平成 27 年度

修士学位論文

ディープアーキテクチャと Extreme Learning Machine の併用による学習精度の向上

A Study on Improvement of Deep Neural Network using Extreme Learning Machine

1185088　　松尾　達郎

指導教員　　吉田　真一

2016 年 2 月 26 日

高知工科大学大学院 工学研究科 基盤工学専攻

情報システム工学コース

要 旨

ディープアーキテクチャと Extreme Learning Machine の併用による学習精度の向上

松尾　達郎

近年，ディープアーキテクチャと呼ばれる多層かされたフィードフォワード型階層的ニューラルネットワークの効果的な学習アルゴリズムが提案されている．一般にディープラーニングと呼ばれ，自動的な特徴設計も実現できるとして注目されている．特に畳み込みニューラルネットワーク (CNN) は一般物体認識のコンペティションにおいて 2 位のチームに誤差 10%以上の差をつけ優勝したことで，ディープラーニングが注目されるきっかけを作っており，その後も画像認識の分野において大きな成果を挙げている．ディープラーニングが効率的な多層ネットワークの学習手法として注目される一方で，隠れ層を 1 層に限定して学習の高速化と最適化を図った Extreme Learning Machine（ELM）が提案されており，理論的な近似能力も高いことが分かっている．そこで，本研究では，画像認識分野において成果を挙げている畳み込みニューラルネットワークの最終段に ELM を組み合わせた CNN-ELM を提案し，その判別精度と学習時間について比較する．データセットには手書き数字認識の「MNIST」，四角形の大小判別を行う「Rectangles」，一般物体認識向けの「Caltech 101」の 3 種類を用い，CNN と CNN-ELM の判別実験を行う．実験の結果，CNN-ELM は ELM の隠れ層におけるニューロン数を適切に設定することで，CNN より優れた認識精度を実現可能であり，ノイズが含まれた画像に対するロバストな判別能力を有していることを示す．また，CNN-ELM は目標とする認識誤差を CNN より短時間の学習で実現可能であることを示す．この結果は，高い精度を求める場合だけでなく，ノイズが含まれた画像に対して判別タスクを行う場合，ある程度の精度の学習を短時間で達成したい場合などに CNN-ELM

が有用であることを示している．

キーワード　畳み込みニューラルネットワーク，Extreme Learning Machine，ディープアーキテクチャ，ディープラーニング，画像認識

Abstract

A Study on Improvement of Deep Neural Network using Extreme Learning Machine

Tatsuro MATSUO

Recently, the deep learning, which is learning methods for deeply multiple-layered neural network is widely used for various scene. Deep learning methods enable the automatic feature extraction. They use or make sparse network and then it is able to learn weights of links between neurons placed in the early layers. Extreme learning machine(ELM) has been proposed as the learning method of the full-connected neural network whose number of hidden layers is limited to one. ELM enables the learning speed increasing and the higher approximation capability. In this research, CNN-ELM, which combines ELM in the final stage of the CNN(convolutional neural network) is proposed. CNN-ELM is applied to three image recognition tasks. The datasets are MNIST, which is for handwritten digits recognition, and Rectangles, which recognize the square size, and Caltech 101, which is for general object recognition. The results of the experiments show that the CNN-ELM is possible to realize a better recognition accuracy compared with that of CNN, and CNN-ELM has the robustness for recognition capability for the images including gaussian noise when the number of neurons in the hidden layer of ELM is appropriately set. Also, the recognition accuracy of CNN-ELM converges the same recognition accuracy of CNN in shorter time. This result shows that CNN-ELM is useful not only in the case of obtaining a high accuracy but also in the case of recognition tasks for the images including noise, and in the case of early learning

convergence.

key words Convolutional Neural Network, Extreme Learning Machine, Deep architecture, Deep learning, Image Recognition

目次

図目次

表目次

第 1 章

序論

近年，多層化されたニューラルネットワークを効果的に学習し，人間の知見に基づいて行っていた特徴設計より判別に適した特徴量を自動設計するようなディープラーニング（Deep Learning）が注目されており，階層型ニューラルネットワークを多層化し，深い層を形成することから深層学習と呼ばれている．従来のパターン認識手法では，正規化やノイズ除去などの前処理を行ったデータに対して，人間の知見に基づいた特徴設計を行い，判別器を用いて得られた特徴を判別する．特徴設計は認識対象から判別に用いる特徴の設計を行う工程であり，画像認識における SIFT（Scale-Invariant Feature Transform）特徴や音声認識におけるメル周波数ケプストラム係数（Mel-Frequency Cepstrum Coefficients）など，認識タスクごとに人間の知見に基づく特徴の設計を行う．判別器では，特徴抽出により得られた特徴に対して，最近傍決定則やニューラルネットワーク，サポートベクターマシンなどの手法を用いた判別ルールの生成を行い，判別を行う．こうした従来手法に対し，ディープラーニングでは，多層ニューラルネットワークの内部において，前半部が判別に有用な特徴の設計，後半部で設計された特徴を用いた判別を行う判別器の役割を担うことで，高い精度を示している．米 Google 社は YouTube の動画からランダムにサンプリングされた 100 万枚の画像を多層ニューラルネットワークの入力とし，ディープラーニングを用いた学習を行い，人間の顔や猫の顔に強く反応するニューロンを発見することで，ディープラーニングによる有用な特徴設計が行えていることを示している [12].

ディープラーニングが考案される以前からニューラルネットワークの多層化についての研究は行われているが，少数の層で構成されるニューラルネットワークの学習方法として一般に用いられている誤差逆伝搬法をそのまま多層ニューラルネットワークの学習に用いると，

局所収束や勾配消失の問題が深刻になり，効率的な学習方法は提案されてきていない．しかし，画像認識分野における Convolutional Neural Network（CNN）が一般物体認識のコンペティション ILSVRC2012（ImageNet Large Scale Visual Recognition Challenge 2012）において，2 位以下のチームに識別率で 10%以上の差をつけ優勝したことで，企業や大学において様々なディープラーニングの手法が開発，改良されている．その中でも，画像認識における CNN は人間の視覚野の構造を模し，物体が微少な移動やスケール変化をしても検出できるように特徴抽出部の構造設計が行われている．こうしたニューラルネットワークを多層化する考え方に対して，3 層に限定して学習の高速化と最適化を図った Extreme Learning Machine（ELM）が提案されている．ELM は一部のパラメータをランダムに決定するが，隠れ層のニューロン数を十分大きくすることで高い関数近似能力を持つことが知られている．そこで，本研究では，画像認識分野において，大きな成果を挙げている CNN をそのまま用いるのではなく，まず CNN の学習により自動的に設計された特徴抽出部を用いて次元削減された中間特徴を抽出し，その後高速な学習が可能な Extreme Learning Machine を教師付き学習として用いる CNN-ELM モデルを提案し，ELM の隠れ層を十分に用意することで，CNN よりも認識精度が向上することを示す．また，ノイズを含んだデータセットを生成し，CNN-ELM を用いた判別を行うことで，ノイズに対してロバストな判別能力があることを示す．

第 2 章では，提案手法の関連技術であるニューラルネットワークと ELM，CNN に加えてディープラーニングにおける特徴設計について記す．第 3 章では，提案する CNN-ELM モデルについて記す．第 4 章では，画像データセットを用いた CNN-ELM モデル性能評価実験を行う際のデータセットや前処理，学習適用時のパラメータについて記す．第 5 章では，第 4 章で行った実験の結果を示し，考察する．第 6 章で，研究全体についてまとめる．

第 2 章

関連技術

本章では，本研究で提案するモデルのうち，ネットワーク前段の特徴設計に用いる CNN（Convolutional Neural Network:畳み込みニューラルネットワーク）および後段の 3 層 NN の学習手法である Extreme Learning Machine について説明する．

2.1 パターン認識における特徴設計

パターン認識とは観測された事象をあらかじめ定義されたカテゴリと対応づける処理であり，一般物体認識に代表されるような画像認識や音声認識など，多くのタスクへ応用されている．図 2.1 に従来手法とディープラーニングにおけるパターン認識の手順を示す．まず，入力されるデータのサイズ変換や正規化，ノイズ除去などの前処理を行う．次に，前処理が行われたデータから判別に有用な特徴を設計する．最後に，判別器を用いて設計した特徴を定義したクラスに判別する．パターン認識の精度は，特徴設計において判別しやすい特徴を設計できているかに大きく影響を受ける．渡辺は醜い家鴨の子の定理において，有限個のサンプルで 2 つの物を区別する際に，任意の 2 つの物を区別できる特徴を全て選出したとすると，それは高々有限個であり，さらにそれらの特徴の重要度が全て同じであったとすると，任意の 2 つの物が同じカテゴリを示す特徴の数は，2 つの物の選び方によらず一定であり，2 つの物が同じカテゴリとなる特徴の多さで，2 つの物の類似性を定義しようとしても，すべての 2 つの物は等しく類似してしまうことを示している [15]．つまり，1 羽の白鳥と 1 羽の家鴨の類似度と，1 羽の家鴨と 1 羽の別の家鴨との類似度は同じになるということになる．従って，白鳥と家鴨を判別するためには，選出した特徴に対して，特にそれら

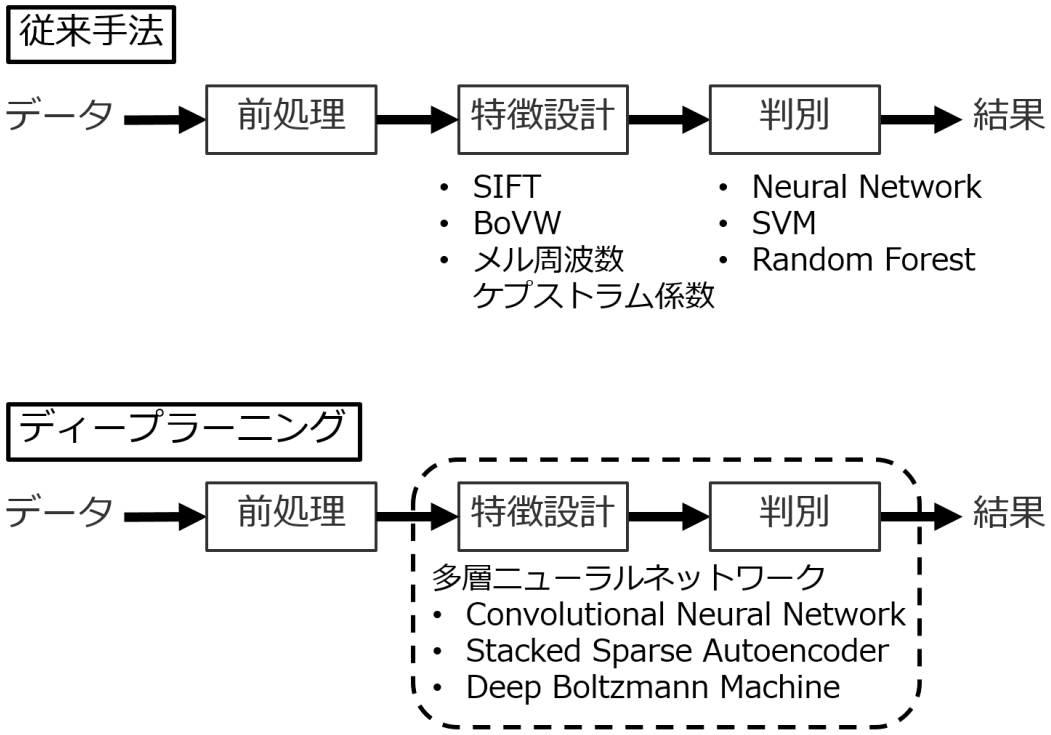

図 2.1　従来の手法とディープラーニングにおけるパターン認識手順

を判別しやすい特徴に対して重要度を加味した価値の重み付けを行う必要がある．従来の特徴設計手法では，人間の知見に基づいて特徴の設計と価値の重み付けを行っており，画像認識におけるスケール不変性を実現した局所特徴量である SIFT（Scale-Invariant Feature Transform）特徴量や SIFT 特徴量を用いて 1 枚の画像をベクトル表現する BoVW（Bag of Visual Words），音声認識における声道の特性を考慮したメル周波数ケプストラム係数（Mel-Frequency Cepstrum Coefficients）などがあり，それぞれの認識タスクに特化して判別しやすい特徴を設計している．一方，ディープラーニングでは，前処理を行った画像や音声のデータを直接多層ニューラルネットワークに入力し，多層ニューラルネットワークの学習を行うことで，自動的な特徴の設計を行うことを目指している．

2.1.1　ディープラーニングの特徴設計

ディープラーニングを用いた学習では，前処理を行ったデータを多層ニューラルネットワークに直接入力し，判別器を構成する．従来の特徴設計が人間の知見に基づいて行われていたのに対し，ディープラーニングでは，多層ニューラルネットワークの前半が特徴設計として動作し，後半が判別を行う判別器として動作する．2012 年に Le らは米 Google 社が提供する YouTube からサンプリングした 1000 万枚の画像を訓練用データセットとして用い，11 層のニューラルネットワークに対して，ディープラーニングを用いた学習を行うことで，ニューラルネットワークの中間層に人の顔に強く反応するニューロンと猫の顔に強く反応するニューロンがそれぞれ生成され，ディープラーニングの学習過程において，ニューラルネットワーク前半のパラメータが調整されることで，自動的に特徴設計が行われていることを示している [12]．Zeiler らは ILSVRC2012 で高精度を示した多層ニューラルネットワークである CNN（Convolutional Neural Network）の中間層の可視化を行い，層を重ねるごとに単純な構造の特徴から複雑な構造の特徴の設計が行われていることを示している [11]．Donahue らは CNN の学習により，浅い層で単純な特徴の設計が行われ，層が深くなるにつれて単純な特徴を組み合わせた画像の意味を良く表現している複雑な特徴が設計されていき，クラス分離能力が向上することを示している [7].

2.2　ニューラルネットワーク

ニューラルネットワーク (NN) とは，人間の神経細胞を模倣した複数の人工的な素子を接続する工学モデルであり，パターン認識問題に対し，計算機に学習を行わせることで識別関数を得る手法である．人間の神経回路と区別して人工ニューラルネットワークと呼ぶことがあるが，本論文では単純にニューラルネットワークと呼ぶ.

