階層フィッシャー重みマップを用いた識別的初期化による深層畳み込みニューラルネットワーク構築法

中山 英樹 †
† 東京大学 大学院情報理工学系研究科
E-mail: nakayama@ci.i.u-tokyo.ac.jp

Abstract

深層学習は近年人工知能に関わる多くの分野で驚異的な性能を示しており，画像認識分野においては畳み込みニューラルネットワーク (CNN) が従来の特徴量ベースの手法を大きく上回る識別性能を達成したことから特に注目を集めている．しかしながら，CNN は学習に必要な計算コストが非常に大きいことや，ネットワーク構造の設計や最適化における多くのハイパーパラメータが存在することなどから，容易に使える技術としては確立していないのが現状である．

本研究では，フィッシャー重みマップ (FWM) を用い，CNN の畳み込み層を識別的かつ解析的に学習する手法を提案する．本手法は単純な固有値問題に帰着するため，比較的少サンプルで高次元の入力から安定的に単層ネットワークの学習を行うことができる．更に，本手法による畳み込み層をプーリング層と合わせて階層的に積み上げることで，深層構造を効率よく構築できる．性能比較実験において，提案手法は最新の深層学習手法に匹敵する良好な識別精度を得た．

1 はじめに

計算機の著しい進歩やデータ量の指数的な増加を背景に，多層ニューラルネットワークを用いた深層学習 (deep learning) が再び注目を集めている [12, 17]．特に画像認識の研究分野においては，畳み込みニューラルネットワーク (convolutional neural networks, CNN) [25, 30, 19] が多くのベンチマークやコンペティションにおいて大差で従来の手法を上回っている [5, 21, 23]．CNN の構造は視覚野の生物学的な分析に基づいており [18]，層の間の結合を局所領域（受容野）のみに限り，結合の重みパラメータを全受容野で共有する．この結果，CNN は単純な全結合のネットワークと比較して大幅にパラメータ数を減らしており，認識において本質的に重要な構造を学習できると考えられている．

しかしながら，CNN には依然として多くの課題が存在する．まず，高性能なネットワークの学習に要する計算コストは非常に大きく，多くの場合 GPU やクラスター計算機等を用いた特殊な実装が必要となる [5, 21]．また，CNN は自由度の大きいモデルであるため過学習を起こしやすい．過学習を起こさずに，CNN の学習を行うためには多数の教師付サンプルを用いるか，教師なし事前学習 [11] を行うなどの工夫が必要となる．さらに，ドロップアウト [16] 等の経験的な正則化手法や，最適化における数多くのパラメータの設定・スケジューリングなど多くのノウハウを必要とする [2]．このため，CNN を用いた深層学習は高い認識精度を達成しうる反面，一般的に使いやすいツールとしては未だ確立していないのが現状といえる．

本研究では，多変量解析による単純な次元圧縮手法を用いた，畳み込み層の解析的な初期化方法について検討を行う．特に，フィッシャー重みマップ (FWM) を用いた識別的な初期化が有効であることを示す．FWM の適用により，畳み込み後の画像表現（特徴マップ）のクラス間分離（判別規準）を最大化する射影を陽に導出することができる．FWM は一般化固有値問題として解析的に解けるため，中間層として取り出したいニューロンと同数の上位固有ベクトルを使用することで簡便に畳み込み層を構築できる．提案手法は一層ごとに単純な固有値を問題を解くため，計算コストが比較的小さく，少数の教師付サンプルからも安定的に学習が可能である．更に，本手法による畳み込み層を適切な活性化関数やプーリング層と合わせて階層的に積み上げることで，強力な深層構造を効率よく構築できる．

いくつかのデータセットを用いた実験により，提案手法の有効性を示す．また，網羅的に提案手法の挙動を調査し，性能へ与える影響に関する考察を行う．

2 フィッシャー重みマップ

フィッシャー重みマップ (FWM) [29] は，画像中の画素（または局所領域）へ与える適切な重みを求めるための方法として考案され，固有顔法 [33] やフィッシャー顔法 [1] の考え方に起源を有する．固有顔法やフィッシャー顔法では単純に画素ベクトルに対し主成分分析またはフィッシャー判別分析を適用するのに対し，FWM は入力が行列である場合へ拡張され，各画素が複数の特徴

