Vol.2016-ICS-182 No.1
2016/3/2

DeepLearningにおける中間層の情報表現を利用した，物体の外観変化を予測する転移学習モデル

芦原 佑太[1,a)]　佐藤 聡[2]　栗原 聡[1]

概要：近年, 画像認識などで注目されている DeepLearning の研究領域において，各問題に応じた手法が確立しつつある一方で，ネットワークの中間層における情報表現を活用する手法については確立されていない．そこで，本研究では，画像認識に用いた DeepLearning における中間層を再利用し，物体の外観変化の予測問題へ応用する転移学習モデル D-CAFL を提案する．この手法によって，中間層にある情報表現を利用しながら，物体が回転中の画像の想起をすることに成功した．

キーワード：DeepLearning，深層学習，転移学習

Middle layers shareing for transfer learning to predict rotating image in DeepLearning

Yuta Ashihara[1,a)]　Akira Sato[2]　Satoshi Kurihara[1]

1. はじめに

2000 年代後半より，DeepLearning と呼ばれるニューラルネットワークを多層化したものが，機械学習の手法として注目されるようになり，Google などの企業でも活発に研究されるようになった [1].

ニューラルネットワークの研究で，一般的に用いられる三層パーセプトロンでは，入力層，中間層，出力層の 3 層で構成され，各層内に配置されたニューロンによる値の受け渡しによって問題を解決する．ニューラルネットワークの問題解決能力は，ニューロンの結合にかかる係数 (重み) によって決まる．それぞれの重みをデータからどのように学習するかがニューラルネットワークの重要な要素となっている．DeepLearnig は，ニューラルネットワークの中間層の数を多くした階層構造になっており，先に述べたニューラルネットワークの重みの数に比べ，中間層から中間層への値の受け渡しが追加される．その際に起こる問題は，中間層が多層になることで，重みの学習が困難になるということであった [2]．重みの学習に関する問題を解決する方法として，Pre-training と Fine-tuning という方法が開発され，多層化された重みの学習に成功した [3], [4]. そして，DeepLearning が学習した中間層の重みは，学習に使用したデータの特徴量が表現されているということが明らかになった [1]

また，DeepLearning に適用させるデータの範囲は 2010 年代から徐々に増えてきており，自然言語処理などの時系列データに対して応用する事例 [8] や，強化学習のアルゴリズムと DeepLearning を合わせて学習させることで，ゲームを人間同様に操作できるような DeepLearning を構築するなど [9]，データの適用幅が広くなった．さらに，

[1] 電気通信大学 大学院情報システム学研究科 社会知能情報学専攻
The University of Electro-Communications, Graduate School of Information Systems, Department of Social Intelligence and Informatics
〒 182-8585 東京都調布市 調布ケ丘 1 丁目 5-1
[2] 株式会社クロスコンパス
XCompass Ltd.
[a)] y.ashi@ni.is.uec.ac.jp

近年では音声と画像の同時入力や，画像と特徴ベクトルの同時入力などといった，マルチモーダルなデータの入力に対応する DeepLearning も開発されており [10]，複数の DeepLearning をつなげて画像から文章を生成するなど [11]，DeepLearning の可能性はさらに広がっている．しかし，DeepLearning にも，学習用に大量のデータを用意しなければいけない点や，中間層の数が増えるほど学習して調節すべきパラメータの数が膨大になり，学習に時間がかかる点など，デメリットも存在する．そこで，本研究では，DeepLearning の中間層を再利用することで，新しい問題に DeepLearning を応用するときに，学習用のデータが少ない状態でも学習が可能な手法の提案を目的とする．画像認識に用いた DeepLearning の中間層は入力画像の特徴量を抽出する能力を得ており [12]，この特徴量を新しい問題に応用する際に使い回し，少ない学習データから学習が可能なモデル，Deep-CNN-AE-FV-LSTM(以降，D-CAFL と記載する) を提案する．2 節では関連研究を紹介し，3 節では提案するモデルについて述べる．4 節ではモデルを使った実験について述べ，最後に 5 節でまとめを述べる．

2. 関連研究

これまでにも，DeepLearning の中間層に関する考察は様々な角度から行われている．Kavukcuoglu ら [13] は，画像認識で高い精度を出した DeepLearning の中間層が抽出している特徴量を分析し，入力層に近い中間層側では線分や点などの抽象的な特徴が捉えられていることを明らかにした．また，中間層の可視化に関する研究では，Zeiler ら [12] が中間層の可視化について行い，[13] と同様，入力層に近い中間層は抽象的な特徴量を抽出していることに加え，出力層に近い中間層が学習した特徴量は具体的な特徴であることを明らかにした．