ニューラルネットワーク以前にはマカロック・ピッツの形式ニューロンモデルの考え方を基にして考案された古典的な学習規則として入力層と出力層のみの 2 層からなる単純パーセプトロンが存在する [3]．サンプル数 N の学習データ $(\mathbf{x}^n, \mathbf{t}^n) \in \mathbf{R}^d \times \mathbf{R}^m$

が与えられた際の n 番目のサンプルにおける単純パーセプトロンの出力 o^n は入力を $\mathbf{x}^n = (x_0^n = 1, x_1^n, \cdots, x_d^n)^T$，結合重みを $\mathbf{w} = (w_0, w_1, \cdots, w_d)$ とすると，

$$o^n = \sum_{i=0}^{d} w_i x_i^n = \mathbf{w}^T \mathbf{x}^n \tag{2.1}$$

の重み付き線形和で表され，第一成分 w_0 は，バイアス項（重み付き線形和の定数項）である．この出力 o^n は線形識別関数となるため，すべての o^n が理想の出力となるように結合重み w_i を調整する学習規則を用いて線形識別関数の学習を行う．しかし，単純パーセプトロンで実現できる識別関数は線形識別関数であり，線形分離可能な問題を解くことは出来るが，線形分離不可能な問題は解くことが出来ないことが証明されている [10]．このような問題はニューロンの層を多層化し，ニューロン内部での処理において非線形な処理を加えることで解決することが出来る．つまり，入力層と出力層の間に隠れ層と呼ばれる層を追加することで，3 層以上のネットワークを構成し，隠れ層のニューロン内部において活性化関数と呼ばれる非線形関数を用いることでネットワーク全体の出力を非線形関数としており，線形分離不可能な問題に対応している．この非線形な活性化関数を取り入れた 3 層以上のネットワークを一般にニューラルネットワークと呼ぶ．ニューラルネットワークにおいて，パーセプトロンの学習規則を用いて学習できるのは最終層の結合重みのみであり，中間層における重みの学習を行うことは出来ない．1986 年に Rumelhart らは 3 層以上のニューラルネットワークの学習手法として誤差逆伝搬法を提案している [1]．誤差逆伝搬法では理想とする出力と実際のニューラルネットワークの出力の 2 乗誤差最小化を勾配降下法を用いて行うことで，中間層における結合重みの更新を可能にしており，Rumelhart らは 5 層のニューラルネットワークを用いて 2 つの同型の家系図における人間関係を学習させることで，隠れ層のノードで 2 つの家系図に共通な人間関係情報と家系図ごとに異なる名前情報とが分離されており，効率的な特徴設計がなされていることを示している．この誤差逆伝搬法が，階層型（フィードフォワード型）ニューラルネットワークの学習法として広く用いられてきた．

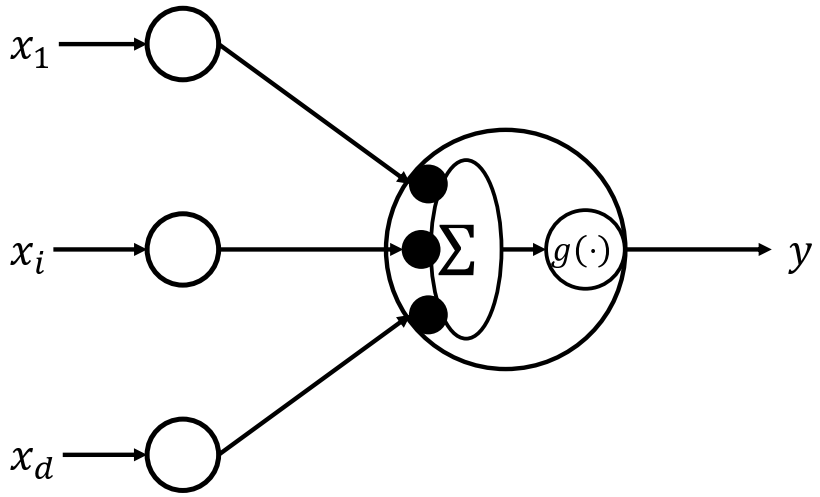

図 2.2　ニューラルネットワークにおける単一ニューロンの入出力処理

2.2.1　3 層ニューラルネットワークの構造

人間のニューロンが入力信号に対して，全か無かの法則に従った二値情報を出力するのに対して，ニューラルネットワークにおけるニューロンの出力は図 2.2 のように前の層の出力値に結合重みを乗じた値の総和に対して，微分可能な活性化関数 $g(\cdot)$ を用いて算出される．

図 2.3 は入力層，隠れ層，出力層からなる 3 層のニューラルネットワークの構造を表している．入力層のニューロンは，入力値をそのまま出力するため，サンプル数 N の学習データ $(\mathbf{x}^n, \mathbf{t}^n \in \mathbf{R}^d \times \mathbf{R}^m)$ が与えられた際の，n 番目のサンプルにおける入力層の出力は $\mathbf{x}^n = (x_0^n \equiv 1, x_1^n, \cdots, x_d^n)^T$ となり，隠れ層のニューロン数を L とすると，n 番目のサンプルにおける隠れ層の j 番目のニューロンへの入力値 h_j^n は，

$$h_j^n = \sum_{i=0}^{d} w_{ji} x_i^n, \tag{2.2}$$

となる．ここで，$w_{ji} = (w_{j0}, w_{j1}, \cdots, w_{jd})$ は入力層から隠れ層への結合重みベクトル，w_{j0} はバイアス項であり，常に 1 である．n 番目のサンプルにおける隠れ層の j 番目のニューロ

ンの出力値 H_j^n は，この入力に対して活性化関数 $g(\cdot)$ を用いて

$$H_j^n = g(h_j^n), \tag{2.3}$$

で表される．活性化関数には式 2.4 のシグモイド関数や式 2.5 のハイパボリックタンジェント関数に加え，近年の多層ニューラルネットワークでは式 2.6 の ReLU(Rectifiled Linear Units) などが用いられている．

シグモイド関数

$$g(x) = \frac{1}{1 + \exp(-\alpha x)} \tag{2.4}$$

ハイパボリックタンジェント関数

$$g(x) = \tanh x \tag{2.5}$$

ReLU

$$g(x) = \max(0, x) \tag{2.6}$$

判別クラス数と等価な値となる出力層のニューロン数が m となるため，n 番目のサンプルにおける出力層の k 番目のニューロンへの入力値 $h_k^n (k = 1, \cdots, m)$ は，

$$h_k^n = \sum_{j=0}^{L} w_{kj} H_j^n = \sum_{j=0}^{L} w_{kj} g(\sum_{i=0}^{d} w_{ji} x_i^n), \tag{2.7}$$

で表され，そのニューロンの出力 o_k^n は，

$$o_k^n = \tilde{g}(h_k^n) = \tilde{g}(\sum_{j=0}^{L} w_{kj} H_j^n) = \tilde{g}(\sum_{j=0}^{L} w_{kj} g(\sum_{i=0}^{d} w_{ji} x_i^n)), \tag{2.8}$$

となる．ここで，w_{k0} は隠れ層におけるバイアス項であり H_0^n は常に 1 である．$\tilde{g}(\cdot)$ は出力層における線形または非線形の活性化関数で，多クラス分類の場合は，式 2.9 のソフトマックス関数を用いてネットワークの出力に確率的解釈を与え，最大事後確率を与えるクラスを判別結果とする場合がある．

$$\tilde{g}(h_k^n) = p(t_k^n = 1 \mid \mathbf{x}^n) = \frac{\exp h_k^n}{\sum_{k=1}^{m} \exp h_k^n} \tag{2.9}$$

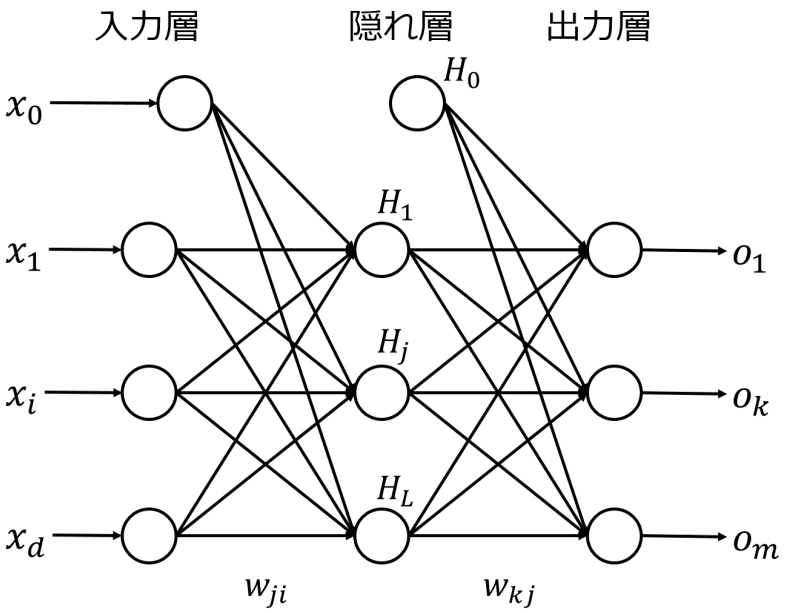

図 2.3　ニューラルネットワークの構成

2.2.2　誤差逆伝搬法を用いたニューラルネットワークの学習

ニューラルネットワークにおいて隣接する 3 つの層を考え，ある層における j 番目のニューロンをニューロン j，1 段前の層の i 番目のニューロンをニューロン i，1 段後の層の k 番目の層のニューロンをニューロン k とすると，サンプル数 N の学習データ $(\mathbf{x}^n, \mathbf{t}^n \in \mathbf{R}^d \times \mathbf{R}^m)$ が与えられた際の n 番目のサンプルにおけるニューロン j への入力 h_j^n は式 2.2 に従って，

$$h_j^n = \sum_{i=0} w_{ji} H_i^n, \tag{2.10}$$

となる．ここで，w_{ji} は結合重みで w_{j0} はバイアス項となり，H_i^n は 1 段前の層のニューロンの出力である．また，ニューロン j の出力は式 2.3 となる．誤差逆伝搬法では，全サンプルにおけるニューラルネットワークの出力と理想とする出力との二乗誤差の最小化を勾配法を用いた重みの修正により行う．ここで，出力層の l 番目のニューロンをニューロン l とすると，n 番目のサンプルにおける誤差関数 $E^n(\mathbf{w})$ は

$$E^n(\mathbf{w}) = \frac{1}{2} \sum_{l=1}^{m} (H_l^n - t_l^n)^2, \tag{2.11}$$

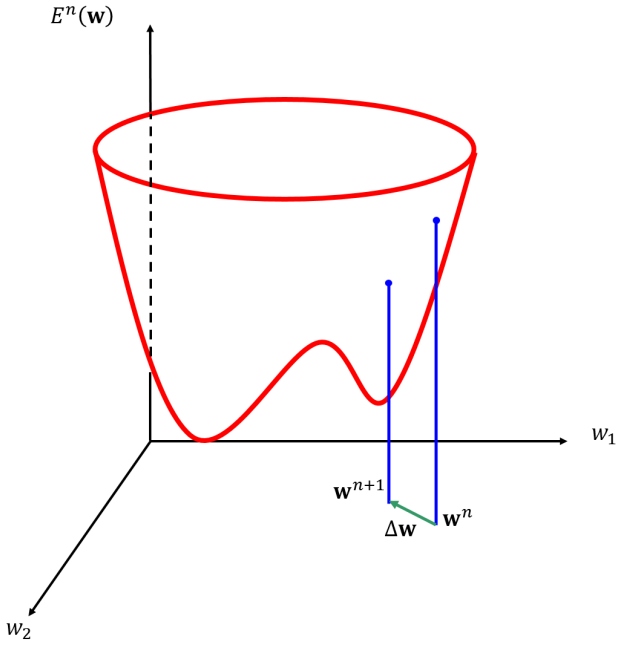

図 2.4　勾配法を用いた誤差関数 $E^n(\mathbf{w})$ のパラメータ $\mathbf{w}$ の更新

の二乗誤差で表され，全サンプルにおける誤差関数 $E(\mathbf{w})$ は

$$E(\mathbf{w}) = \sum_{n=1}^{N} E^n(\mathbf{w}), \tag{2.12}$$

となる．ここで，$\mathbf{w}$ はネットワークの結合重みのパラメータセットである．図 2.4 は，重みが 2 つの場合の誤差関数を示している．誤差関数は，パラメータ数と等しい次元数で凸部分や凹部分を含んだ関数となる．そのため，勾配法における重みの更新において，誤差関数の値が小さくなる方向に重みの更新を行う．

更新後の重み w_{ji}^{n+1} は，現在の重み w_{ji}^{n} から更新量を減算した

$$w_{ji}^{n+1} = w_{ji}^{n} - \eta \frac{\partial E^n(\mathbf{w})}{\partial w_{ji}^{n}}, \tag{2.13}$$

で表される．ここで，η は小さな正の定数をとり，一度の更新でどの程度パラメータを更新するかを制御する学習係数である．誤差 $E^n(\mathbf{w})$ に対する w_{ji} での偏微分

$$\frac{\partial E^n(\mathbf{w})}{\partial w_{ji}} = \frac{\partial E^n(\mathbf{w})}{\partial h_j^n} \cdot \frac{\partial h_j^n}{\partial w_{ji}}, \tag{2.14}$$

は，右辺第 1 項を誤差信号 ε_j^n，式 2.10 より右辺第 2 項を

$$\frac{\partial h_j^n}{\partial w_{ji}} = H_i^n, \tag{2.15}$$

とすると，式 2.13 は次のように表される．

$$w_{ji}^{n+1} = w_{ji}^n - \eta \varepsilon_j^n H_i^n \tag{2.16}$$

個別のニューロンに対する誤差信号は ε_j^n は

$$\begin{aligned} \varepsilon_j^n &= \frac{\partial E^n(\mathbf{w})}{\partial h_j^n} \\ &= \frac{\partial E^n(\mathbf{w})}{\partial H_j^n} \cdot \frac{\partial H_j^n}{\partial h_j^n} \\ &= \frac{\partial E^n(\mathbf{w})}{\partial H_j^n} g_j'(h_j^n), \end{aligned} \tag{2.17}$$