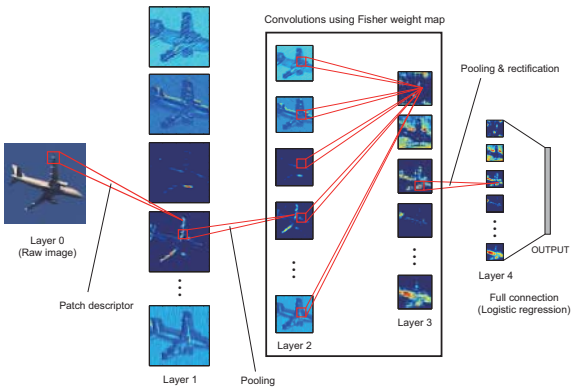

図 1　提案手法により構築されるネットワークの全体図（畳み込み層が一つの場合）．

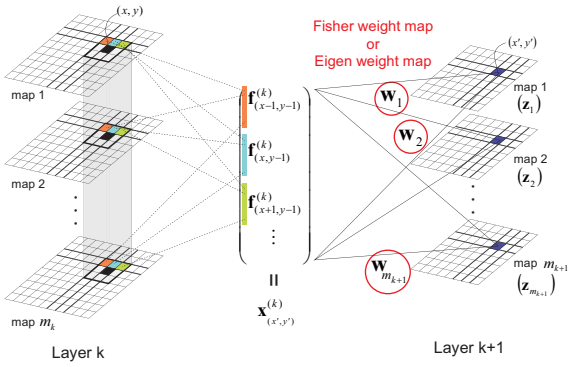

図 2　畳み込み層の詳細図．

値を有する場合に対応する．すなわち，各画素の特徴ベクトルの重み付き和として得られる大域的特徴ベクトルのフィッシャー判別規準を最大化するように重みパラメータを決定する．この考え方は原田ら [15] により一般物体認識の問題において検証および拡張がなされており，spatial pyramids [22] の各領域に対する適切な重みを導出するために利用されている．この結果，最終的な特徴ベクトルの次元数は大幅に削減されているにも関わらず，最新の pyramid matching の手法と遜色のない識別精度が得られることが報告されている．

本研究では，前述の先行研究とは逆方向に FWM を適用し，局所特徴の畳み込みへ利用する．すなわち，近傍を含めた局所特徴の全ての要素の重み付け和により得られる，特徴マップベクトルのクラス間分離を最大とする射影を求める．このように，FWM を特徴量の畳み込みへ利用する考え方は本研究における新規な着想である．

FWM の数理的な核はフィッシャー判別分析の考え方であり，PCA に基づき定式化された固有重みマップ法 (EWM) [29] の識別的な問題における自然な拡張となっている．本研究では CNN の学習において，EWM を教師なし初期化，FWM を教師付の識別的な初期化と位置づけ，比較検証する．

3　提案するネットワーク構築手法

図 1 に示すフィードフォワード型の多層ネットワークを考える． 第 k 層 (L_k) において，各サンプルは m_k 個の特徴マップ[1]で構成され，各特徴マップの大きさは $P_k \times P_k$ であるとする．また，特徴マップ上の各要素はニューロンと呼ぶことにする．例えば，カラー画像 (L_0) は RGB の各チャンネルに対応する 3 つの特徴マップからなり．各画素がニューロンに相当する．

ネットワークは複数の畳み込み層を主な構成要素とし，プーリング層を交えながら階層的に積み上げることで深層構造を構成する．また，畳み込み層の出力に適切な活性化関数を適用することも重要な要素となる．最後に，最終層の出力全てを説明変数として用いロジスティック回帰による識別器を構築する．これは，最終層のみ全結合の単層パーセプトロンを学習することに相当し，深層学習の研究分野においては教師なし事前学習により得られるネットワークを識別タスクへ利用するための簡便な方法としてしばしば用いられる [17]．

また，最終層のみならず，中間の畳み込み層の出力も識別器の構築の際に利用することで，より識別精度を向上させることが可能である．

3.1　重みマップによる畳み込み

図 2 に畳み込み層を取り出したものを示す．$\boldsymbol{f}^{(k)}_{(x,y)} \in R^{m_k}$ を第 L_k 層の座標 (x,y) における特徴ベクトルとする．つまり，$\boldsymbol{f}^{(k)}_{(x,y)}$ の各要素は各特徴マップにおけるニューロンの出力値をとる．畳み込みフィルタのサイズ，すなわち受容野の大きさを $n \times n$ とし，受容野内の特徴ベクトル $\boldsymbol{f}^{(k)}$ を列挙したベクトルを $\boldsymbol{x}^{(k)}_{(x',y')} \in R^{m_k \times n^2}$ とする．ここで，(x',y') は畳み込み後の特徴マップにおける新しい座標とする．この操作を全受容野に対して行うと $(P_k - n + 1) \times (P_k - n + 1)$ 個のベクトル $\boldsymbol{x}^{(k)}_{(x',y')}$ が得られる[2]．これを座標順に並べ，行列としたものを次のように表記する．

$$\boldsymbol{X} = \left(\boldsymbol{x}^{(k)}_{(1,1)} \; \boldsymbol{x}^{(k)}_{(2,1)} \; \cdots \; \boldsymbol{x}^{(k)}_{(P_k-n+1,P_k-n+1)} \right). \qquad (1)$$

畳み込みの目的は，受容野内の全ての特徴の線形結合により，局所構造を埋め込んだ新しい特徴マップ $\boldsymbol{z}$ を与える射影 $\boldsymbol{z} = (\boldsymbol{X} - \bar{\boldsymbol{X}})^T \boldsymbol{w}$ を得ることである（$\bar{\boldsymbol{X}}$ は $\boldsymbol{X}$ の訓練サンプルにおける全平均である）．EWM，FWM はともに固有値問題として定式化され，畳み込みの射影は上位 m_{k+1} の固有ベクトルを用いることで構成できる．以下，EWM と FWM の詳細をそれぞれ説明する．

[1] ここで定義する“ 特徴マップ ”は，FWM 等の“ 重みマップ ”とは用語的に無関係であることに注意されたい．

[2] 大きさが減少することが望ましくない場合は，畳み込み前の層の周囲をゼロで埋めることにより回避できる

Eigen weight map (EWM)