また，Yoshinki ら [14] では，ある画像データで学習させた DeepLearning の中間層を他の画像データに適用する実験を行っている．Yoshinki らの主張は，画像認識において学習された線分や点といった抽象的な特徴量はどの画像にも見られるような共通する特徴であることから，どの画像データで学習したとしても，抽象的な特徴量は変わり得ることはないため，どの画像データに対しても適用可能であるということであった．実際に，[14] では入力層に近い中間層を再利用して他の画像データを認識する実験を行い，高い精度を実現している．この結果は，DeepLearning の中間層が転移学習を可能とすることを示す結果でもあり，抽象的な特徴量を学習した中間層の再利用が，有用であることを示している．他にも，野田ら [15] では，感覚運動統合学習システムと して，900 次元の画像情報を DeepLearning によって 30 次元に圧縮し，その圧縮された中間表現とロボットの関節情報を統合した情報が別の DeepLearning に入力されることで，環境に応じたロボットの行動選択を可能としている．そこで用いられている DeepLearning は，入力される高次元データを低次元データに圧縮しながら，特徴を上手く取り出すように学習している．

また，中間層にさらに情報を付加して，学習をより精度よく行う研究についても様々な研究が行われている．Kiros ら [16] では，画像の特徴を抽出した中間層にテキスト情報をベクトル化して付加することで，画像データを説明する英文を出力として生成する DeepLearning のモデルを提案している．特徴量を抽出した中間層は画像データそのものよりも低次元に圧縮することができ，テキスト情報のベクトルを付加して計算する際にも次元が大きくなりすぎることを懸念する必要がない．また，画像データそのものにテキスト情報を付加して入力するよりも，画像認識の DeepLearning の中間層に情報を付加する工夫がなされていることによって，画像データに対して次元数の少ないテキスト情報のベクトルが無視されることを防ぐこともできると Kiros らは主張している．これらの研究から DeepLearning の中間層が得た特徴量がどのようなものか明らかになってきたことや，抽象的な特徴量を学習した中間層は他のデータに対しても適用が可能であるということ，中間層で得た特徴量にさらに情報を付加することで，解ける問題の幅が広がることが示されているが，中間層の再利用によって学習時間が削減された事例や，目的とする問題に対して用意できる学習データが少なく場合にも，中間層の再利用によって少ないデータ数で学習が可能になったという事例はない．本研究では，[14] のように事前に別のデータで学習した DeepLearning の中間層を再利用し，学習用のデータの数が少ない場合についても DeepLearning がうまく学習できるようなモデルを提案する．

3. DeepLearning

本章では提案モデルに用いる DeepLearning のアルゴリズムについて説明する．まず，Deep Neural Network について解説し，その後，入力そのものを復元する自己符号化器とも呼ばれる Autoencoder，画像認識で広く用いられる Convolutional Neural Network，時系列データを扱うときに用いられる Long Short-Term Memory について述べる．

3.1 Deep Neural Network

Deep Neural Network(以下 DNN) は，Rosenblatt が [17] で提案した多層パーセプトロンを DeepLearning がベースとしているモデルのであり，人間の神経回路網を模した，学習能力を持つパターン識別器である．

Rosenblatt の多層パーセプトロンは 3 層構造からなり，それぞれ S(Sensory) 層，A(Association) 層，R(Response) 層と呼ばれる層にニューロンが有限個配置されている．結合については S → A → R の単方向のみであり，S → S といった層内の結合や，S → R といった層を通り越しての

結合は存在しない．結合には結合重み，あるいは単に重みと呼ばれる係数がかかっているが，S → A の結合重みは不変量とし，A → R の結合重みのみが学習によって係数を変化させる能力を持っている．DNN は A 層の層が多層になった場合と考えることができ，中間層から中間層への結合重み $A_1 \to A_2$ から $A_{n-1} \to A_n$ が追加されたネットワークだと仮定すれば，長いパーセプトロンとしてみることができる (図 1)．

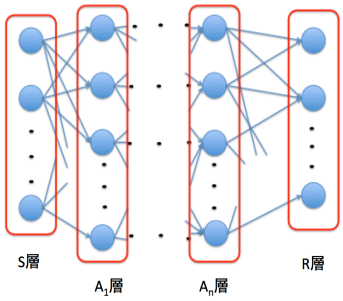

図 1 Deep Neural Network

DNN が行う計算について，A 層が 1 層のみの単純なモデルについて考えれば，第 n 層の j 番目のニューロンが第 $n-1$ 層のニューロンから受ける入力刺激を $i_j^{n-1\to n}$，第 n 層の j 番目のニューロンによる出力は o_j^n，第 n 層の j 番目のニューロンが持つ閾値を θ_j^n，第 $n-1$ 層の i 番目のニューロンと，第 n 層の j 番目のニューロンとの結合の強さは $w_{i\to j}^{n-1\to n}$，適当な関数を f とすると，A 層の j 番目のニューロンへの入力，A 層の j 番目のニューロンの出力は以下の 1, 2 式