で計算可能である．ここで，$g_j'(\cdot)$ は活性化関数の微分である．式 2.17 の第 1 項はユニット j の位置により場合分けされる．まずユニット j が出力層にあるとき，式 2.11 より

$$\frac{\partial E^n(\mathbf{w})}{\partial H_j^n} = H_j^n - t_j^n, \tag{2.18}$$

となる．次にユニット j が中間層にあるときは

$$\begin{aligned} \frac{\partial E^n(\mathbf{w})}{\partial H_j^n} &= \sum_k \frac{\partial E^n(\mathbf{w})}{\partial h_k^n} \cdot \frac{\partial h_k^n}{\partial H_k^n} \\ &= \sum_k \varepsilon_k^n w_{kj}, \end{aligned} \tag{2.19}$$

となる．誤差逆伝搬法による重み更新では活性化関数の微分が行われるため，用いる活性化関数は微分可能である必要がある．活性化関数にシグモイド関数を用いた場合の誤差信号 ε_j^n は次のように求めることが出来る．

$$\varepsilon_j^n = \begin{cases} (H_j^n - t_j^n) H_j^n (1 - H_j^n) & (\text{ユニット } j \text{ が出力層にあるとき}) \\ (\sum_k \varepsilon_k^n w_{kj}) H_j^n (1 - H_j^n) & (\text{ユニット } j \text{ が中間層にあるとき}) \end{cases} \tag{2.20}$$

2.2.3　多層ニューラルネットワークの学習困難性

一般にニューラルネットワークは，式 2.20 により誤差信号を逆伝搬する誤差逆伝搬法を用いて，パラメータの学習を行う．これは誤差逆伝搬法がニューロンの層が少ないときに効

果的に学習できるからである．しかし，式 2.17 において誤差信号に含まれる活性化関数の微分によりその値は指数的に減少する．したがって，多層ニューラルネットワークでは出力層に近い層の結合重みのみが学習され，入力層に近い層の結合重みでは誤差信号が消失する勾配消失問題が存在する．また，勾配を算出する誤差関数の空間における次元数はパラメータの個数に依存しており，パラメータが増加することで同一の出力を行う点が増えることになる．したがって，誤差関数にプラトーや鞍点が増加し，局所最適解において学習が終了する可能性が高くなり，少ない層数のニューラルネットワークよりかえって精度が悪くなることが多い．これに対する解決策としては，多層ニューラルネットワークにおけるニューロン同士の結合数を少なくし，疎なネットワークを構築することで誤差信号の減少の影響を少なくし，勾配消失問題を解決する方法と入力層から 1 層ごとに教師なし学習をおこなうことで，あらかじめパラメータの初期パラメータを学習しておくことで，誤差の修正量を減らすという方法が挙げられる．

2.3 Extreme Learning Machine

Extreme Learning Machine（ELM）は，2004 年に G.B.Huang らによって提案された，ニューラルネットワークの学習手法である [5][6]．ELM は隠れ層が一層のみの Single-hidden Layer Feedforward Neural Networks（SLFNs），すなわち入力層，隠れ層，出力層がそれぞれ 1 層からなる 3 層のモデルを対象としている．この SLFNs は，隠れ層におけるニューロンの活性化関数に非多項式関数を用いた荷重加算型ニューロンや，Radial-Basis Function（RBF）型ニューロンを用いた場合は，十分な数の隠れ層のニューロンと適切な重み，閾値のパラメータを設定することで，任意の連続関数を任意の誤差で近似可能であることが示されている [14]．ELM の特徴は，すべてのデータに対する最適なネットワークのパラメータ（ニューロン間の重み）を一度の演算で求めることにある．一般のフィードフォワード型ニューラルネットワークではパラメータの調整に誤差逆伝搬法（Back Propagation）を用いることが多い．誤差逆伝搬法では，勾配法を用いて，出力値と教師信号の平均二乗誤差を

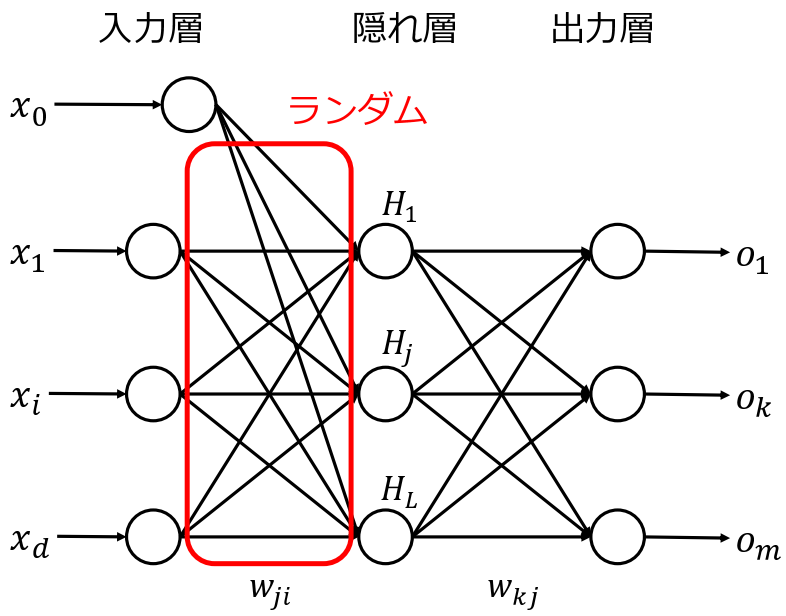

図 2.5　Extreme Learning Machine の構成

小さくすることを目標に，出力層側のニューロンにかかる重みから逐次的にパラメータの調整を行う．しかし，このパラメータの調整では必ずしも大域的な最適解が見つかる訳ではなく，局所最適解に陥る可能性がある．これに対して，ELM は入力層から隠れ層にかかる重みの最適化を諦め，ランダムに決定し，隠れ層から出力層にかかる重みの最適化を一度の演算で求めることで，学習の高速化を図っている．そのため，ELM は，入力層から隠れ層にかかる重みをランダムに決定した SLFNs になるが，十分な数の隠れ層のニューロンと有界で非定数かつ区分連続な活性化関数を用いることで，任意の関数を近似可能である [4]．ELM は勾配法の繰り返し処理ではなく，一度の演算で重みの最適化を行うため，局所最適解に陥ることがなく，隠れ層の出力に対して，最適な隠れ層から出力層にかかる重みを求めることできる．図 2.5 に ELM の構成を示す．隠れ層のニューロン数 L，隠れ層における活性化関数 $g(\cdot)$ の ELM に対して，サンプル数 N の学習データ $(\mathbf{x}^n, \mathbf{t}^n) \in \mathbf{R}^d \times \mathbf{R}^m$ が与えられた際の，n 番目のサンプルにおける隠れ層の j 番目のニューロンの出力 H_j^n は式 2.21 で表される．

$$H_j^n = g(\mathbf{w}_j \cdot \mathbf{x}^n), \tag{2.21}$$

ここで，$\mathbf{w}_j$ と $\mathbf{x}^n$ は

$$\mathbf{w}_j = \begin{pmatrix} w_{j0} \\ w_{j1} \\ \vdots \\ w_{jd} \end{pmatrix}, \mathbf{x}^n = \begin{pmatrix} x_0^n \equiv 1 \\ x_1^n \\ \vdots \\ x_d^n \end{pmatrix}, \tag{2.22}$$

で表され，第 1 成分 w_{j0} はバイアス項（前層ニューロン出力の重み付き線形和の定数項）である．ELM における活性化関数は有界で，非定数かつ区分連続であれば，微分不可能な関数を用いても良いが，本研究では式 2.4 に示すシグモイド関数を用いる．よって，出力層における k 番目のニューロンにかかる重みを $\mathbf{w}_k = (w_{k1}, w_{k2}, \cdots, w_{kL})^T$ とすると，その出力 y_k^n は

$$y_k^n = \sum_{j=1}^{L} w_{kj} H_j^n = \sum_{j=1}^{L} w_{kj} g(\mathbf{w}_j \cdot \mathbf{x}^n), \tag{2.23}$$

となり，ELM の出力値行列 $\mathbf{Y}$ は，隠れ層における出力値行列 $\mathbf{H}$ と隠れ層から出力層にかかる重みベクトルをそれぞれ

$$\mathbf{H} = \begin{bmatrix} H_1^1 & \cdots & H_L^1 \\ \vdots & \ddots & \vdots \\ H_1^N & \cdots & H_L^N \end{bmatrix}, \mathbf{w} = \begin{bmatrix} w_{11} & \cdots & w_{m1} \\ \vdots & \ddots & \vdots \\ w_{1L} & \cdots & w_{mL} \end{bmatrix}, \tag{2.24}$$

と表すと，以下のようになる．

$$\mathbf{Y} = \mathbf{H}\mathbf{w}. \tag{2.25}$$

ELM は式 2.25 で表される出力が教師データのベクトル

$$\mathbf{T} = \begin{pmatrix} \mathbf{t}^{1T} \\ \mathbf{t}^{2T} \\ \vdots \\ \mathbf{t}^{NT} \end{pmatrix}, \tag{2.26}$$

となるように学習を行うため，

$$\mathbf{T} = \mathbf{H}\mathbf{w}, \tag{2.27}$$

となる $\mathbf{w}$ を求めればよいことになる．そのため，隠れ層における出力値行列 $\mathbf{H}$ に対して Moore-Penrose の疑似逆行列を用い，

$$\mathbf{w} = \mathbf{H}^{-1}\mathbf{T}, \tag{2.28}$$

として重み $\mathbf{w}$ を算出する．算出した隠れ層から出力層にかかる重みベクトル $\mathbf{w}$ とランダムに決定した入力層から隠れ層にかかる重みベクトル $\mathbf{w}_j(j = 1, \cdots, L)$ を最適な重みとして ELM 判別器を構成する．一般的なニューラルネットワークでは，隠れ層のニューロン数に加え，学習率 η や収束判定値 ϵ などのパラメータの設定により，判別器の精度は大きく左右されるが，ELM では設定するパラメータは隠れ層のニューロン数のみでの学習を行うため，パラメータの数が少なく，ロバストな解を求めることが出来る．

2.4　畳み込みニューラルネットワーク（CNN）

畳み込みニューラルネットワーク（Convolutional Neural Network）は，2 次元画像に対して高い認識精度を実現しているフィードフォワード型ニューラルネットワークである．そのルーツは視覚神経科学における Hubel と Wiesel の研究にある．Hubel と Wiesel らは，猫の第一次視覚野において，特定の傾きを持つ線分に対して選択的に反応する単純型細胞（Simple Cell）と特定の傾きを持つ線分を移動させても反応する複雑型細胞（Complex Cell）が存在し，これらが受容野を形成していることを確認している [2]．福島らは図 2.6 のような視覚野における受容野の構造の数理モデル化となる Neocognitron という視覚パターン認識能力を持つ多層の人工神経回路を考案している [8]．Neocognitron では S 細胞と名付けた細胞の層 U_S と，C 細胞と名付けた細胞の層 U_C が交互に何段も積層されている．S 細胞は単純型細胞として働き，入力信号の局所的な特徴を抽出し，複雑型細胞として働く C 細胞は複数の S 細胞の出力を集積することで入力信号の位置ずれを許容し，幾何学的な変化に対する不変性を実現している．Neocognitron は積層された中間層の学習に winner-take-all 型の競合学習や add-if-silent 則を用いた教師無し学習を用いているが，CNN では，Neocognitron に対する勾配法を用いた誤差逆伝搬法による教師あり学習を実

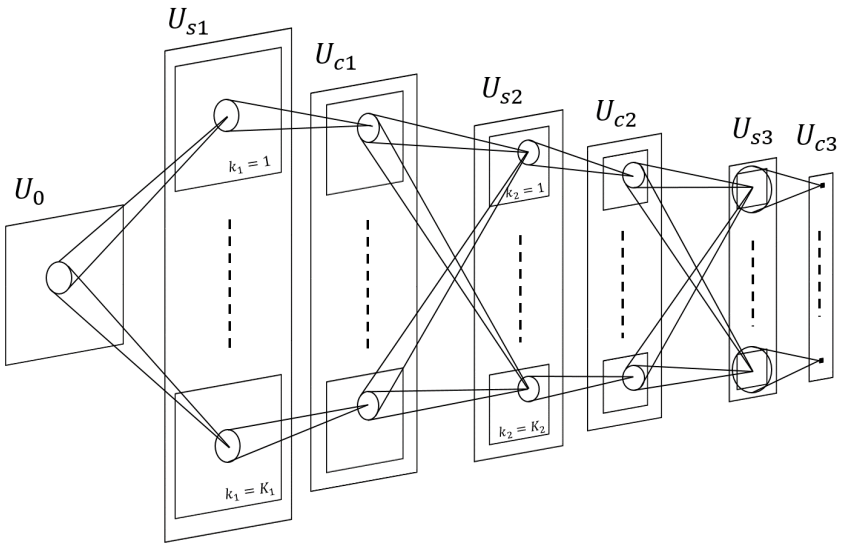

図 2.6　Neocognitron[8] の回路構造

現している [13]．CNN の構成は大きく特徴設計部と判別部に分けられる．図 2.7 に畳み込みニューラルネットワークの構成を示す．CNN において入力画素は入力層のニューロンと一対一に対応し，二次元の特徴マップ（Feature Map）を形成する．その特徴マップに対して，特徴設計部では畳み込みとプーリングを繰り返すことで，幾何学変化に対する不変性を得た特徴量を設計し，判別部において判別を行う．また，畳み込み層とプーリング層における局所受容野の考え方を用いて，部分的な領域のみニューロンを結合させる考え方により，ニューロンの結合数を少なく抑えているため，誤差逆伝搬法を用いて多層ニューラルネットワークの学習する際の誤差の拡散が抑制されている．誤差逆伝搬法を用いた CNN の学習では，判別部の全結合ニューラルネットワークだけでなく，特徴設計部も含めたネットワーク全体での学習を行う．