EWM は $\boldsymbol{z}$ の分散 $J_E(\boldsymbol{w})$ を最大化する規準のもとで学習を行う．

$$\begin{aligned} J_E(\boldsymbol{w}) &= \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{z}_i-\bar{\boldsymbol{z}})^T(\boldsymbol{z}_i-\bar{\boldsymbol{z}}) \\ &= \boldsymbol{w}^T\left\{\frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{X}_i-\bar{\boldsymbol{X}})(\boldsymbol{X}_i-\bar{\boldsymbol{X}})^T\right\}\boldsymbol{w} \\ &= \boldsymbol{w}^T\Sigma_X\boldsymbol{w}, \end{aligned} \tag{2}$$

ここで，N は訓練サンプルの総数であり，$\bar{\boldsymbol{z}}$ は $\boldsymbol{z}_i$ の全平均である．$J_E(\boldsymbol{w})$ を最大化する射影ベクトルは以下の固有値問題の解として得られる．

$$\Sigma_X\boldsymbol{w}=\lambda\boldsymbol{w}. \tag{3}$$

このように，EWM は PCA を二次元（ベクトルベース）へ拡張した手法であるとみなすことができる．

Fisher weight map (FWM)

FWM は $\boldsymbol{z}$ のクラス間分離を最大とする識別的な射影を学習する．EWM は教師なし学習手法であるのに対し，FWM は教師付学習によりフィッシャーの判別規準 $J_F(\boldsymbol{w})$ の最大化を行う．したがって，識別において EWM よりも有効な畳み込み構造を学習できると期待される．$\tilde{\Sigma}_W$ と $\tilde{\Sigma}_B$ をそれぞれ $\boldsymbol{z}$ のクラス内分散，クラス外分散とする．すなわち，

$$\tilde{\Sigma}_W = \frac{1}{N}\sum_{j=1}^{C}\sum_{i=1}^{N_j}(\boldsymbol{z}_i^{(j)}-\bar{\boldsymbol{z}}^{(j)})(\boldsymbol{z}_i^{(j)}-\bar{\boldsymbol{z}}^{(j)})^T, \tag{4}$$

$$\tilde{\Sigma}_B = \frac{1}{N}\sum_{j=1}^{C}N_j(\bar{\boldsymbol{z}}^{(j)}-\bar{\boldsymbol{z}})(\bar{\boldsymbol{z}}^{(j)}-\bar{\boldsymbol{z}})^T, \tag{5}$$

ただし，C はクラス数，N_j はクラス j の訓練サンプル数，$\boldsymbol{z}_i^{(j)}$ はクラス j の i 番目の訓練サンプルであり，$\bar{\boldsymbol{z}}^{(j)}$ はその平均である．$\tilde{\Sigma}_W$ および $\tilde{\Sigma}_B$ のトレースは以下のようになる．

$$\begin{aligned} tr\tilde{\Sigma}_W &= \frac{1}{N}\sum_{j=1}^{C}\sum_{i=1}^{N_j}(\boldsymbol{z}_i^{(j)}-\bar{\boldsymbol{z}}^{(j)})^T(\boldsymbol{z}_i^{(j)}-\bar{\boldsymbol{z}}^{(j)}) \\ &= \boldsymbol{w}^T\left\{\frac{1}{N}\sum_{j=1}^{C}\sum_{i=1}^{N_j}(\boldsymbol{X}_i^{(j)}-\bar{\boldsymbol{X}}^{(j)})(\boldsymbol{X}_i^{(j)}-\bar{\boldsymbol{X}}^{(j)})^T\right\}\boldsymbol{w} \\ &= \boldsymbol{w}^T\Sigma_W\boldsymbol{w}. \end{aligned} \tag{6}$$

$$\begin{aligned} tr\tilde{\Sigma}_B &= \frac{1}{N}\sum_{j=1}^{C}N_j(\bar{\boldsymbol{z}}^{(j)}-\bar{\boldsymbol{z}})^T(\bar{\boldsymbol{z}}^{(j)}-\bar{\boldsymbol{z}}) \\ &= \boldsymbol{w}^T\left\{\frac{1}{N}\sum_{j=1}^{C}N_j(\bar{\boldsymbol{X}}^{(j)}-\bar{\boldsymbol{X}})(\bar{\boldsymbol{X}}^{(j)}-\bar{\boldsymbol{X}})^T\right\}\boldsymbol{w} \\ &= \boldsymbol{w}^T\Sigma_B\boldsymbol{w}. \end{aligned} \tag{7}$$

従って，フィッシャーの判別規準は以下のように定義される．

$$J_F(\boldsymbol{w})=\frac{tr\tilde{\Sigma}_B}{tr\tilde{\Sigma}_W}=\frac{\boldsymbol{w}^T\Sigma_B\boldsymbol{w}}{\boldsymbol{w}^T\Sigma_W\boldsymbol{w}}. \tag{8}$$