$$i_j^{S\to A} = \sum_i (w_{i\to j}^{S\to A} o_i^S - \theta_j^A) \tag{1}$$

$$o_j^A = f(i_j^{S\to A}) \tag{2}$$

で与えられ，R 層の k 番目のニューロンへの入力，R 層の k 番目のニューロンの出力は以下の 3, 4 式

$$i_k^{A\to R} = \sum_j (w_{j\to k}^{A\to R} o_j^A - \theta_k^R) \tag{3}$$

$$o_k^R = f(i_k^{A\to R}) \tag{4}$$

で表される．個々のニューロンに関する 1, 2, 3, 4 式をモデル全体に拡張した式については，第 n 層内のニューロンが k 個存在するとき，その n 層のニューロン全体を表す $\mathbf{n}$ を

$$\mathbf{n} = (n_1, n_2, \cdots, n_k)^{\mathrm{T}} \tag{5}$$

で表し，第 $n-1$ 層から第 n 層への結合重みを行列形式で表現したものを $\mathbb{W}^{n-1\to n}$ とすると，

$$\mathbb{W}^{n-1\to n} = \begin{pmatrix} w_{1\to 1}^{n-1\to n} & \cdots & w_{1\to n}^{n-1\to n} \\ \vdots & \ddots & \vdots \\ w_{m\to 1}^{n-1\to n} & \cdots & w_{m\to n}^{n-1\to n} \end{pmatrix} \tag{6}$$

と書ける．各ニューロンの入力 i，出力 o，閾値 θ についても 5 式と同様の定義のしかたでベクトル形式 $\mathbf{i}$，$\mathbf{o}$，θ として与えれば，1, 2 式を層内全体に拡張した

$$\mathbf{i}^{S\to A} = \mathbb{W}^{S\to A}\mathbf{o}^S - \theta^A \tag{7}$$

$$\mathbf{o}^A = f(\mathbf{i}^{S\to A}) \tag{8}$$

3, 4 式を層内全体に拡張した

$$\mathbf{i}^{A\to R} = \mathbb{W}^{A\to R}\mathbf{o}^A - \theta^R \tag{9}$$

$$\mathbf{o}^R = f(\mathbf{i}^{A\to R}) \tag{10}$$

を得る．DNN は全体として S 層からの入力を A 層で変換する．A 層が多層の場合については A_1 層から A_2 層といった中間層から中間層への入出力が同様に追加され，最終的に 9 式の $\mathbf{o}^R$ によって出力する．理想的な出力を求めるためには，各層をつなげる結合重みを学習して調整する．f はよくシグモイド関数，Relu 関数が用いられる．

学習については，Rumelhart ら [18] で提案された Back-Propagation で行う．Back Propagation では，入力に対応する理想的な出力 (教師信号) を $\mathbf{y}$ として，結合重みの変化量 $d\mathbb{W}$ を，損失関数と呼ばれる以下の $\mathbf{E}$ の値によって決定する．

$$\mathbf{E} = \frac{1}{2}(\mathbf{y} - \mathbf{o}^R)^2 \tag{11}$$

第 $n-1$ 層の i 番目のニューロンから第 n 層の j 番目のニューロンに対する結合重みで $\mathbf{E}$ を微分すると

$$dw_{i\to j}^{k-1\to k} = -\frac{\partial \mathbf{E}}{\partial w_{i\to j}^{k-1\to k}} \tag{12}$$

が得られ，12 式の右辺を変形すると

$$\frac{\partial \mathbf{E}}{\partial w_{i\to j}^{k-1\to k}} = \frac{\partial \mathbf{E}}{\partial i_j^k}\frac{\partial i_j^k}{\partial w_{i\to j}^{k-1\to k}} = \frac{\partial \mathbf{E}}{\partial i_j^k} o^{k-1} \tag{13}$$

$$\begin{aligned} \frac{\partial \mathbf{E}}{\partial i_j^k} &= \sum_l \frac{\partial \mathbf{E}}{\partial i_l^{k+1}}\frac{\partial i_l^{k+1}}{\partial o_j^k}\frac{\partial o_j^k}{\partial i_j^k} \\ &= \sum_l \frac{\partial \mathbf{E}}{\partial i_l^{k+1}} w_{j\to l}^{k\to k+1} f'(i_j^k) \end{aligned} \tag{14}$$

が得られる．ここで，$\partial\mathbf{E}/\partial i_j^k = d_j^k$ とおくと，

$$\begin{aligned} dw_{i\to j}^{k-1\to k} &= -d_j^k o^{k-1} \\ d_j^m &= (o_j^m - y_j) f'(i_j^m) \\ d_j^k &= (\sum_l w_{j\to l}^{k\to k+1} d_l^{k+1}) f'(i_j^k) \end{aligned} \tag{15}$$

を満たす．15 式の計算の過程で，出力層での理想出力と実際の出力との誤差が，出力層から入力層の方向へ，計算時と逆の方向に $w_{i\to j}^{k\to k+1}$ で重みを付加しながら伝播されていく形をとることから，Back Propagation は誤差逆伝播法とも呼ばれる．

3.2 Autoencoder

本節では，次章で提案モデル内に取り入れた Autoencoder(以下 AE) について解説を行う．AE は自己符号化器とも呼ばれており，入力されたデータを出力で再現するように学習を行う．