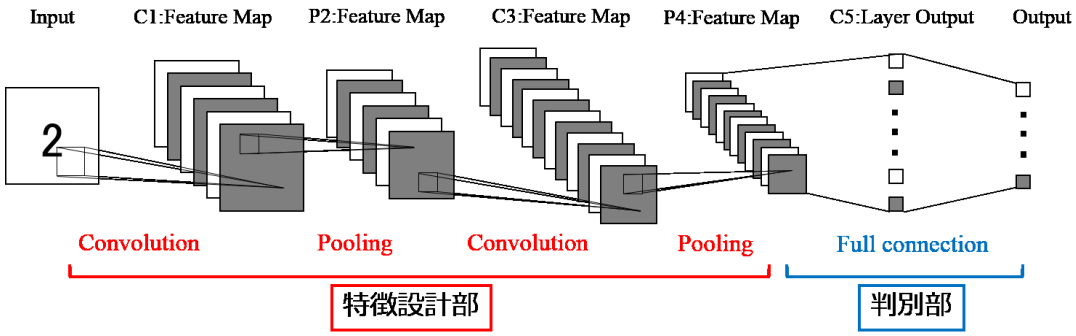

図 2.7　畳み込みニューラルネットワーク（CNN）の構成

2.4.1　特徴設計部

CNN の特徴設計部は畳み込み層とプーリング層を繰り返しすことで特徴量の自動生成を行っている．ここでは，特徴設計部における畳み込み層とプーリング層の処理について述べる．

畳み込み層（Convolution Layer）

図 2.8 は畳み込み層における畳み込み処理を示している．入力は縦横サイズ $S \times S$ 画素の C 枚の特徴マップを $S \times S \times C$ と表し，枚数 C をチャネル数と呼ぶ．入力層のチャネル数はグレースケール画像なら 1 チャネル，RGB カラー画像なら 3 チャネルとなる．これ以降，$S \times S \times C$ の入力を $x_{ij}^{(c)}((i,j,c) \in \{0,1,\cdots,S-1\} \times \{0,1,\cdots,S-1\} \times \{1,2,\cdots,C\})$ と記述する．畳み込み層では前の層の 2 次元の特徴マップに対して，2 次元フィルタの畳み込みを行う．2 次元フィルタを $w_{i,j}^{(c)}((i,j,c) \in \{0,1,\cdots,K-1\} \times \{0,1,\cdots,K-1\} \times \{1,2,\cdots,N\})$，畳み込みを行う際にウィンドウをずらす画素の間隔であるストライドを s とすると，入力特徴マップにおけるサイズ $S \times S$ 画素の各チャネルごとに対応するチャネルのサイズ $K \times K$ の 2 次元フィルタを畳み込み，すべてのチャネルにわたって加算することで，1 チャネルの

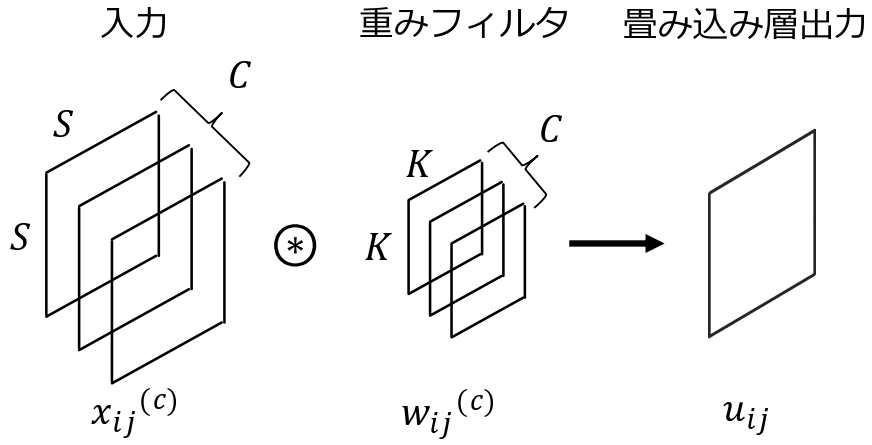

図 2.8　畳み込み処理における特徴マップ一つに関する計算内容

特徴マップ u_{ij} が

$$u_{ij} = \sum_{c=1}^{C} \left[\sum_{(p,q)\in\mathcal{P}_{ij}} \left[x_{pq}^{(c)} w_{p-i,q-j}^{(c)} \right] + b^{(c)} \right], \tag{2.29}$$

のように生成される．ここで，$\mathcal{P}_{ij}$ は，画像中の画素 (si, sj) を頂点とするサイズ $K \times K$ 画素の正方領域,

$$\mathcal{P}_{ij} = \{(si + i', sj + j') | i', j' \in \{0, \cdots, L-1\}\}, \tag{2.30}$$

であり，b^c は各チャネルにおけるバイアス項である．このとき，フィルタの個数を N とすると，畳み込みの出力は,

$$u_{ij}^{(k)} = \sum_{c=1}^{C} \left[\sum_{(p,q)\in\mathcal{P}_{ij}} \left[x_{pq}^{(c)} w_{p-i,q-j}^{(c)(k)} \right] + b^{(c)(k)} \right], \tag{2.31}$$

で算出される．ここで，$k \in \{1, \cdots, N\}$ である．畳み込みの出力 $u_{ij}^{(k)}$ が次の層への入力となるため，フィルタ 1 個につき，1 チャネルの画像が出力される．図 2.9 は畳み込み層での処理を視覚的に表したものである．下位層の特徴マップに対して 2 次元フィルタを用いた

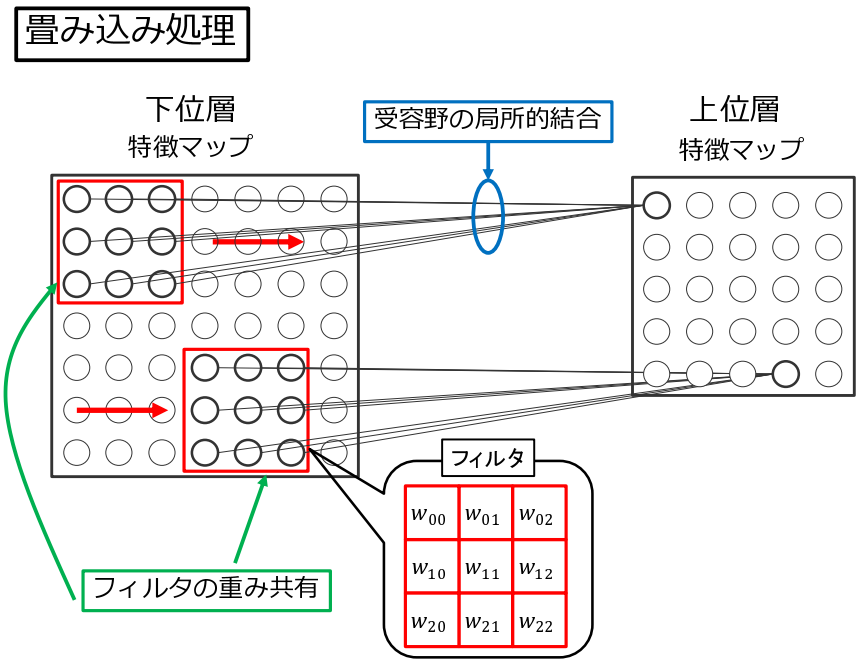

図 2.9　畳み込み層でのニューロンの結合

畳み込みを行うため，上位層の各ニューロンは下位層の一部のニューロンとのみ結合しており，この結合の様子を受容野が局所的であるといい，下位層の一部のニューロンが形成する領域を局所受容野と呼ぶ．また，畳み込みに用いる 2 次元フィルタの係数である結合重みは各ニューロン間で共通の値を用いることになり，この構造を重み共有と呼ぶ．

プーリング層（Pooling Layer）

プーリング層では，畳み込み層の局所的な受容野によって得られたニューロンの反応を一定の領域においてプーリングするため，得られたニューロンの反応における位置情報を一部捨てており，これにより，入力信号の微細な位置ずれに対しても頑健な移動不変性を実現している．図 2.10 は，プーリング層における最大プーリングの処理を表している．プーリング層の一つのニューロンは，畳み込み層と同様に，下位層に局所受容野 $\mathcal{P}_{ij}$ を持っており，このニューロンにおける出力は下位層における局所受容野 $\mathcal{P}_{ij}$ に対してチャネルごとに独立にプーリング処理を用いることで得られる．このため，プーリング層の入力チャネル数と出

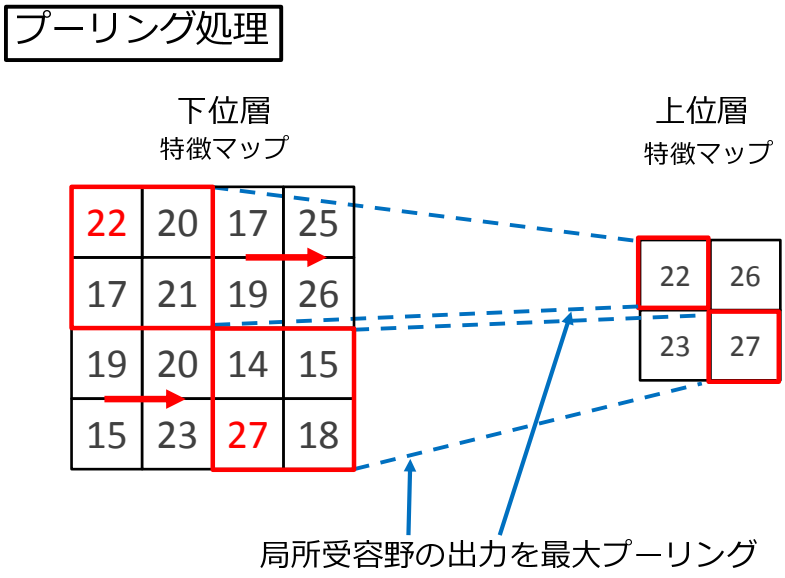

図 2.10　プーリング層におけるプーリング処理

力チャネル数は一致する．プーリングを行う局所受容野のサイズは畳み込み層のサイズに依存しない．代表的なプーリング処理には平均プーリングと最大プーリングがある．ここで，$x_{pq}^{(c)}$ を入力，$y_{ij}^{(c)}$ を出力とすると，平均プーリングは，式 2.32 で表されるように局所受容野の領域内のニューロンからの入力の平均を出力とする．

$$y_{ij}^{(c)} = \frac{1}{[\mathcal{P}_{ij}]} \sum_{(p,q)\in\mathcal{P}_{ij}} x_{pq}^{(c)}. \tag{2.32}$$

また，最大プーリングは，式 2.33 で表されるように局所受容野の領域内のニューロンからの入力の最大値を出力とする．

$$y_{ij}^{(c)} = \max_{(p,q)\in\mathcal{P}_{ij}} x_{pq}^{(c)}. \tag{2.33}$$

2.4.2　判別部

CNN における判別部には下位層のニューロンすべてと結合した全結合のニューラルネットワークを用いる．また，CNN の最終層にはクラス分類の場合はクラス数と同数のニューロン n 個を用意し，活性化関数には，式 2.9 で与えられるようなソフトマックス関数を用い

ることで，出力層のニューロンへの入力に対して，各クラスに対する確率 $p_1, \cdots, p_n$ を与える．

2.4.3　誤差逆伝搬法を用いた学習

CNN は局所受容野や共有重みという考え方によって，複雑な構造をしているが，ネットワークの形状としては多層で疎な結合を持ったフィードフォワード型ニューラルネットワークであるといえるため，一般的なフィードフォワード型のニューラルネットワークの学習方法と大きく変わらない．最終段の活性化関数を式 2.9 で与えられるソフトマックス関数とするため，j 番目のクラスに対応する確率 p_j が出力として得られる．あるサンプルにおける出力に対する誤差は，この出力と入力サンプルに対する理想的な出力 $t_1, \cdots, t_n$ の間の交差エントロピー

$$E = -\sum_{j=1}^{n} t_j \log p_j \tag{2.34}$$

で定義する．この誤差 E を最小化するように畳み込み層のフィルタ係数 $w_{ij}^{(c)}$ および，各ニューロンにおけるバイアス $b^{(c)(k)}$，判別部の全結合ニューラルネットワークにおける重みとバイアスを誤差逆伝搬法を用いて学習する．このときの重みの更新量 Δw_{ij}^{t+1} は更新前の重みを w_{ij}^{t} とすることで，

$$\Delta w_{ij}^{t+1} = -\eta \frac{\partial E^t}{\partial w_{ij}^t} + \alpha \Delta w_{ij}^t - \eta \lambda w_{ij}^t, \tag{2.35}$$

となる．ここで，第 1 項は勾配法による重み更新の項であり，η は学習率である．第 2 項はモメンタムであり，前回の更新量に乗じて加算することで更新量 Δw_{ij}^t が大きい間は修正量を大きくし，更新量が小さくなると修正量を小さくするという慣性を与えるパラメータとして学習の高速化を目的としている．第 3 項は重み減衰を実現する項であり，重みが極端に大きくならないように修正する正則化として働き，その作用の量を係数 λ にて調整する．

第 3 章

CNN-ELM モデル

本章では，前章で説明した CNN の最終段にある全結合層に ELM を接続したハイブリッドアーキテクチャニューラルネットワークである CNN-ELM について説明する．

3.1 CNN-ELM

CNN-ELM は，人間の視覚野を数理モデル化することで，2 次元画像における物体の移動やスケール変化に対して不変な特徴を抽出することを可能にしている CNN を事前学習として捉え，CNN の学習を行うことで設計される特徴を用いて ELM の学習を行うことで，CNN による判別に適した特徴の設計と ELM の高速で高い判別能力を組み合わせたハイブリッドアーキテクチャニューラルネットワークである．2015 年に Guo らにより手書き数字認識のベンチマークデータセットである MNIST dataset に対して適用され，CNN と精度に関して比較することで，CNN より高精度を示すことが報告されているが，CNN の学習パラメータやプーリング手法，その特性など詳細は明らかにされていない．図 3.1 に CNN-ELM の構成を示す．CNN は，誤差逆伝搬法による繰り返し処理によってパラメータを学習するが，ELM は一度の演算によりパラメータを学習する．そのため，CNN の全結合層を単純に ELM に変更するだけでは誤差逆伝搬法による学習は出来ず，まず CNN の学習により畳み込み層の重みを学習し，特徴設計が行われた全結合層における特徴を抽出する．抽出した特徴を入力として ELM の学習を行うことで，CNN-ELM 全体の学習を行う．したがって，CNN-ELM は，CNN を用いて入力画像から判別に有用な特徴を設計する CNN 部と設計された特徴を用いて判別を行う ELM 部に分けられる．

