中間に位置する特徴において，新規タスクに対する表現と学習済みタスクに対する表現の統合が満足に行えていないといえる．

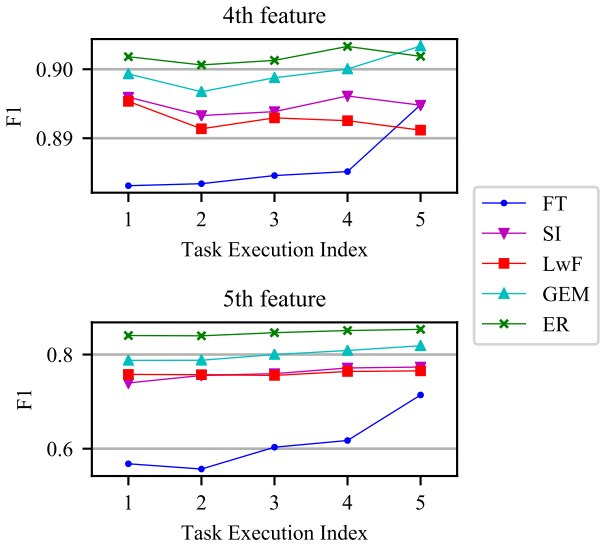

図 3 SVHN における各タスクに対する F1
（上：4 番目の特徴，下：5 番目の特徴）

Fig. 3 Scores of F1 for each task on SVHN
(upper: 4th feature, lower: 5th feature)

6. おわりに

本稿では，ニューラルネットワークの各層の特徴に生じる忘却について，実験的解析の結果，忘却の大きさおよび特徴上のバイアスの大きさと方向が層の深さによって異なることを示した．これらの結果は，各継続学習手法の学習傾向を明らかにするものであり，より洗練された継続学習手法の開発への助けとなることが期待される．なお，本実

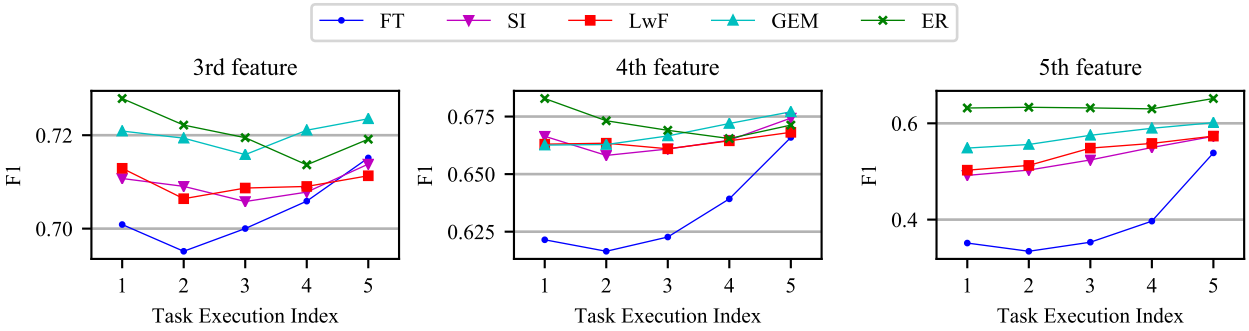

図 4　CIFAR10 における各タスクに対する F1
（左：3 番目の特徴，中央：4 番目の特徴，右：5 番目の特徴）

Fig. 4　Scores of F1 for each task on CIFAR10
(left: 3rd feature, middle: 4th feature, right: 5th feature).

験では，5 タスクからなるタスク列のみを対象としたが，実世界にはより長いタスク列が存在する．また，浅い層の特徴については，Representational Forgetting が確認できたものの，バイアスがほとんど存在しなかったため，浅い層の特徴に生じる忘却に対するより深い理解には至らなかった．今後は，より長いタスク列を対象とした実験的解析，および，浅い層の特徴における Representational Forgetting の更なる解析を行う予定である．

参考文献

[1] McCloskey, M. and Cohen, N. J.: Catastrophic interference in connectionist networks: The sequential learning problem, *Psychology of learning and motivation*, Vol. 24, pp. 109–165 (1989).
[2] Xiong, Y., Ren, M. and Urtasun, R.: Learning to Remember from a Multi-Task Teacher, *arXiv preprint arXiv:1910.04650* (2019).
[3] De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G. and Tuytelaars, T.: Continual learning: A comparative study on how to defy forgetting in classification tasks, *arXiv preprint arXiv:1909.08383* (2019).
[4] Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A. et al.: Overcoming catastrophic forgetting in neural networks, *PNAS*, Vol. 114, No. 13, pp. 3521–3526 (2017).
[5] Zenke, F., Poole, B. and Ganguli, S.: Continual Learning Through Synaptic Intelligence., *ICML*, Vol. 70, pp. 3987–3995 (2017).
[6] Li, Z. and Hoiem, D.: Learning without forgetting, *IEEE TPAMI*, Vol. 40, No. 12, pp. 2935–2947 (2017).
[7] Hsu, Y.-C., Liu, Y.-C., Ramasamy, A. and Kira, Z.: Re-evaluating continual learning scenarios: A categorization and case for strong baselines, *arXiv preprint arXiv:1810.12488* (2018).
[8] Lopez-Paz, D. and Ranzato, M.: Gradient episodic memory for continual learning, *NeurIPS*, pp. 6467–6476 (2017).
[9] Shin, H., Lee, J. K., Kim, J. and Kim, J.: Continual learning with deep generative replay, *NeurIPS*, pp. 2990–2999 (2017).
[10] Serra, J., Suris, D., Miron, M. and Karatzoglou, A.: Overcoming Catastrophic Forgetting with Hard Attention to the Task, *ICML*, Vol. 80, pp. 4548–4557 (2018).
[11] Mallya, A. and Lazebnik, S.: Packnet: Adding multiple tasks to a single network by iterative pruning, *CVPR*, pp. 7765–7773 (2018).
[12] Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y. and Fu, Y.: Large Scale Incremental Learning, *CVPR* (2019).
[13] Hou, S., Pan, X., Loy, C. C., Wang, Z. and Lin, D.: Learning a unified classifier incrementally via rebalancing, *CVPR*, pp. 831–839 (2019).
[14] Nguyen, C. V., Achille, A., Lam, M., Hassner, T., Mahadevan, V. and Soatto, S.: Toward understanding catastrophic forgetting in continual learning, *arXiv preprint arXiv:1908.01091* (2019).
[15] LeCun, Y., Cortes, C. and Burges, C. J.: The MNIST database of handwritten digits, 1998, *URL http://yann.lecun.com/exdb/mnist*.
[16] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B. and Ng, A. Y.: Reading Digits in Natural Images with Unsupervised Feature Learning, *NeurIPS Workshop* (2011).
[17] Krizhevsky, A.: Learning multiple layers of features from tiny images, Technical report, Citeseer (2009).
[18] van de Ven, G. M. and Tolias, A. S.: Three scenarios for continual learning, *arXiv preprint arXiv:1904.07734* (2019).
[19] Kingma, D. P. and Ba, J.: Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980* (2014).
[20] He, K., Zhang, X., Ren, S. and Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, *ICCV*, pp. 1026–1034 (2015).