AE によるネットワークの計算については，入力層と出力層を一致することが目的のため，入力層と出力層のちょうど間にある中間層を基準としてエンコード部分 (Encoder) と，デコード部分 (Decoder) に分けて計算を考える．通常，Encoder で入力層に入力されたデータの特徴量を抽出し，圧縮することができる．Decoder では圧縮された特徴量からデータを復元する計算を行う [19]．このとき，Encoder と Decoder それぞれは 7 と同様に計算を行い，結合重み $\mathbf{w}$ が特徴量を保存していることになる．学習についても，基本的に Back Propagation で行うことができる．そのため，11 の教師信号 $\mathbf{y}$ を入力データと同じものとして扱うことで，入力データを再現するような学習が可能である．

3.3 Convolutional Neural Network

本節では, Convolutional Neural Network(以下 CNN) について述べる．CNN は畳み込みニューラルネットワークと呼ばれており，画像を入力データに用いる時によく扱われる．

CNN は [20] の Neocognitron を画像データへの応用に特化させたネットワークである．S 細胞と C 細胞をベースとした中間層を入力層に近い部分で取り入れ，入力された画像内に存在する特徴量を抽出する構造を取り，出力層に近い中間層では，DNN と同様のニューロンを配置し，学習によって結合重みを調節する構造を取る (図 2)．

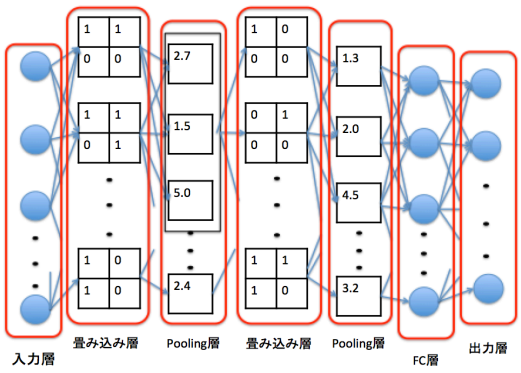

図 2 Convolutional Neural Network

CNN では，Neocognitron の S 細胞に相当する中間層は畳み込み層と呼ばれ，層内にはニューロンの代わりに行列型のフィルターが配置されている．7 式の $\mathbb{W}$ に相当する部分がフィルターであり，入力された信号はフィルターとの内積をとる．C 細胞に相当する中間層は Pooling 層と呼ばれ，畳み込み層でフィルターと掛け合わされた入力信号のいくつかをまとめあげるような構造を取っている．まとめあげる際には，平均を取る Average Pooling，最大値を取る Max Pooling 等があり，一般的には Max Pooling が用いられる．Max Pooling を行っている Pooling 層では特徴量の位置不変性を再現できる．CNN はこの畳み込み層と Pooling 層を何回か繰り返し，その後 Full-Connected 層 (FC 層) という通常のニューラルネットワークと同じ構造を持つ層へと値が受け渡される．つまり，CNN の計算アルゴリズムについては，畳み込み層と Pooling 層以外は，7 と同様の計算で出力層に向けて値が受け渡される．畳み込み層，Pooling 層はそれぞれ Back Propagation によって学習する．

3.4 Long Short-Term Memory

前節までで紹介した DeepLearning のネットワークは，入力されるデータが画像，ベクトル信号など，入力される順番によらず，入力されるデータの順番が変化することで結果が変わることはない．データの順番 (系列) が重要とされるデータは時系列データと呼ばれ，時系列データに対応させるネットワークを構築するためには，時系列データの性質を理解する必要がある．一般に機械学習における時系列データとは，現在のデータを $\mathbf{x}(t)$ としたとき，そのデータが 1 単位時間前のデータ $\mathbf{x}(t-1)$ や 2 単位時間前のデータ $\mathbf{x}(t-2)$ と関係を持っており，次の時刻のデータ $\mathbf{x}(t+1)$ を予測する問題を解く際に，現在のデータ，あるいはそれより前のデータの情報を利用して予測することのできる系列のデータを指す．

本節ではこうした時系列データに対応する DeepLearning の手法である Long Short-Term Memory(以下 LSTM) について解説する．

LSTM のネットワークは図 3 のように，中央にセルがあり，そのセルの入出力を制御する 3 つのゲートを持つ構造を取っている．中央のセルは，情報を記憶するニューロンであり，そのセルに入力される情報を制御する Input gate，前の時刻のセルの情報を結合重みをかけて伝える Forget gate，セルからの出力を制御する Output gate が配置されている．各 gate がセルの入出力を調整することによって，セルに適切な情報が記憶されるように学習することができる．