3.1.1 CNN 部の構成

CNN 部では，CNN の順伝搬処理を行うことで，全結合層において ELM の判別に用いる中間特徴を設計する．CNN の入力層は画像データのピクセル値が与えられる．したがって，画像認識を行う画像データのサイズに伴って，CNN の入力層のニューロン数は変化する．そのため，入力画像セットの画像サイズが不均一である場合などは幾何学的変換もしくはパディングによるサイズの均一化を行う必要がある．入力層の特徴マップのチャネル数は，入力画像の色チャネル数と等しくなる．したがって，MNIST database や Rectangles などのグレースケール画像データセットの 1 サンプルにおける最初の畳み込み層の処理は，式 2.31 より

$$u_{ij}^{(k)} = \sum_{(p,q)\in\mathcal{P}_{ij}} \left[x_{pq} w_{p-i,q-j}^{(k)} \right] + b^{(k)}, \tag{3.1}$$

となり，Caltech 101 などの RGB カラー画像では，

$$u_{ij}^{(k)} = \sum_{c=1}^{3} \left[\sum_{(p,q)\in\mathcal{P}_{ij}} \left[x_{pq}^{(c)} w_{p-i,q-j}^{(c)(k)} \right] + b^{(c)(k)} \right], \tag{3.2}$$

となる．畳み込みによって得られた特徴マップに対して，式 2.33 に従い，

$$y_{ij}^{(k)} = \max_{(p,q)\in\mathcal{P}_{ij}} x_{pq}^{(k)}, \tag{3.3}$$

で表される一定領域でプーリングを行うことで畳み込み層とプーリング層からなる 1 回分の処理となり，この処理を繰り返すことでスケールや位置に不変な特徴を設計し，全結合層の各ニューロンへの入力，

$$y_k = \sum_c \sum_{(i,j)} w_{kij}^{(c)} x_{ij}^{(c)}, \tag{3.4}$$

を中間特徴として抽出し，ELM 部への入力とする．ここで，$w_{kij}^{(c)}$ は各チャネルにおける特徴マップのニューロンから全結合層における k 番目のニューロンにかかる重みである．また，CNN は畳み込み処理とプーリング処理を繰り返すことで特徴マップを小さくしながら判別に有用な特徴を設計している．そのため，全結合層のニューロン数は，CNN の入力層におけるニューロン数より少なくし，少ないニューロンで元の情報の本質的な構造を表現させる．

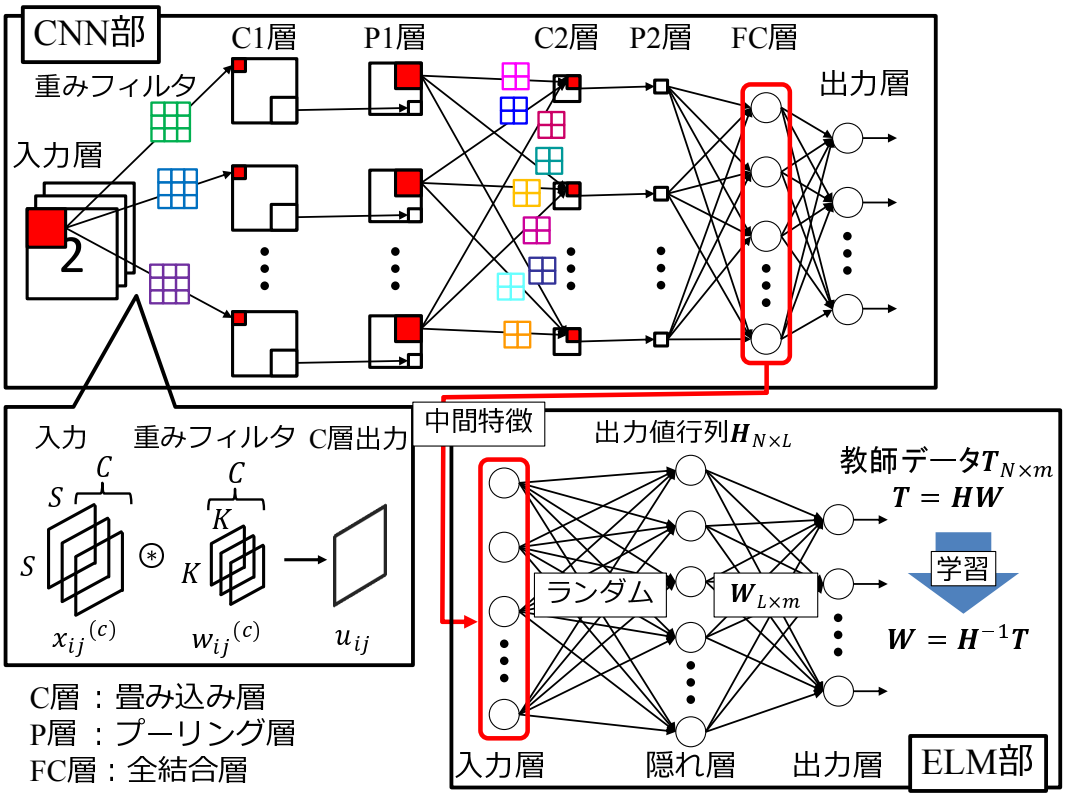

図 3.1　CNN-ELM の構成と学習動作の概要

3.1.2　ELM 部の構成

ELM 部では，CNN 部で抽出した中間特徴を入力として ELM の学習を行うことで，CNN-ELM の出力を求める．CNN 部で抽出された中間特徴を ELM 部の入力とするため，ELM の入力層のニューロン数は CNN 部の全結合層のニューロン数と等しくなる．したがって，ELM に入力される中間特徴は，画像データが CNN 部における事前学習により次元削減され，ELM に入力されると見なせる．また，ELM 部において高い汎化性能を実現するために隠れ層におけるニューロン数を入力次元数である入力層のニューロン数より十分に多く準備し，出力層は判別を行うクラスの数だけ準備する．ELM 部の出力層の k 番目のニューロンの出力は，抽出されたサンプル数 N の中間特徴を入力 $\mathbf{x}^n$ とすると，式 2.23 となり，この出力が CNN-ELM の出力となる．

3.1.3 CNN-ELM の学習

CNN-ELM の学習は，CNN 部による事前学習と事前学習により抽出された中間特徴を用いた ELM 部の学習に分けられる．CNN 部の学習は，通常の CNN と同様に式 2.13 に従った誤差逆伝搬法を用いて行われ，右辺第 2 項の重みの更新量は式 2.35 に従う．したがって，勾配法における重み更新式は，

$$w_{ij}^{t+1} = w_{ij}^{t} - \eta \frac{\partial E^t}{\partial w_{ij}^t} + \alpha \Delta w_{ij}^t - \eta \lambda w_{ij}^t, \tag{3.5}$$

となる．ELM の出力層における k 番目のニューロンの出力は，式 2.23 となるため，通常の ELM の学習と同様に，Moore-Penrose の疑似逆行列を用いた式 2.28 により隠れ層から出力層にかかる重みパラメータを求めることで，ELM 部の学習を行う．この CNN 部の学習と ELM 部の学習が，CNN-ELM 全体の学習となる．また，CNN 部と ELM 部をそれぞれ学習することで，CNN のメリットである人間の第一次視覚野の処理を模倣した効率的な特徴設計と十分な隠れ層のニューロン数による ELM の高い汎化性能を実現することが可能になる．ただし，学習時間については，同様のサイズの画像データセットを同様のパラメータで学習した場合，単純に ELM の学習分だけ CNN より長い時間が必要になる．しかし，CNN 部における誤差逆伝搬法を用いた繰り返し処理により精度を向上させて行くのに対して，中間特徴を用いた ELM の学習が少ない学習回数における CNN の低い学習精度を補うことで，少ない学習回数においても CNN-ELM 全体としては高い精度を実現し，単純に CNN を学習してある精度を実現するための学習時間より短い時間での学習が可能であることを 4 章に示す学習過程と学習時間に関する比較実験により示す．

第 4 章

CNN-ELM モデルの性能評価

本章では，CNN-ELM の性能を評価し，その特性を明らかにするために行った画像認識実験について説明する．性能の評価は，最高精度の比較と学習過程における精度の比較，学習時間についての比較とする．また，CNN-ELM の特性を明らかにするために，ノイズを加えた画像データセットを生成し，ノイズの強度に応じた認識精度の変化を比較する．

4.1 実験環境

本実験を行うにあたって用意した環境は表 4.1 のとおりである．CNN による学習は，Ubuntu 上にインストールした Deep Learning フレームワークである Caffe を用いる．Caffe は，C++と CUDA により記述されており，GPU による処理が含まれている．また，Python をインターフェースとして用いて特徴の抽出を行う．得られた特徴に対して，Python において ELM を実現するライブラリ Python-ELM と機械学習用ライブラリである scikit-learn を用いて ELM の学習を行うことで，CNN-ELM としての学習を行う．

4.2 画像データセット

本実験では，画像認識実験のための画像データセットとして MNIST（MNIST database），Rectangles，Caltech 101 を用いている．本項では，それぞれのデータセットと Caltech 101 に対して行った前処理について説明する．

表 4.1　実験に用いたハードウェア・ソフトウェア

OS	Ubuntu14.04(64bit)
CPU	Intel(R) Xeon(R) CPU X5675 @ 3.07GHz x2
GPU	NVIDIA Quadro K4200
メモリ	144GB
使用言語	Python 2.7.6 C++ CUDA 7.0
Python 機械学習ライブラリ	scikit-learn 0.13[16] Python-ELM 0.3[18]
Deep Learning フレームワーク	Caffe[17]

4.2.1　MNIST database

MNIST とは，アメリカ国立標準技術研究所（NIST）のデータセットである SD-1 と SD-3 から抽出され構築された図 4.1 に示すような 0 から 9 までの手書き数字に関するデータセットである [20]．ベンチマークデータとして CNN の性能評価だけでなく，多くの機械学習アルゴリズムの性能評価に用いられている．画像データは表 4.4 に示すようにサイズ 28×28 のグレースケール画像であり，各ピクセルは 0 から 255 までの値をとる．訓練データは 60,000 サンプル，テストデータは 10,000 サンプルであり，0 から 9 までの数字を判別する 10 クラス判別となる．

4.2.2　Rectangles

Rectangles とは，図 4.2 に示すような黒い画像に含まれている白い長方形の大小を判別するデータセットであり，MILA（Montreal Institute for Learning Algorithms）により提供されている [21]．各ピクセルは 2 値で 0 または 255 をとり，長方形の境界に対応する画素

図 4.1　MNIST database における 5 サンプルの画像

図 4.2　Rectangles におけるサンプルの画像 ([21] より引用)

は 255 の値を，それ以外の画素は 0 の値となる．データセット中の各データの長方形の高さと幅は一様に分布しており，特定の高さや幅の長方形が多いあるいは少ないということはない．また，高さと幅の差は 3 ピクセル以上であり，全てのデータにおいて四角形は長方形となっている（正方形ではない）．また，長方形はサイズ 28×28 の画像内に収まるようにサンプリングされている．訓練データは 1,200 サンプル，テストデータは 50,000 サンプルであり，長方形が大きいか小さいかを判別する 2 クラス判別となる．表 4.4 に詳細を示す.

4.2.3　Caltech 101

Caltech 101 とは，2003 年にカリフォルニア工科大学でコンピュータビジョンの研究や手法を容易にするために作成された一般物体認識向けの画像データセットである [19]．画像データは 101 個のカテゴリ（airplanes，faces，Motorbikes，pianos， etc.）にバックグラウンドカテゴリである BACKGROUND_google を加えた 102 個のカテゴリに分類される合計 9146 枚の RGB カラー画像となっており，各ピクセルは 0 から 255 までの数値をとる

チャネルが 3 つあり，その 3 つの値からなる．表 4.2 と表 4.3 はカテゴリごとの画像数を比較し，それぞれ上位 5 カテゴリと下位 5 カテゴリを示したものである．各カテゴリごとの画像数は偏っており，最も多いカテゴリで 800 枚，最も少ないカテゴリで 31 枚，ほとんどのカテゴリは 50 枚程度となっている．画像サイズは均一でなく全データを通じて，300×200 程度の画像となっている．

Caltech 101 に対する前処理

Caltech 101 の画像は画像サイズが不均一であり，CNN へ入力するために画像サイズを変換し，均一にする．画像サイズは CNN の入力層のニューロン数と等しくなるため，学習時間（計算コスト）を考慮し 100×100 のサイズへ縮小変換を行う．本実験では，カテゴリごとの画像数の偏りによる精度への影響を排除するためにカテゴリに属する画像数が近い airplanes カテゴリと Motorbikes カテゴリを用いた 2 クラス判別を行う．従って，airplanes カテゴリと Motorbikes カテゴリの総数 1598 枚の画像を 2 クラスの画像の比率が偏らないようにように訓練用データ 799 サンプルとテストデータ 799 サンプルを生成し，画像データセットとして用いる．また，CNN と CNN-ELM のノイズに対するロバスト性を明らかにするため，生成した画像データセットに対して，強度の異なる 2 種類のノイズを加えた，ノイズあり画像データセットを生成する．ノイズはガウスノイズを仮定し，画素の各 RGB 値 $\mathbf{x} = (x_g, x_b, x_r)$ に各チャネルで独立した分散 σ^2 を持つ 3 次元ガウスノイズ

$$f(\mathbf{x}) = \frac{1}{\sqrt{|\Sigma|\,(2\pi)^d}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right),$$

$$\mu = (0,0,0),\ \Sigma = \begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma^2 \end{bmatrix},$$

を加えたノイズ入りの画像セットもそれぞれ生成する．分散 σ^2 は，20，40 の 2 種類で実現する．従って，Caltech 101 では，表 4.4 に示すように画像サイズ 100×100，訓練データ 799 サンプル，テストデータ 799 サンプル，色チャネル数 3 となり，ノイズの強度を変化させた 3 種類の画像データセットに対して，airplanes カテゴリと Motorbikes カテゴリの 2