これを最大化する射影ベクトルは以下の一般化固有値問題の解として得られる．

$$\Sigma_B\boldsymbol{w}=\lambda\Sigma_W\boldsymbol{w}. \tag{9}$$

3.2 活性化関数

畳み込みそのものは単純な線形射影であるため，その出力に適切な活性化関数を適用し非線形性を与えることが非常に重要である．古典的なニューラルネットワークでは，tanh 関数やシグモイド関数などがしばしば用いられてきた [25, 19]．近年の研究においては，Rectified Linear Units (ReLU) が多層ニューラルネットワークの学習において極めて有効であるが判明し，広く用いられている [27, 21]．これは，入力 x に対し $R(x)=\max(0,x)$ をとる関数である．バイアス項と合わせて学習することでシグモイド関数と定性的に近い効果が得られるが，入力値の大きさに関わらず勾配が減少しないことなどから，非常に早く学習を収束させることが示されている．我々の提案手法においては勾配法による学習は必要ないが，ReLU を活性化関数として適用することで識別精度が大きく向上できることを示す．

なお，提案手法では畳み込み層のバイアス項が存在しないため，単純に ReLU を用いると負の反応を全て捨ててしまうことになる．そこで，Coates ら [8] の手法を参考に，以下のように負の反応も同様に抽出する二次元の活性化関数も試行する．

$$R_2(x)=\begin{pmatrix}\max(0,x)\\ \max(0,-x)\end{pmatrix}. \tag{10}$$

3.3 プーリング層

プーリングは局所領域内のニューロンの反応の統計量を抽出し，要約する操作である．段階的に平行移動不変性を与えつつ大域的なスケールへ情報を集約する重要な構造をネットワークに与えるものであり，画像認識においては畳み込み層同様に識別精度へ大きな影響を与える．初期の研究では単純な sub-sampling による方法がとられていた [30] が，近年では密に統計量を抽出することが多い [6, 5, 34]．本研究では，平均値プーリング (average pooling)，最大値プーリング (max pooling)，L2 プーリングを比較検討する [4]．また，プーリング後の特徴ベクトルに L2 ノルムの正規化を行う．

3.4 入力層からの特徴抽出

入力層は層数が少ないため，固有値問題で学習する場合得られる畳み込み層の数が少なくなる．例えば，5×5 の大きさの畳み込みフィルタをカラー画像から学習する場合，原理的に最大で $5\times5\times3=75$ の畳み込み層

しか求めることができず，表現能力が不足することが懸念される．このため，以下の二つの特徴記述子を入力層に用い，データセットに応じて使い分ける．これらはいずれも unsupervised な畳み込み層の学習に相当するものと解釈できる．

ランダムフィルタ: 重みをランダムに-0.05 から 0.05 の範囲で与えたフィルタとする．非常にシンプルであるにも関わらず，適切な多層構造とともに利用すると良好な識別精度を得ることが先行研究により示されている [19]．また，CNN 全体の初期値としてもよく用いられる．本研究では，後述する MNIST データセット [24] においてこれを用いる．

K-means: Coates ら [6, 9] によって提案された特徴記述子であり，生画像パッチベースの bag-of-words [10] と解釈できる．まず前処理として，局所領域ごとにコントラスト正規化を行った後，zero component analysis (ZCA) による白色化を行う．その後，K-means 法により visual words を生成し，triangular encoding により特徴ベクトルを生成する．この手法は，K-means で得られる基底ベクトルによる畳み込みに非線形な活性化関数を適用していると解釈できる．

3.5 表記方法

以上に述べたネットワークの各構成要素を以下のように表記する．

- $Rand(n, d)$: $n \times n$ サイズの画像パッチを入力とする d 次元のランダムフィルタ．
- $K_m(n, d)$: $n \times n$ サイズの画像パッチを入力とする d 次元の K-means (bag-of-words) によるフィルタ．
- $C(n, m)$: $n \times n$ サイズの受容野を入力とし，m 層からなる畳み込み層．C_{EWM}, C_{FWM} のように，畳み込みに用いる次元圧縮手法を添え字で表記する．添え字の表記がない場合は FWM を用いた畳み込み層とする．
- R, R_2: Rectified linear units による活性化関数．
- $AP[MP, L_2P](n, s)$: 平均値プーリング [最大値プーリング，L2 プーリング] を，s ピクセル（ニューロン）ずつ離しながら $n \times n$ の各局所領域に適用したプーリング層．
- $AP[MP, L_2P]_p$: 平均値プーリング [最大値プーリング，L2 プーリング] を，元の入力全体における $p \times p$ の p^2 個の領域に対応するように適用したプーリング層（最終層に用いる）．

例えば，$Rand(5, 200)$-R-$AP(4, 4)$-$C(3, 100)$-R-AP_2 という表記は，(1) 200 次元のランダムフィルタ，(2) ReLU による活性化関数，(3) 4×4 サイズの局所領域に対する平均値プーリング層，(4) 3×3 サイズの FWM による 100 層からなる畳み込み層，(5) ReLU による活性化関数，(6) 2×2 の大域的領域に対応する平均値プーリング層，の順に構成されるネットワークを示す．

図 3　STL-10 [6]，CIFAR-10/100 [20]，MNIST [24] の各データセットの画像例．

4 評価実験

深層学習の分野におけるいくつかの標準的なデータセットを用い，提案手法の挙動を網羅的に調査すると共に，最新の手法との比較を行う．実験は，12 コアの CPU (Xeon 2.7GHz) を持つデスクトップ PC を用いて実施した．GPGPU 等の特殊なハードウェアは利用していない．