4. Deep-CNN-AE-FV-LSTM

本節では，前章で述べた DeepLearning のアルゴリズム

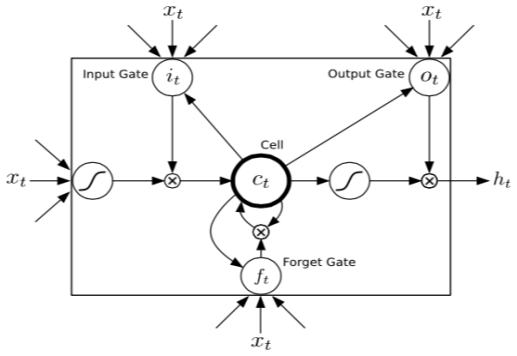

図 3 LSTM のモデル図．中央に配置されたセルへの入力，セルの出力を，ゲートが制御する．
※?より引用

をいくつか組み合わせて，中間層に抽出された特徴量を再利用するモデル，Deep-CNN-AE-FV-LSTM(D-CAFL) を提案する．

提案モデルでは，前章に述べた CNN 部，AE エンコーダ部，LSTM 部，AE デコーダ部が結合した構造になっている (図 4)．各部分が独立した役割を持っており，CNN 部では入力される画像を特徴量に分解する．AE エンコーダ部では CNN から受け取った特徴量を圧縮する．LSTM 部では AE エンコーダ部から受け取った入力を時系列として処理し，受け取った入力から，次の時刻のエンコーダの出力を予測し，その結果を AE デコーダ部へと出力する．LSTM では，時系列処理を精度よく行うための Flag-Vector(FV) が付加される．AE デコーダ部では，LSTM 部から得た出力から，画像を復元する．

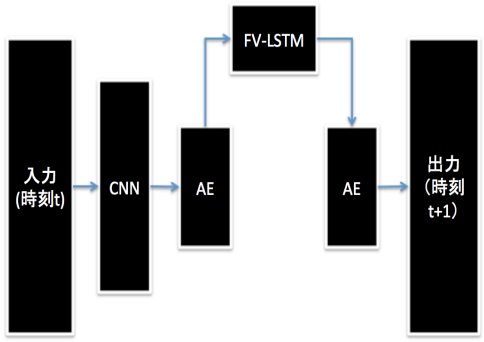

図 4 D-CAFL の全体図

提案モデルが行う計算は，各部分の独立した計算アルゴリズムを合わせたものであり，前章で述べた各計算アルゴリズムを基盤とし計算を行う．本節では，各部に配置されたネットワークが行う計算アルゴリズムについて述べる．

- CNN 部の計算アルゴリズム

  D-CAFL の CNN 部では，従来の CNN と違い，図 2 と比較して，FC 層が無く Pooling 層から直接出力する形をとる．畳み込み層 →Pooling 層の組み合わせを 2 層重ねた構造をしており，畳み込み層は 1 層目，2 層目それぞれ，フィルターの大きさが $5*5$，$3*3$ の大きさで設定されている．Pooling 層では 1 層目，2 層目それぞれ，畳み込み層でフィルターとの畳み込みを行った結果の 9 枚を，最大値の 1 枚にまとめる．各部の計算は図??, 図??に基づいて行われ，CNN 部での計算においては，一般的なアルゴリズムとの差はない．

- AE 部の計算アルゴリズム

  AE 部では，従来の AE と同じ構造，同じ計算アルゴリズムをとっており，エンコーダ部，デコーダ部ともに，前節のものと同じ構造のものを採用している．エンコーダ部で情報を圧縮するが，提案手法では入力される情報を 25 次元に圧縮し，LSTM 部に値を渡す．デコーダ部分では 25 次元の入力情報から，画像を復元する．

- LSTM 部の計算アルゴリズム

  LSTM 部では，計算アルゴリズムそのものは前節で述べたアルゴリズムと変わらないが，入力部分に Flag-Vector を付加して計算を行う．Flag-Vector は，入力される情報に合わせて設計する．本研究では，物体の回転画像を使用するため，物体の回転角に合わせてベクトルを 11 次元で用意する (図 5)．

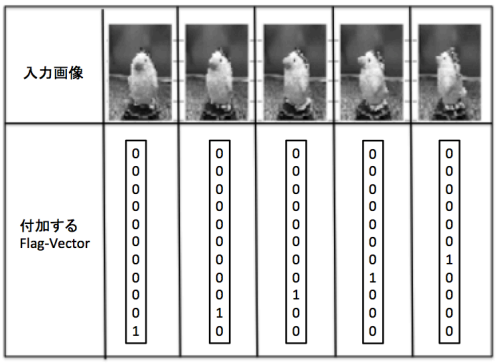

図 5 生成した Flag-Vector の例

D-CAFL の各部分は，Back Propagation を用いて学習を行う．ただし，一般的な DeepLearning の学習手順とは異なる．そのため，本節では行う学習の手順について述べる．まず，CNN 部と AE 部は用意した回転画像のデータで学習する前に，画像のデータセットである，Anime-Character-Dataset[*1] あるいは CIFAR-10[*2] を使って，事前学習を行う．特に CNN 部については，図 2 のようにネットワーク全体を用意し，事前学習用のデータセットによる分類問題を行った後，入力部に近い畳み込み層，Pooling 層を 2 層ずつコピーしてそれを CNN 部とする．AE 部については，事前学習用のデータセットを使って，情報を圧