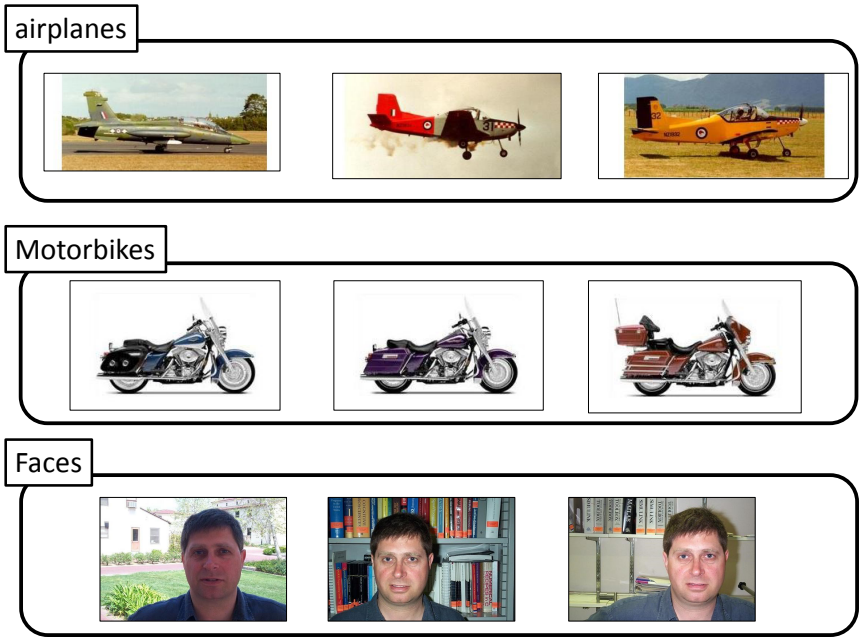

図 4.3　Caltech 101 におけるサンプルの画像

表 4.2　Caltech 101 の画像数が多い上位 5 カテゴリ

順位	カテゴリ名	枚数
1	airpalnes	800
2	Motorbikes	798
3	BACKGROUND_Google	468
4	Faces	435
5	Faces_easy	435

クラス判別を行う．CNN を用いたカラー画像の判別では画像から訓練データの平均画像を減算した値を入力として与える．そのため，Caffe 付属の平均画像作成用スクリプトを用い，各訓練データから平均画像を生成しておく．

表 4.3　Caltech 101 の画像数が多い下位 5 カテゴリ

順位	カテゴリ名	枚数
98	gerenuk	34
99	garfield	34
100	binocular	33
101	metronome	32
102	inline_skate	31

表 4.4　学習に用いる画像データセットの詳細

	MNIST	Rectangles	Caltech 101
画像サイズ	28×28	28×28	100×100
訓練データ数	60,000	1,200	799
テストデータ数	10,000	50,000	799
色チャネル数	1	1	3(RGB)
判別クラス数	10	2	2
ノイズ	なし	なし	なし，$\sigma^2 = 20$，$\sigma^2 = 40$

4.3　各データセットに適用する CNN の構造とパラメータ

表 4.4 に示すように 3 種類の画像データセットの画像サイズが異なっている．従って，学習を行う CNN の構造を画像データセットに合わせて変更する．Guo らの研究では，MNIST に対して，畳み込み層とプーリング層をそれぞれ 2 回繰り返す 7 層 CNN を用いて学習を行っている．しかし，プーリング層はダウンサンプリング処理と記述されているのみであり，平均プーリングと最大プーリングのようにどのようなダウンサンプリング手法を用いたか明確に記述されていない．また，CNN の学習における学習係数などのパラメータは説明されていない．そのため，MNIST を用いた実験では，Guo らの研究との比較を行うため

に，同一の構造を持つ CNN を構築し，明らかにされていない学習パラメータとプーリング手法については独自に設定する．Rectangles については，入力画像サイズが MNIST と同じ 28×28 であるため，判別クラス数と等価である出力層のニューロン数を変更し，その他は MNIST と同一に設定した CNN を構築する．Catech 101 については入力画像サイズが 100×100 と他の 2 つの画像データセットに比べて大きいため，CNN の構造と学習パラメータを含めて独自に設定する．画像データセットに適用するすべての CNN において，活性化関数は式 2.4 で示すシグモイド関数，プーリング処理は最大プーリングを用いる．

4.3.1　MNIST database に適用する CNN の構造とパラメータ

MNIST を用いて画像認識実験を行うために図 4.4 に示す入力層，C1 層，P1 層，C2 層，P2 層，FC 層，出力層からなる 7 層の CNN を構築する．入力層では，28×28 ピクセル 256 階調の入力画素を 0 から 1 に正規化して出力する．C1 層では，入力層の出力に対して，ストライド数 1 で 5×5 のフィルタを畳み込み，24×24 の特徴マップを生成する．P1 層では，C1 層の特徴マップに対して，ストライド数 2 で 2×2 のフィルタを用いたプーリングを行い，12×12 の特徴マップを生成する．C2 層では，ストライド数 1 で 5×5 のフィルタを畳み込み，8×8 の特徴マップを生成する．P2 層では，ストライド数 2 で 2×2 のフィルタを用いたプーリングを行い，4×4 の特徴マップを生成する．FC 層では，P2 層での出力を 24 個のニューロンに集約し，最後の出力層は 10 クラス判別を行うため，ニューロン数を 10 個に設定する．入力層から P2 層までの特徴マップのチャネル数は順に 1，6，6，12，12 として設定する．したがって，入力されるデータの次元数が 784 であるのに対し，FC 層で抽出された中間特徴は 24 次元に削減されることになる．学習を行う際の学習係数は 0.01，モメンタムは 0.9，重み減衰は 0.0005 にそれぞれ設定する．

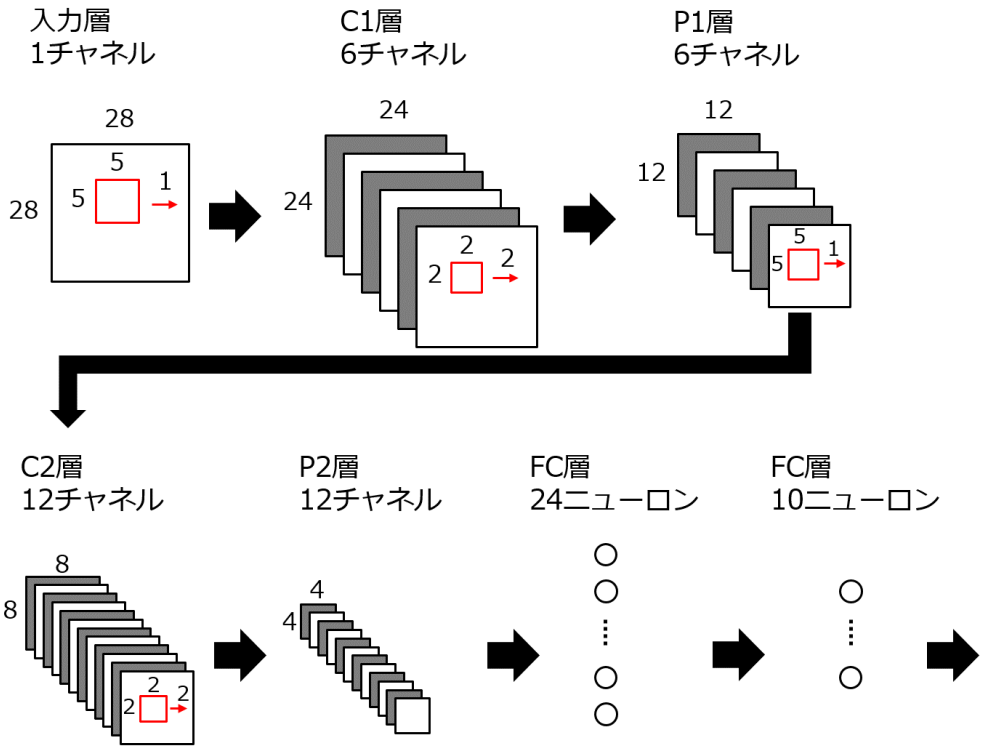

図 4.4　MNIST に適用する 7 層 CNN の構造

4.3.2　Rectangles に適用する CNN の構造とパラメータ

Rectangles は MNIST と同じ 28×28 ピクセルの入力画像となっているため，MNIST と同様の 7 層 CNN を構築する．ただし，MNIST とは異なり，2 クラス判別であるため，図 4.5 に示している出力層のニューロン数を 2 に変更した CNN を構築する．

4.3.3　Caltech 101 に適用する CNN の構造とパラメータ

Caltech 101 は入力画像のサイズが 100×100 ピクセルであり，MNIST や Rectangles と異なっている．したがって，CNN の構造を変化させながら独自に設定を行い，図 4.6 に示す入力層，C1 層，P1 層，C2 層，P2 層，C3 層，P3 層，FC 層，出力層からなる 9 層の CNN を構築する．入力層では，100×100 ピクセルで RGB256 階調の 3 チャネル画像を訓練データの平均画像で減算し，出力する．C1 層では，入力層の出力に対して，ストライド

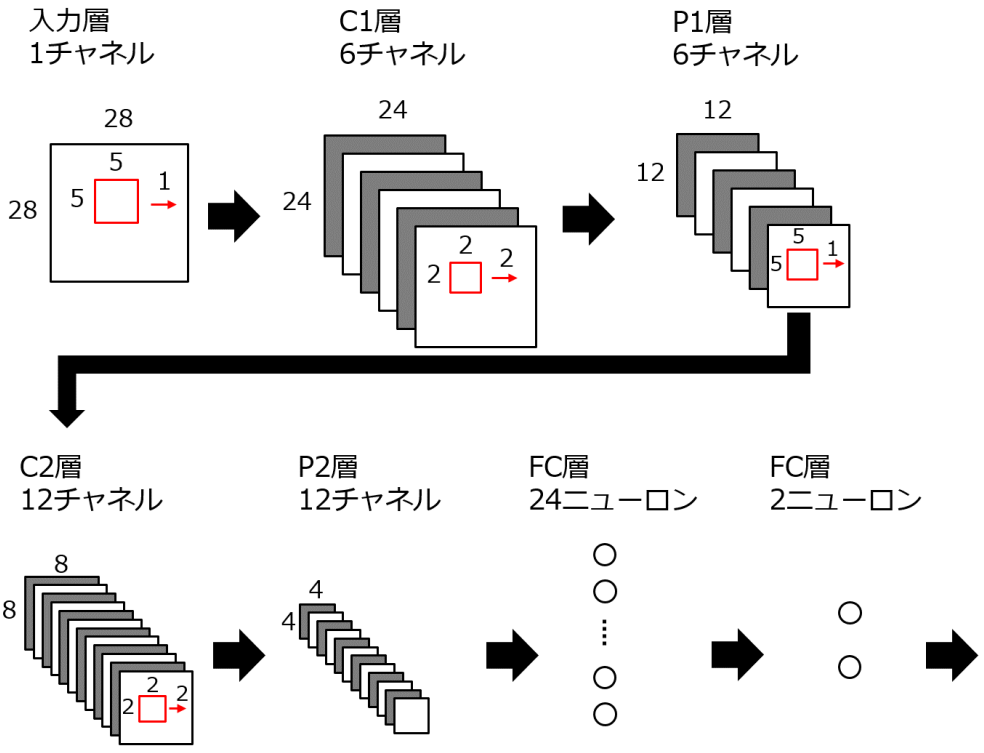

図 4.5　Rectangles に適用する 7 層 CNN の構造

数 3 で 7×7 のフィルタを畳み込み，32×32 の特徴マップを生成する．P1 層では，C1 層の特徴マップに対して，ストライド数 2 で 2×2 のフィルタを用いたプーリングを行い，16×16 の特徴マップを生成する．C2 層では，ストライド数 2 で 4×4 のフィルタを畳み込み，7×7 の特徴マップを生成する．P2 層では，ストライド数 1 で 2×2 のフィルタを用いたプーリングを行い，6×6 の特徴マップを生成する．C3 層では，ストライド数 2 で 2×2 のフィルタを畳み込み，3×3 の特徴マップを生成する．P3 層では，ストライド数 1 で 2×2 のフィルタを用いたプーリングを行い，2×2 の特徴マップを生成する．FC 層では，P3 層での出力を 1000 個のニューロンに集約し，最後の出力層は 2 クラス判別を行うため，ニューロン数を 2 個に設定する．入力層から P3 層までの特徴マップのチャネル数は順に 3，16，16，32，32，24，24 として設定する．したがって，入力されるデータの次元数が 10000 であるのに対し，FC 層で抽出された中間特徴は 1000 次元に削減されることになる．学習を行う際の学習係

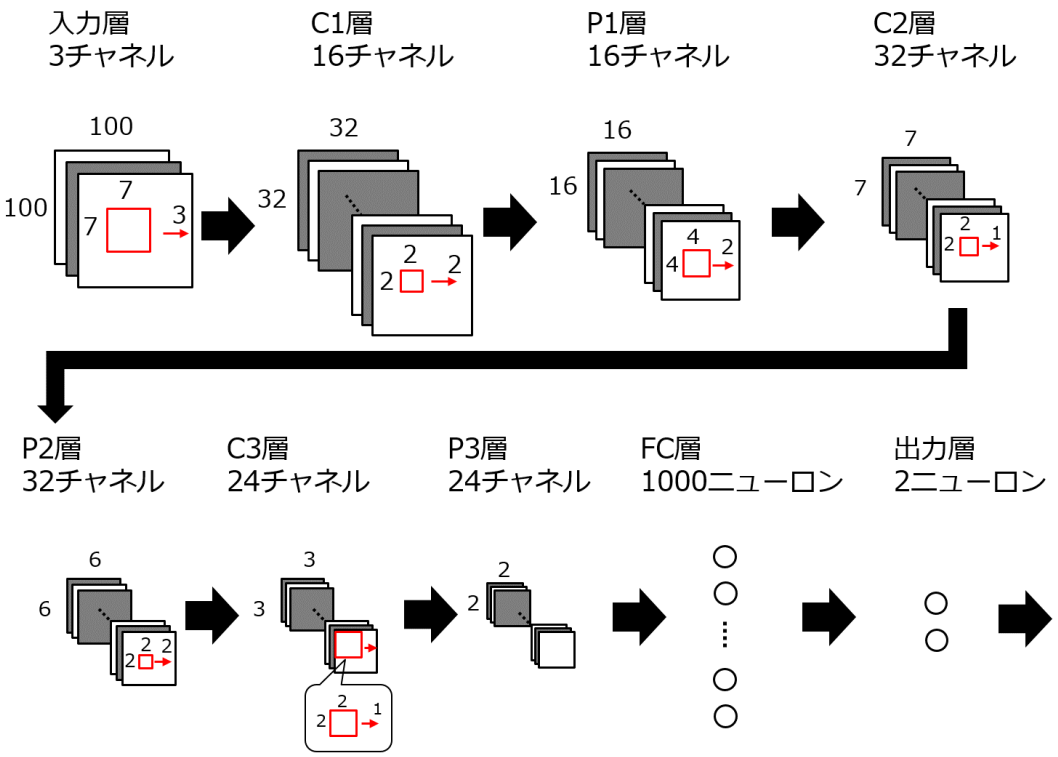

図 4.6　Caltech 101 に適用する 9 層 CNN の構造

数は 0.0001，モメンタムは 0.9，重み減衰は 0.0005 にそれぞれ設定する．

4.4　CNN-ELM と CNN の比較実験

CNN-ELM の性能とその特性を明らかにするために，実験環境上で CNN と CNN-ELM を用い，次の 3 種類の画像認識実験を行う．

- CNN-ELM と CNN の最高精度の比較
- CNN-ELM と CNN の学習過程における精度の比較
- CNN-ELM と CNN の学習時間の比較

画像認識実験では，構築した CNN により前処理を施した画像データセットを学習する．CNN の学習における訓練データ 1 回分の学習を 1 エポックとして，最大 200 エポックまで