4.1 データセット

本稿では，STL-10 [6]，CIFAR-10/100 [20], MNIST [24] データセットを用いる（図 3）．

STL-10 は 96×96 ピクセルのカラー画像からなるデータセットで，10 クラスから構成される．元々教師なし事前学習を前提とした最適化の研究向けに作成されたデータセットであり，10 万個の教師なしサンプルが提供されているが，教師付サンプルは各クラス 100 サンプルしか存在しない．提案手法は教師付サンプルのみを用いて識別的学習を直接行うことが可能であるため，本稿では Gens らの研究 [13] と同様に教師なしサンプルは使用しない．全 10 試行の訓練サンプルセットがデータセット提供者から指定されており，その平均によって評価を行う．CIFAR-10/100 は，Tiny images [32] の画像を用いて構築されたデータセットであり，それぞれ 10 クラス，100 クラスの物体カテゴリからなる．

画像サイズは 32×32 ピクセルと比較的小さいが，訓練サンプル数は CIFAR-10 が各クラス 5000 サンプル，CIFAR-100 が各クラス 500 サンプルと比較的豊富であり，画像認識の深層学習における最も標準的なデータセットとなっている．MNIST は 28×28 ピクセルのグレー画像からなる手書き文字のデータセットであり，数字の 0 から 9 の 10 クラスからなる．各クラスあたり約 6000 個の訓練サンプルと 1000 個のテストサンプルからなり，古くからニューラルネットワークの研究に用いられてきたデータセットである．

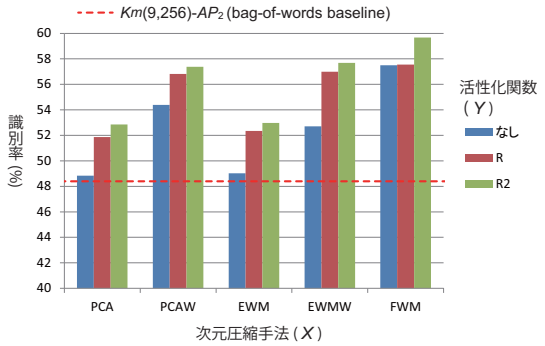

図 4　次元圧縮手法 X と活性化関数 Y の影響 (STL-10)．比較に用いるネットワークは $K_m(9,256)$-$AP(4,2)$-$C_X(3,256)$-Y-AP_2 とする．

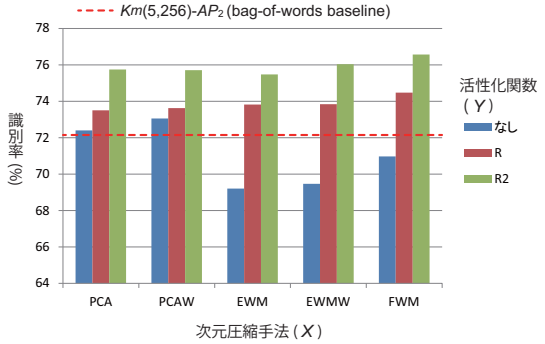

図 5　次元圧縮手法 X と活性化関数 Y の影響 (CIFAR-10)．比較に用いるネットワークは $K_m(5,256)$-$AP(3,2)$-$C_X(3,512)$-Y-AP_2 とする．

4.2 構成要素の検証

畳み込み層を一層のみに留め，畳み込みに用いる次元圧縮手法，プーリング手法，活性化関数の各構成要素が性能に与える影響を網羅的に調査する．様々な構成のネットワークにおいて，最終層のプーリングを 2×2 の領域にそろえる (AP_2, MP_2) ことで，大域的な空間情報を統一した公平な比較を行う．

まず，プーリング手法を平均値プーリング (AP) に固定し，畳み込み層に用いる次元圧縮手法と活性化関数の影響について，図 4,5 に STL-10, CIFAR-10 における比較結果をそれぞれまとめる．EWM，FWM に加え，PCA による畳み込みについても調査を行った．また，PCA と EWM については次元圧縮後の分散の正規化を行った場合についても調べ，それぞれ PCAW, EWMW と表記している．$K_m(n,256)$-AP_2 が提案手法による畳み込み学習なしのベースラインとなる．いずれの畳み込み手法においても，一定の性能向上が見られることが分かる．PCA と EWM には有意な差は見られず，FWM が最もよい性能を得ることが分かった．また，STL-10 においては，PCAW，EWMW が PCA，EWM をそれぞれ大きく上回っており，畳み込み後の特徴の正規化が重要である可能性が示唆された．FWM はこのような性質をもともと有している点でも適切な手法であるといえる．さらに，いずれの次元圧縮手法においても ReLU による活性化関数の適用により大きく性能が向上し，R_2 が常に最も良い結果を与えた．

次に，畳み込み層を FWM に固定し，プーリング手法を変更した場合の挙動を表 1,2 にまとめる．入力となる K-means 特徴に関して，STL-10 では MP，CIFAR-10 では AP がよい結果を与えることが分かる．提案手法による畳み込み後は，いずれの活性化関数を用いた場合でも，AP が最もよい結果を示した．

また，CIFAR-10 において畳み込みサイズを変更した場合の挙動を表 3 に示す．括弧の中の数字は，各場合における受容野内のニューロン数（特徴次元数）を示す．畳み込みサイズは識別性能に影響するが，サイ

表 1　プーリング手法の識別精度 (%) への影響 (STL-10).