[*1] http://www.nurs.or.jp/~nagadomi/animeface-character-dataset/
[*2] http://www.cs.toronto.edu/~kriz/cifar.html

縮し，復元する能力をエンコーダ部，デコーダ部それぞれに学習させる．

事前学習を終えたら，CNN 部と AE 部を連結して，回転画像のデータを用いて学習を行う．学習が終わったら，AE 部をエンコーダ部とデコーダ部に分けて，あいだに LSTM 部を挟む形で，再度回転画像を入力し，LSTM 部を学習する．LSTM 部の学習の際には，CNN 部と AE 部の結合重みは変化させない．

以上をまとめると，D-CAFL の全体の振る舞いとして，画像データが入力されたとき，まず CNN 部で特徴量で分解し，次に AE のエンコーダ部で分解された特徴量を圧縮し，その後 LSTM 部による圧縮された情報に対して時系列処理を行った後，最後に AE のデコーダ部によって画像を復元するというモデルが全体として形成される (図 6)．

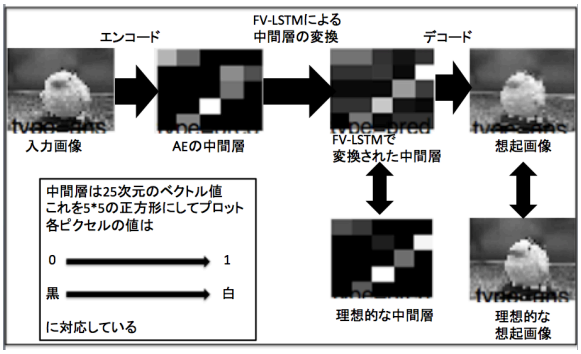

図 6 D-CAFL で行われる処理の全体図

5. 実験と考察

提案モデルの妥当性を示すため，2 つの物体の回転画像を用いて，物体の回転画像の想起実験を行う．この実験により，転移学習の効果について考察する．

5.1 実験設定

各実験に用いる物体は，回転させながら撮影したものであり，提案モデルの予測実験に用いるデータセットである．物体 A は 256*256 ピクセルの大きさの画像を，正面画像を 0 度とした位置から，回転角を 18 度とし，1 枚ごとに 18 度回転する画像を，20 枚用意した (図 7)．物体 B は，背面画像 (180 度) から正面画像 (360 度) まで回転角を 18 度とし，1 枚ごとに 18 度回転する画像を，11 枚用意した (図 8)．学習に用いる際には 50*50 の大きさに圧縮し，物体 A は学習に 20 枚全てを用いる一方で，物体 B は正面画像 (360 度) と背面画像 (180 度) の 2 枚のみを用いる．

図 7 回転画像データセットの物体 A

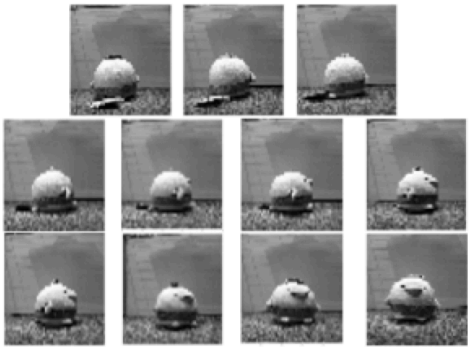

図 8 回転画像データセットの物体 B

物体 A に対して回転画像の想起実験を行う際の，実験手順について表 1 に示す．

次に，物体 B に対して回転画像の想起実験を行う際の，実験手順については，表 1 で学習を行ったモデルのうち，事前学習を行った 2 つのモデルを引き続き使用し，そのモデルに，物体 B の後ろ姿の画像と正面を向いた画像のみ学習させる．

5.2 実験結果

物体 A の回転画像について表 1 に従って学習した 3 つのモデルの実験結果を以下に示す．まず，事前学習を行わなかったモデルの実験結果を示す．

事前学習を行わずに物体 A のみを学習させると，物体の特徴量をうまく学習することができず，予測結果として出力された画像はぼやけた画像になっている．事前学習を行わなかったモデルは，少ない枚数では学習できていないことがわかった．

次に，Anime-Character-Dataset で事前学習したモデルで行った実験結果を示す．

表 1 物体 A の回転画像想起実験の手順

Step1：D-CAFL について，CIFAR-10 で事前学習したモデル，
Anime-Character-Dataset で事前学習したモデル，
事前学習しなかったモデルの 3 種類を準備する．

Step2：それぞれのモデルに対し，物体 A の画像を 20 枚全て
学習させる．学習方法は，入力するデータに対して，
左回りに 18 度回転した画像を教師データとし，
11 式の値を誤差逆伝播法で
ネットワーク全体に伝播し，学習する．