学習を行う．本項では，3 種類の実験についてその目的と手順について説明する．

4.4.1　CNN-ELM と CNN の最高精度の比較

Guo らの実験では，CNN で取得した判別精度と CNN-ELM のベストパフォーマンスとなる判別精度の比較を行っており，MNIST において CNN-ELM は CNN に比べ，高精度に判別可能であることを示している．本実験においても同様の結果を再現することに加え，他のタスクにおいても優れていることを示すため，MNIST と Rectangles に関して CNN の判別精度と CNN-ELM のベストパフォーマンスとなる判別精度の比較を行う．まず，画像データセットを用いて事前学習として CNN の学習を 200 エポック行い，最も高い判別精度を示したエポックの精度を CNN の判別精度とし，そのパラメータセットをベストパラメータとして，中間特徴を抽出する．抽出した中間特徴を ELM の入力として与え，ELM の学習を行う．ELM の学習で取得したパラメータを用いて判別を行い，CNN-ELM の判別精度を取得し，CNN-ELM の判別精度と CNN の判別精度を比較する．MNIST における ELM の隠れ層のニューロン数は，1000 から 2900 まで 100 区切りの合計 20 条件に設定し，Rectangles では，ELM の隠れ層のニューロン数を 100 から 3000 まで 100 区切りの合計 30 条件に設定する．CNN-ELM の判別精度は，ELM における隠れ層のニューロン数ごとに ELM を 5 回学習し，取得する判別誤差の平均値である．

4.4.2　学習過程における判別精度の比較

CNN は誤差逆伝搬法を用いた学習を行うため，初期のランダムなパラメータでは低い判別精度となり，エポックを重ねるごとに重みが更新され判別精度が向上する．CNN-ELM は後段に ELM を接続しているため，少ないエポック数において CNN より高い精度を示す特性があると考えられる．したがって，Rectangles と Caltech 101 を用い，CNN の学習での各エポックにおける判別精度の推移とその際の CNN-ELM の判別精度の推移について比較する．また，ノイズを加えた Caltech 101 を利用することで，CNN と CNN-ELM の判別精

度がノイズから受ける影響について明らかにする．まず，画像データセットを用いて CNN の学習を 200 エポック行い，各エポックでのパラメータセットを用いて，中間特徴をそれぞれ抽出する．抽出した中間特徴を ELM の入力として与え，ELM の学習を行うことで，各エポックごとの判別精度を取得し，CNN と CNN-ELM の学習過程における判別精度を比較する．Rectangles は，ELM の隠れ層のニューロン数に 1000，2000，3000 の 3 条件を設定し，Caltech 101 では，ELM の隠れ層のニューロン数を 500 から 10000 まで 500 区切りの 20 条件に設定する．CNN-ELM の結果は，抽出した中間特徴を用いて ELM を 10 回学習し，取得する判別誤差の平均の値である．

4.4.3　CNN-ELM と CNN に関する学習時間の比較

CNN-ELM が少ないエポック数において，CNN より高い精度を示し，CNN-ELM を用い，ある一定の判別誤差を実現する際の学習時間が CNN より短いことを示すために，CNN の学習時間と CNN-ELM の学習時間を比較する．あるエポックにおける CNN-ELM の学習時間は CNN の学習時間と ELM の学習時間を合計した時間となる．したがって，CNN を同一エポック数だけ学習させるより長い学習時間が必要となるが，異なるエポック数において同一の精度を示す場合がある．CNN-ELM は少ないエポック数において CNN より高い精度を示すと考えられるため，CNN と CNN-ELM の学習時間を計測し，一定の判別誤差 0.05 を実現する最も早い学習時間について比較する．まず，学習過程における判別精度の比較実験を行う際に，CNN の 1 エポック分の学習時間を取得する．CNN は誤差逆伝搬法を用いており，誤差の更新箇所数は常に一定である．そのため，あるエポック数における CNN の学習時間はエポック数に 1 エポック分の学習時間を乗じた値とする．CNN の学習により取得した中間特徴に対して，ELM の隠れ層のニューロン数を 500 から 10000 まで 500 区切りで変化させた場合の学習時間を取得する．CNN-ELM は CNN のエポック数分の学習時間に ELM の学習時間を合計した値とする．

第 5 章

結果および考察

5.1 CNN-ELM と CNN の最高精度の比較

図 5.1 は，CNN の最高精度に対して，CNN-ELM における ELM の隠れ層のニューロン数を変化させた際の判別精度の比較である．CNN-ELM の判別誤差は，20 条件中 13 条件で CNN の判別誤差 0.0113 より高い精度を示しており，最も悪い精度でも判別誤差 0.0114 で CNN との差は 1.0×10^{-4} ポイントである．CNN-ELM の最高精度は，ELM の隠れ層のニューロン数 1600 における判別誤差 0.0110 であり，CNN の判別精度より 3.0×10^{-4} ポイント優れている．図 5.2 は Rectangles に関する同様のグラフである．CNN-ELM の判別誤差は，すべての条件において CNN の判別誤差 6.8×10^{-4} より劣っており，最も悪い精度は, ELM の隠れ層のニューロン数 200 における判別誤差 2.2×10^{-3} で CNN との差は 1.5×10^{-3} ポイントである．CNN-ELM の最高精度は，ELM の隠れ層のニューロン数 2300 における判別誤差 7.2×10^{-4} であり，CNN の判別精度より 0.4×10^{-4} ポイント劣っている．表 5.1 は，それぞれのデータセットにおける CNN の最高精度と CNN-ELM の最高精度であり，MNIST と Rectangles におけるそれぞれの判別誤差の差が小さいことを示している．

表 5.1　CNN-ELM と CNN に関する判別誤差の比較

	CNN	CNN-ELM(Best)
MNIST	0.0113	**0.0110**
Rectangles	**6.8×10^{-4}**	7.2×10^{-4}

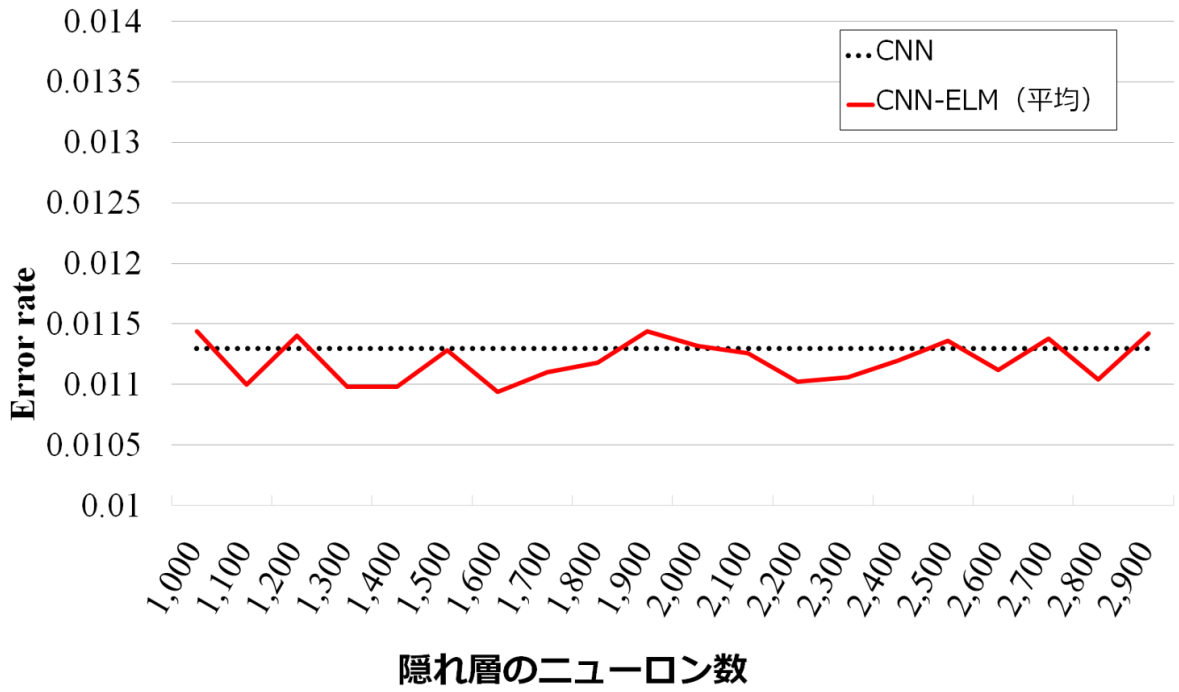

図 5.1　MNIST における CNN-ELM と CNN の判別誤差の比較

5.2　学習過程における精度の比較

図 5.3 は Rectangles における判別誤差の推移を示しており，20 エポックを超えると CNN-ELM と CNN の判別誤差はこれ以上の向上が見込めないほど，ほとんど 0 に近い値をとっているため，CNN と CNN-ELM は同程度の判別能力であることが分かる．Caltech 101 における CNN-ELM と CNN の判別誤差の推移を図 5.4 に示す．CNN-ELM は ELM の隠れ層のニューロン数を増やすことで明らかに精度が向上しているが，ELM の隠れ層のニューロン数 10000 までで，CNN より高い精度を実現することが出来ていない．しかし，少ないエポック数における判別精度では，ELM の隠れ層のニューロン数 500 から 10000 までの全ての条件において，CNN より高い精度を示している．また，CNN-ELM はエポックを重ねることで精度を向上させており，ELM の隠れ層のニューロン数 10000 では 200 エポッ

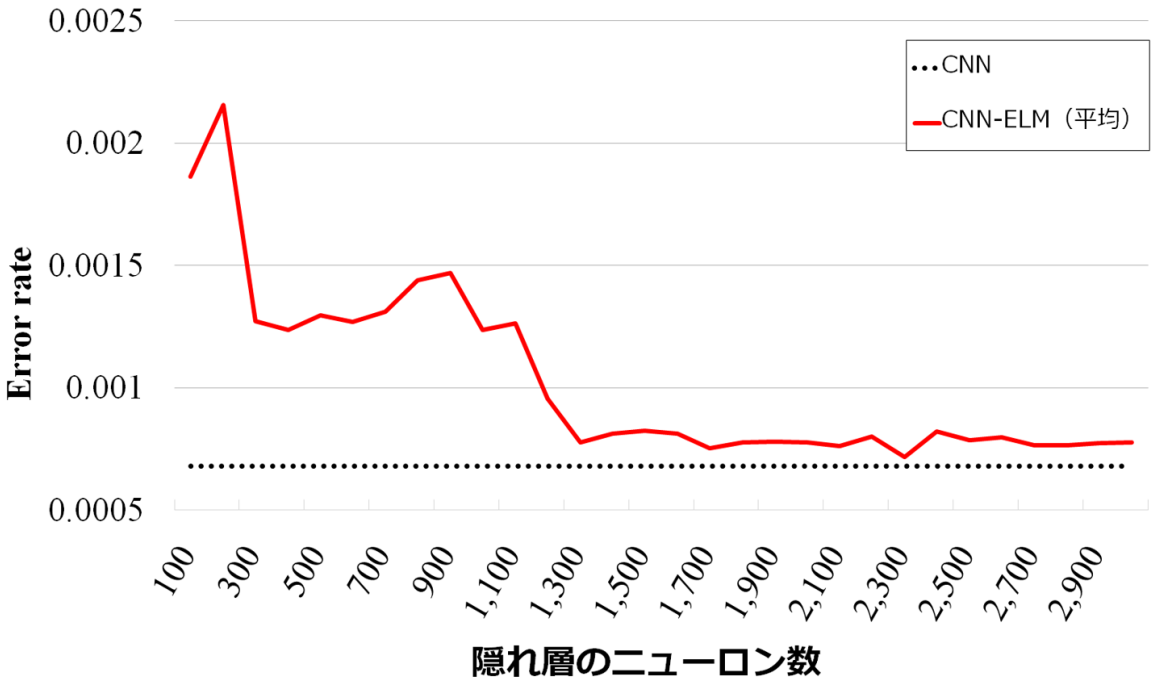

図 5.2　Rectangles における CNN-ELM と CNN の判別誤差の比較

クにおいて判別誤差 1.7×10^{-3} を示し，CNN の判別誤差である 8.6×10^{-4} に 8.4×10^{-4} ポイント差まで精度差が近づいている．図 5.5 は，$\sigma^2 = 20$ のノイズを含んだ Caltech 101 における判別誤差の推移を示している．CNN-ELM の判別誤差は，ELM の隠れ層のニューロン数 500 において CNN より劣っているが，ニューロン数 1500 で CNN と最終的な判別誤差が同程度となり，10000 においてどのエポックにおいても CNN より優れている．最終的な CNN-ELM の判別誤差は 2.3×10^{-2} であり，判別誤差 4.4×10^{-2} の CNN に比べて 2.1×10^{-2} ポイント優れている．図 5.6 は $\sigma^2 = 40$ のノイズを含んだ Caltech 101 における判別誤差の推移を示している．CNN-ELM は，$\sigma^2 = 20$ のノイズを含んだ Caltech 101 と同様に，ELM の隠れ層のニューロン数 500 では，CNN より劣っているが，2000 において最終的な判別誤差が同程度となり，10000 においてどのエポックにおいても CNN より優れて