Architecture	Acc.
$K_m(9,256)$-AP_2	48.4
$K_m(9,256)$-MP_2	**54.1**
$K_m(9,256)$-L_2P_2	51.4
$K_m(9,256)$-$AP(4,2)$-$C(3,256)$-AP_2	56.4
$K_m(9,256)$-$MP(4,2)$-$C(3,256)$-L_2P_2	50.9
$K_m(9,256)$-$MP(4,2)$-$C(3,256)$-AP_2	58.4
$K_m(9,256)$-$MP(4,2)$-$C(3,256)$-MP_2	44.6
$K_m(9,256)$-$MP(4,2)$-$C(3,256)$-R-L_2P_2	59.6
$K_m(9,256)$-$MP(4,2)$-$C(3,256)$-R-AP_2	60.0
$K_m(9,256)$-$MP(4,2)$-$C(3,256)$-R_2-L_2P_2	61.0
$K_m(9,256)$-$MP(4,2)$-$C(3,256)$-R_2-AP_2	**61.2**

表 2　プーリング手法の識別精度 (%) への影響 (CIFAR-10).

Architecture	Acc.
$K_m(5,256)$-AP_2	**72.2**
$K_m(5,256)$-MP_2	68.3
$K_m(5,256)$-L_2P_2	71.9
$K_m(5,256)$-$MP(3,2)$-$C(3,512)$-AP_2	70.4
$K_m(5,256)$-$AP(3,2)$-$C(3,512)$-L_2P_2	64.3
$K_m(5,256)$-$AP(3,2)$-$C(3,512)$-AP_2	71.0
$K_m(5,256)$-$AP(3,2)$-$C(3,512)$-MP_2	66.6
$K_m(5,256)$-$AP(3,2)$-$C(3,512)$-R-L_2P_2	73.8
$K_m(5,256)$-$AP(3,2)$-$C(3,512)$-R-AP_2	74.5
$K_m(5,256)$-$AP(3,2)$-$C(3,512)$-R_2-L_2P_2	76.3
$K_m(5,256)$-$AP(3,2)$-$C(3,512)$-R_2-AP_2	**76.6**

表 3　畳み込みサイズ n と入力の次元数 d（マップ数）の識別精度 (%) に対する影響 (CIFAR-10).

$K_m(5,d)$-$AP(3,2)$-$C(n,512)$-R_2-AP_2			
$n \diagdown d$	256	512	1024
3	76.6 (2304)	77.1 (4608)	78.1 (9216)
4	77.3 (4096)	78.3 (8192)	-
5	77.2 (6400)	77.8 (12800)	-

ズが小さい場合でも入力の次元数を大きくすれば性能を向上させることが可能であり，受容野内の特徴数が重要な要素であることが分かる．

4.3 多層ネットワークの構築

4.2 節の結果に基づき，FWM を用いて多層ネットワークの構築を行う．特に，一回目の畳み込み層に適用する活性化関数が二回目の畳み込みに与える影響について重点的に調べた．表 4,5 に結果を示す．興味深いことに，最終層の出力に適用した場合と異なり，中間の畳み込み層の出力には ReLU を適用しても大きな影響は確認されなかった．この結果を受け，複数の畳み込み層を積み重ねる場合は各層の出力に単純な平均値プーリングを適用し，最後の畳み込み層の出力にのみプーリング前に ReLU を適用することにする．

この結果，基本的に畳み込み層を重ねることで識別精度がより向上することが示された．CIFAR-10 においては三回目の畳み込み層を加えると逆に精度が落ちているが，最終層の出力のみならず各畳み込み層の出力に ReLU を適用し全てを識別器の構築に用いることで，全体としてさらに精度が向上する結果となった．これは，各層において異なる粒度の識別的情報を捉えているためと考えられる．

4.4 先行研究との性能比較

表 6 に，各データセットにおける先行研究と提案手法の識別精度の比較を示す．近年の研究では，反転画像を加える等の擬似的な訓練データ拡張を行うことで大きく性能を向上させているが，これは本研究の関心の範囲外である．ここでは，先行研究のスコアのうち，データセット拡張を行っていない場合で最もよいものをまとめた．また，提案手法のうち，末尾にアスタリスク (*) がついているものは，4.3 の方法で全ての畳み込み層の出力を利用した場合を示す．

提案手法は，STL-10，MNIST において先行研究を上回る識別精度を達成した．また，CIFAR-100 においても，最新の CNN に匹敵する良好な結果を得た．