Step3：20 枚の画像を入力し，学習した時点で 1epoch とし，
1epoch 毎に 11 式をピクセル数で割った値を
確認する．その際，前回の epoch と比べて
平均値の変化量が 0.000001 以下を記録し，
それが 100epoch 連続した場合に，学習を終了させる．
収束しない場合は，20000epoch で終了とする．

Step4：学習が終了した時点で，それぞれのモデルに対し，
物体 A の画像各 1 枚につき想起される画像 1 枚を
モデルに出力させる．

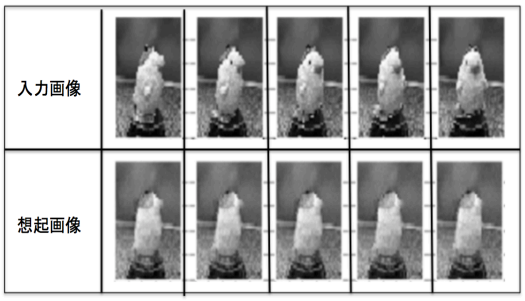

図 9 事前学習なしのモデルによる．物体 A の想起画像，288 度 360 度まで

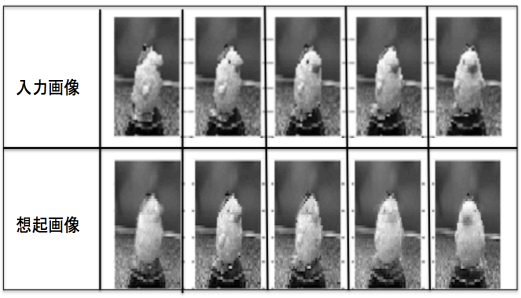

図 10 Anime-Character-Dataset で事前学習したモデルによる．物体 A の想起画像，288 度 360 度まで

Anime-Character-Dataset で事前学習を行ったモデルは，物体の特徴及び，回転という時系列の特徴を獲得しており，入力された画像に対して 18 度左回りに回転した画像の予測ができていることが確かめられた．

最後に，CIFAR-10 で事前学習したモデルによる実験結果を示す．

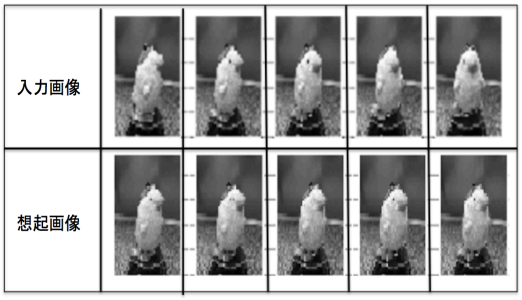

図 11 CIAR-10 で事前学習したモデルによる．物体 A の想起画像，288 度 360 度まで

CIFAR-10 を用いた事前学習を行ったモデルの実験結果は，Anime-Character-Dataset で事前学習したモデルに比べて，出力された画像が物体の形状を捉えている．しかし，回転という時系列要素を学習することがうまくできていない結果となった．

次に，物体 B の回転画像について学習した 2 つのモデルの実験結果について述べる．まず，Anime-Character-Dataset による事前学習を行ったモデルの実験結果を示す．

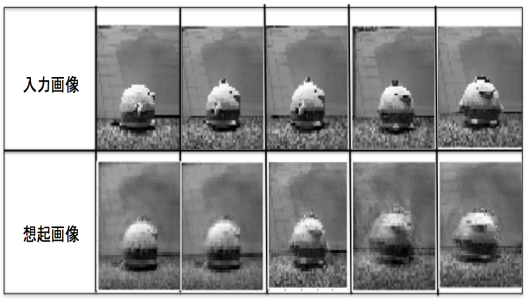

図 12 CIAR-10 で事前学習したモデルによる．物体 A の想起画像，288 度 360 度まで

実験の結果，後ろ姿と正面画像のみを学習に用いても，物体の回転画像が想起できることが確認された．物体 A の実験と比べて，想起された画像がぼやけているが，物体 B の目や口といった特徴を捉えながら，回転という時系列処理を行うことができた．

次に，CIFAR-10 による事前学習を行ったモデルの実験結果を示す．

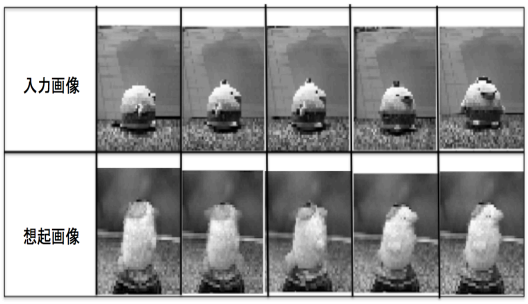

図 13 CIAR-10 で事前学習したモデルによる．物体 A の想起画像，288 度 360 度まで

実験の結果，物体 A の画像に近いものが出力されている．これは，物体 A の学習の段階で局所解に収束している．あるいは，物体 A の実験の時点で過学習しており，物体 B をうまく学習できなかったと考えられる．

6. おわりに

本研究では，DeepLearning における中間層を再利用することで，少ないデータ数で学習することができる転移学習モデルを提案した．提案モデルは，あらかじめ別のデータセットで学習することで，モデルの中間層に特徴量が学習された状態を事前に作る．事前学習を取り入れることにより，目的とする問題用のデータセットの量が少なくても学習を行うことができる．その一方で，事前学習に用いるデータセットによっては，学習がうまくいかない結果も示された．これは，事前学習に使用したデータセットと，回転予測に使用したデータセットの特徴量に違いがあったためであると考えられる．今後の課題として，実験に用いるデータの種類を増やし，より多くの問題に対して，提案モデルを適用することが挙げられる．

参考文献

[1] Marc'aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg Corrado, Jeff Dean, Quoc V. Le, Andrew Y. Ng, "Building High-level Features Using Large Scale Unsupervised Learning", Proceedings of the 29th International Conference on Machine Learning, pp.81–88, 2012.

[2] Lei Jimmy Ba, Rich Caruana, "Do Deep Nets Really Need to be Deep?", Neural Information Processing Systems Conference, 2014.

[3] Yoshua Bengio, Pascal Lamblin, Dan Popovici, Hugo Larochelle, "Greedy Layer-Wise Training of Deep Networks",Advances in Neural Information Processing Systems 19, pp.153–160, 2007.

[4] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, Samy Bengio, "Why Does Unsupervised Pre-training Help Deep Learning?", Journal of Machine Learning Research, pp.625 － 660, 2010.

[5] Clement Farabet, Camille Couprie, Laurent Najman and Yann LeCun, "Learning Hierarchical Features for Scene Labeling ", IEEE Transactions on Pattern Analysis and Machine Intelligence, in press, 2013.

[6] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", Advances in Neural Information Processing Systems 25, pp.1106–1114, 2012.

[7] Olga Russakovsky, Jia Deng*, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge", International Journal of Computer Vision, pp.211-252, 2015.

[8] Oriol Vinyals, Quoc Le, "A Neural Conversational Model", ICML Deep Learning Workshop, 2015.

[9] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, Martin Riedmiller, "Playing Atari with Deep Reinforcement Learning", NIPS Deep Learning Workshop, 2013.

[10] Douwe Kiela, Leon Bottou,"Learning Image Embeddings using Convolutional Neural Networks for Improved Multi-Modal Semantics", Empirical Methods on Natural Language Processing, 2014.

[11] Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, Trevor Darrell, "Long-term Recurrent Convolutional Networks for Visual Recognition and Description",arXiv preprint arXiv:1411.4389, 2014.

[12] Matthew D Zeiler, Rob Fergus, "Visualizing and Understanding Convolutional Networks", In Computer Vision-ECCV 2014, pp. 818-833, 2014.

[13] Koray Kavukcuoglu, Pierre Sermanet, Y-lan Boureau, Karol Gregor, Michael Mathieu and Yann Lecun, "Learning Convolutional Feature Hierarchies for Visual Recognition", Advances in Neural Information Processing Systems 23, pp.1090–1098, 2010.

[14] Jason Yosinski, Jeff Clune, Yoshua Bengio, Hod Lipson, "How transferable are features in deep neural networks?", Advances in Neural Information Processing Systems 27, pp.3320–3328, 2014.

[15] 野田 邦昭, 有江 浩明, 菅 佑樹, 尾形 哲也,"Deep neural network を用いたヒューマノイドロボット による物体操作行動の記憶学習と行動生成", The 27th Annual Conference of the Japanese Society for Artificial Intelligence,2013

[16] Ryan Kiros,Richard S. Zemel,Ruslan Salakhutdinov,"Multimodal Neural Language Models", Proceedings of The 31st International Conference on Machine Learning, pp.595-603,2014

[17] F Rosenblatt,"The perceptron: A probabilistic model for information storage and organization in the brain.",Psychological Review, Vol 65,pp 386-408,1958.

[18] Rumelhart D, Hinton G,Williams R, "Learning representations by back-propagating errors.",Nature, Vol 323-9,pp 533-536,1986.

[19] Pascal Vincent ,Hugo Larochelle,Isabelle Lajoie,Yoshua Bengio,Pierre-Antoine Manzagol, , "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion", The Journal of Machine Learning Research, pp.3371-3408 vol 11, 2010.

[20] Kunihiko Fukushima, Sei Miyake, "NEOCOGNITRON: A NEW ALGORITHM FOR PATTERN RECOGNITION TOLERANT OF DEFORMATIONS AND SHIFTS IN POSITION", Pattern Recognition Vol 15, No 6, pp 455-469, 1982

[21] Wojciech Zaremba, Ilya Sutskever, Oriol Vinyals, "Recurrent Neural Network Regularization", arXiv:1409.2329, 2014.

[22] Gers FA, Schmidhuber J, Cummins F, "Learning to forget: continual prediction with LSTM", Artificial Neural Networks,, pp.850 - 855 vol.2, 1999.