CIFAR-10, CIFAR-100 は似た性質を持つデータセットであるが，提案手法は相対的に見て CIFAR-100 においてより良好な結果を示している言える．例えば，提案手法は CIFAR-100 においては Maxout [14] や Stochastic pooling [34] を用いた CNN を上回るが，CIFAR-10 では及ばない結果となっている．この理由として，CIFAR-10 では一クラスあたりの学習サンプル数が 5000 個であるのに対し，CIFAR-100 では 500 個と少ないため，少サンプルからも安定に学習可能である提案手法の性質が奏功したものと思われる．これは，STL-10 において提案手法が非常に良好な性能を示していることからも裏付けられる．また，別の要因として，FWM は判別分析と同様にクラス数が増えるほど固有値問題のランクが増し多くの特徴がとれるようになるため，よりクラス数の多い CIFAR-100 において有利に働いた可能性がある．今後，よりクラス数の多い大規模なデータセットを用い，この点の検証を進めたい．

5 結論

本研究では，畳み込みニューラルネットワークの畳み込み層を一層ごとに基本的な固有値問題の解析解として導出するアプローチに着目し，さまざまな構成要素を網羅的に検証した．特に，識別的手法であるフィッシャー重みマップを畳み込み層へ用い，ReLU による活性化関数や平均値プーリングとあわせて用いる方法が効果的であることを示した．

提案手法は深層学習の文脈においてはネットワークの識別的な事前学習として位置づけられる．一層ごとに畳み込み射影を解析的に導出するため，最適化等の細かいノウハウが必要なく，扱いが容易である．また，学習に要する計算コストも比較的小さく，汎用 CPU により実行可能である．本手法は各層を分離して学習する近似的な方法であるにも関わらず，全層通じた最適化を行う先行研究に匹敵する性能を示しており，興味深い結果であると言える．今後，本手法により構築されたネットワークを初期状態として fine-tuning を行い，更に性能を向上させるアプローチを検討したい．

謝辞

本研究は，公益財団法人放送文化基金，JST CREST の研究課題「複雑データからのディープナレッジ発見と価値化」の支援を得て実施された．

参考文献

[1] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. PAMI*, 19(7):711–720, 1997.

[2] Y. Bengio. Practical recommendations for gradient-based training of deep architectures. *Neural Networks: Tricks of the Trade*, 2012.

[3] L. Bo, X. Ren, and D. Fox. Unsupervised feature learning for RGB-D based object recognition. In *Proc. ISER*, 2012.

[4] Y. L. Boureau, J. Ponce, and Y. LeCun. A theoretical analysis of feature pooling in visual recognition. In *Proc. ICML*, 2010.

[5] D. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *Proc. IEEE CVPR*, 2012.

表 4　提案手法による多層ネットワークの識別精度 (%) の比較 (STL-10).

Architecture	Acc.
(1) $K_m(9,256)$-MP_2	54.1
(2) $K_m(9,256)$-$MP(4,2)$-$C(3,256)$-R_2-AP_2	61.2
(3) $K_m(9,256)$-$MP(4,2)$-$C(3,256)$-$AP(4,2)$-$C(3,256)$-R_2-AP_2	64.0
(4) $K_m(9,256)$-$MP(4,2)$-$C(3,256)$-R-$AP(4,2)$-$C(3,256)$-R_2-AP_2	63.3
(5) $K_m(9,256)$-$MP(4,2)$-$C(3,256)$-R_2-$AP(4,2)$-$C(3,256)$-R_2-AP_2	64.2
(6) $K_m(9,256)$-$MP(4,2)$-$C(3,256)$-$AP(4,2)$-$C(3,256)$-$AP(4,2)$-$C(3,256)$-R_2-AP_2	**65.7**
(2)+(3)+(6)	**66.0**

表 5　提案手法による多層ネットワークの識別精度 (%) の比較 (CIFAR-10).

Architecture	Acc.
(1) $K_m(5,256)$-AP_2	72.2
(2) $K_m(5,256)$-$C(3,512)$-R_2-AP_2	76.3
(3) $K_m(5,256)$-$C(3,256)$-$AP(3,2)$-$C(3,512)$-R_2-AP_2	**77.0**
(4) $K_m(5,256)$-$C(3,256)$-R-$AP(3,2)$-$C(3,512)$-R_2-AP_2	76.4
(5) $K_m(5,256)$-$C(3,256)$-R_2-$AP(3,2)$-$C(3,512)$-R_2-AP_2	76.4
(6) $K_m(5,256)$-$C(3,256)$-$AP(3,2)$-$C(3,256)$-$AP(3,2)$-$C(3,512)$-R_2-AP_2	76.4
(2)+(3)+(6)	**79.1**

[6] A. Coates, H. Lee, and A. Y. Ng. An analysis of single-layer networks in unsupervised feature learning. In *Proc. AISTATS*, 2011.

[7] A. Coates and A. Ng. Selecting receptive fields in deep networks. In *Proc. NIPS*, 2011.

[8] A. Coates and A. Ng. The importance of encoding versus training with sparse coding and vector quantization. In *Proc. ICML*, 2011.

[9] A. Coates and A. Ng. Learning feature representations with K-means. *Neural Networks: Tricks of the Trade*, 2012.

[10] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Proc. ECCV Workshop on Statistical Learning in Computer Vision*, 2004.

[11] D. Erhan, Y. Bengio, A. Courville, P. A. Manzagol, and P. Vincent. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11:625–660, 2010.

[12] K. Fukushima. Neocognitron for handwritten digit recognition. *Neurocomputing*, 51:161–180, 2003.

[13] R. Gens and P. Domingos. Discriminative learning of sum-product networks. In *Proc. NIPS*, 2012.

[14] I. J. Goodfellow, D. Warde-farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. In *Proc. ICML*, 2013.

[15] T. Harada, Y. Ushiku, Y. Yamashita, and Y. Kuniyoshi. Discriminative spatial pyramid. In *Proc. IEEE CVPR*, pages 1617–1624, 2011.

[16] G. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. In *arXiv preprint*, 2012.

[17] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313:504–507, 2006.

[18] D. H. Hubel and T. N. Wiesel. Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148:574–591, 1959.

[19] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. Lecun. What is the best multi-stage architecture for object recognition? In *Proc. IEEE ICCV*, 2009.

[20] A. Krizhevsky. *Learning multiple layers of features from tiny images*. Master's thesis, Toronto University, 2009.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Proc. NIPS*, 2012.

[22] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. IEEE CVPR*, volume 2, 2006.

[23] Q. V. Le, M. A. Ranzato, R. Monga, M. Devin, K. Chen, G. S. Corrado, and A. Y. Ng. Building high-level features using large scale unsupervised learning. In *Proc. ICML*, 2012.

[24] Y. LeCun. The MNIST database of handwritten digits. `http://yann.lecun.com/exdb/mnist/`.

[25] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 1998.

[26] M. Lin, Q. Chen, and S. Yan. Network in network. In *Proc. ICLR*, 2014.

[27] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proc. ICML*, 2010.

[28] M. A. Ranzato, C. Poultney, S. Chopra, and Y. LeCun. Efficient learning of sparse representations with an energy-based model. In *Proc. NIPS*, 2006.

[29] Y. Shinohara and N. Otsu. Facial expression recognition using Fisher weight maps. In *IEEE FG*, 2004.

[30] P. Simard, D. Steinkraus, and J. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *Proc. ICDAR*, 2003.

[31] N. Srivastava and R. Salakhutdinov. Discriminative transfer learning with tree-based priors. In *Proc. NIPS*, number Icml, 2013.

[32] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: A large dataset for nonparametric object and scene recognition. *IEEE Trans. PAMI*, 30(11):1958–70, Nov. 2008.

[33] M. Turk and A. Pentland. Face recognition using eigenfaces. In *Proc. IEEE CVPR*, 1991.

[34] M. D. Zeiler and R. Fergus. Stochastic pooling for regularization of deep convolutional neural networks. In *arXiv preprint*, 2013.

表 6　STL-10, CIFAR-10/100, MNIST における先行研究との識別精度比較 (%).

STL-10		
先行研究	1-layer Sparse Coding [8]	59.0
	3-layer Learned Receptive Field [7]	60.1
	Discriminative Sum-Product Network [13]	62.3
	Hierarchical Matching Pursuit [3]	64.5
提案手法	$K_m(9,256)$-$MP(4,2)$-$C(3,256)$-$AP(4,2)$-$C(3,256)$-$AP(4,2)$-$C(3,256)$-R_2-AP_2	65.7
	$K_m(9,1024)$-$MP(4,2)$-$C(3,512)$-$AP(4,2)$-$C(3,256)$-$AP(4,2)$-$C(3,256)$-R_2-AP_2	66.4
	$K_m(9,1024)$-$MP(4,2)$-$C(3,512)$-$AP(4,2)$-$C(3,256)$-$AP(4,2)$-$C(3,256)$-R_2-AP_2 (*)	**66.9**
CIFAR-10		
先行研究	3-Layer Learned Receptive Field [7]	82.0
	CNN [16]	83.4
	Discriminative Sum-Product Network [13]	84.0
	CNN (1 locally connected layer) [16]	84.4
	CNN + Stochastic Pooling [34]	84.9
	CNN + Maxout [14]	88.3
	Network in Network [26]	**89.6**
提案手法	$K_m(5,1024)$-$C(3,256)$-$AP(3,2)$-$C(3,512)$-R_2-AP_3	80.4
	$K_m(5,1024)$-$C(3,256)$-$AP(3,2)$-$C(3,256)$-$AP(3,2)$-$C(3,512)$-R_2-AP_3 (*)	**81.9**
CIFAR-100		
先行研究	CNN + Stochastic pooling [34]	57.49
	CNN + Maxout [14]	61.43
	CNN + Tree-based prior [31]	63.15
	Network in Network [26]	**64.32**
提案手法	$K_m(5,6400)$-$C(1,1000)$-$AP(4,2)$-$C(3,1000)$-$AP(3,2)$-$C(3,1000)$-R_2-AP_3	60.80
	$K_m(5,6400)$-$C(1,1000)$-$AP(4,2)$-$C(3,1000)$-$AP(3,2)$-$C(3,1000)$-R_2-AP_3 (*)	**62.05**
MNIST		
先行研究	CNN (Unsupervised pretraining) [28]	99.40
	CNN (Unsupervised pretraining) [19]	99.47
	CNN + Stochastic Pooling [34]	99.53
	Network in Network [26]	99.53
	CNN + Maxout [14]	99.55
提案手法	$Rand(5,1024)$-R-$AP(4,2)$-$C(3,512)$-R_2-AP_4	99.50
	$Rand(5,2048)$-R-$C(1,1024)$-$AP(4,2)$-$C(3,512)$-R_2-AP_4	**99.60**