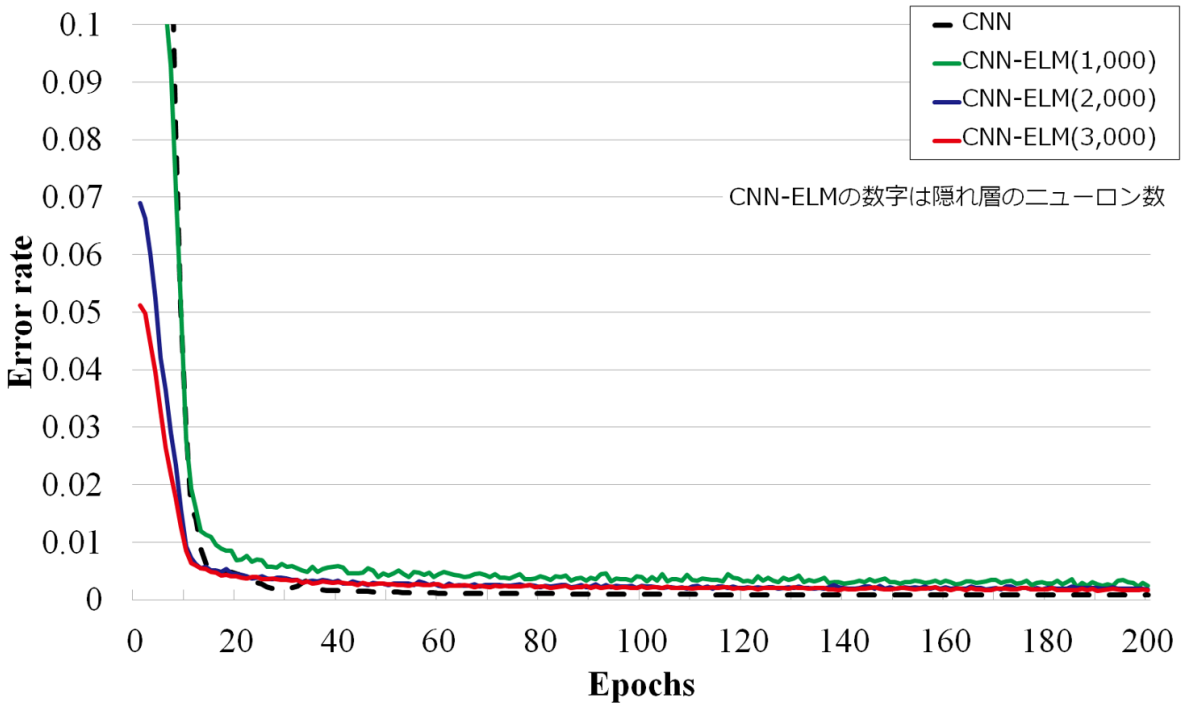

図 5.3　Rectangles における CNN-ELM と CNN の認識誤差の推移

いる．最終的な CNN-ELM の判別誤差は 2.7×10^{-2} であり，判別誤差 3.9×10^{-2} の CNN に比べて 1.2×10^{-2} ポイント優れている．また，図 5.7 は，ノイズを含んでいないオリジナルの Caltech 101 とノイズを含んだ 2 種類の Caltech 101 に関する CNN と CNN-ELM の判別精度の推移を示している．CNN は，オリジナルの Caltech 101 において最高の精度を示しているが，画像データセットがノイズの影響を受けることで判別精度が大きく悪化しており，200 エポックにおいて，$\sigma^2 = 40$ のノイズを含んだ Caltech 101 の判別精度はオリジナルの Caltech 101 の判別精度から 2.0×10^{-2} ポイント悪化している．これに対して，CNN-ELM は，4.5×10^{-3} ポイントの悪化となっており，CNN-ELM の判別精度は CNN-ELM に比べてノイズの影響を受けにくいことが分かる．

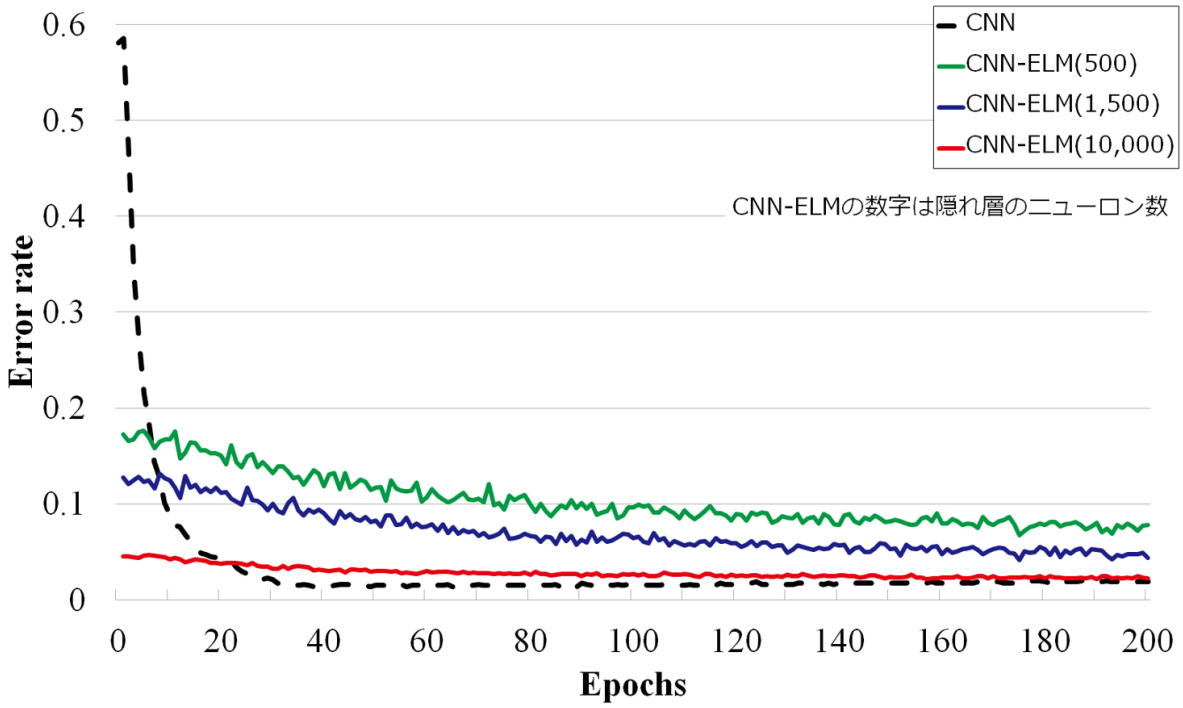

図 5.4　Caltech 101 における CNN-ELM と CNN の認識誤差の推移

5.3　CNN-ELM と CNN の学習時間の比較

図 5.8 に CNN-ELM を用いた Rectangles と Caltech 101 の学習を行う際に，隠れ層のニューロン数を変化させたときの ELM 部の学習時間の推移を示されているように，2 つの画像データセットにおける ELM の学習時間は，隠れ層のニューロン数の増加に対して線形的に増加していることが分かる．表 5.2 に CNN 部の 1 エポック分の学習時間を示す．Rectangles に対して判別精度 0.05 を実現する CNN のエポック数は図 5.3 より，10 エポックであり，45.40s の学習時間が必要になる．これに対して，ELM 部の隠れ層のニューロン数を 3000 とした CNN-ELM では，CNN 部で 2 エポックの学習を行い，ELM 部の学習を行うことで 15.36s で実現可能である．オリジナルの Caltech 101 に対して判別精度 0.05 を実現する CNN のエポック数は図 5.4 より，16 エポックであり，17.44s の学習時間が必要

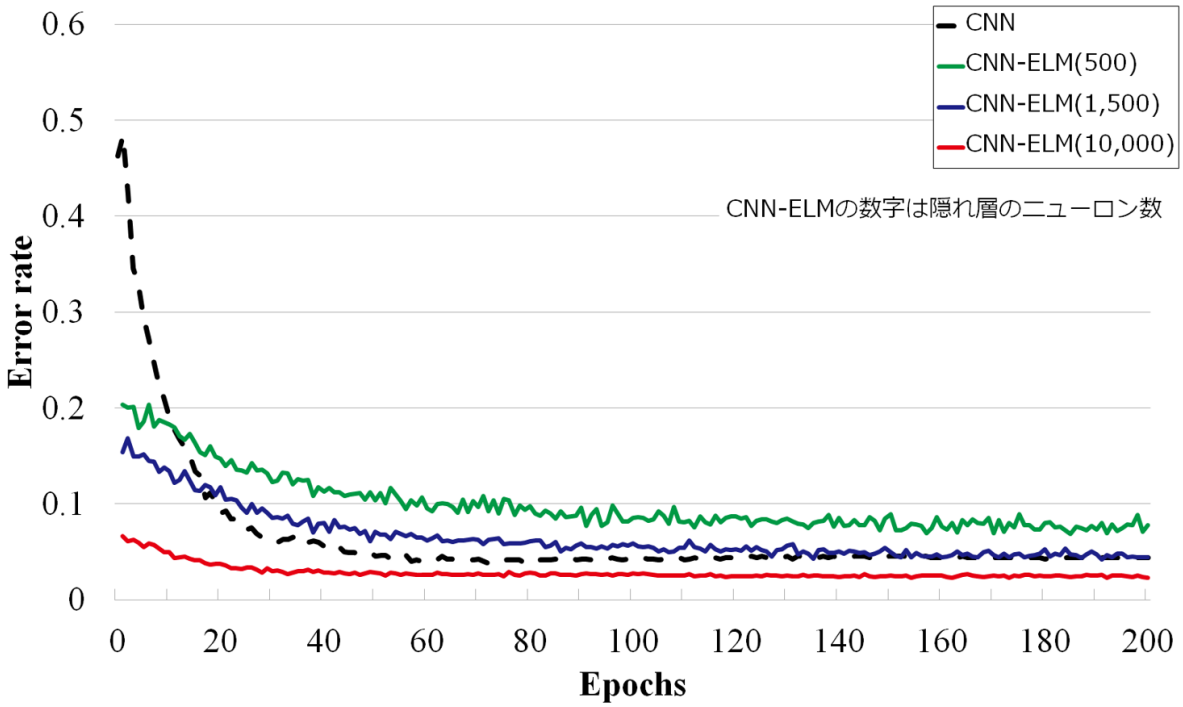

図 5.5　$\sigma^2 = 20$ のノイズを含んだ Caltech 101 における CNN-ELM と CNN の認識誤差の推移

になる．これに対して，ELM 部の隠れ層のニューロン数を 10000 とした CNN-ELM では，CNN 部で 1 エポックの学習を行い，ELM 部の学習を行うことで 10.92s で実現可能である．同様に，$\sigma^2 = 20$ のノイズを含んだ Caltech 101 において CNN は，45 エポック 49.05s の学習で判別精度 0.05 を実現するのに対し，CNN-ELM では CNN 部 9 エポックの学習により，19.64s で実現可能である．$\sigma^2 = 40$ のノイズを含んだ Caltech 101 において CNN は，31 エポック 33.79s の学習で判別精度 0.05 を実現するのに対し，CNN-ELM では CNN 部 1 エポックの学習により，10.92s で実現可能である．

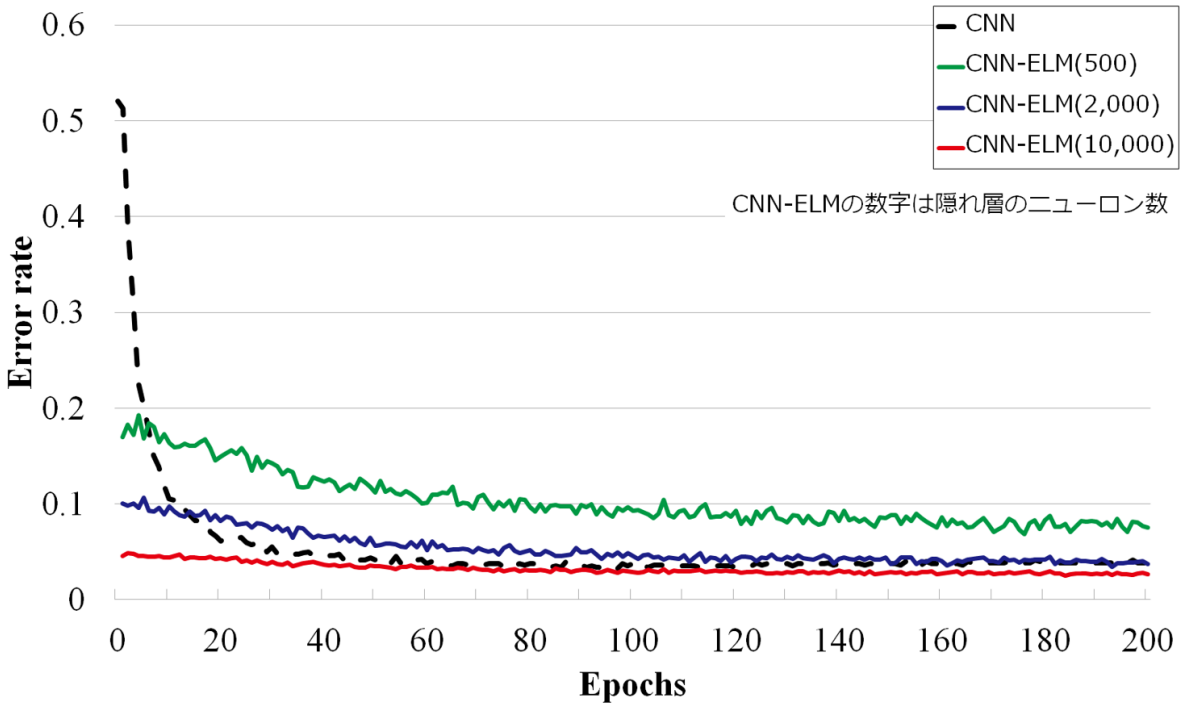

図 5.6　$\sigma^2 = 40$ のノイズを含んだ Caltech 101 における CNN-ELM と CNN の認識誤差の推移

5.4　考察

CNN-ELM と CNN の最高精度の比較実験では，MNIST と Rectangles において CNN-ELM と CNN の判別誤差の差が小さいことを示している．図 5.3 においても 2 つのアル

表 5.2　CNN の 1 エポックあたりの学習時間 [s]

	CNN
MNIST	1.58
Rectangles	4.54
Caltech101	1.09

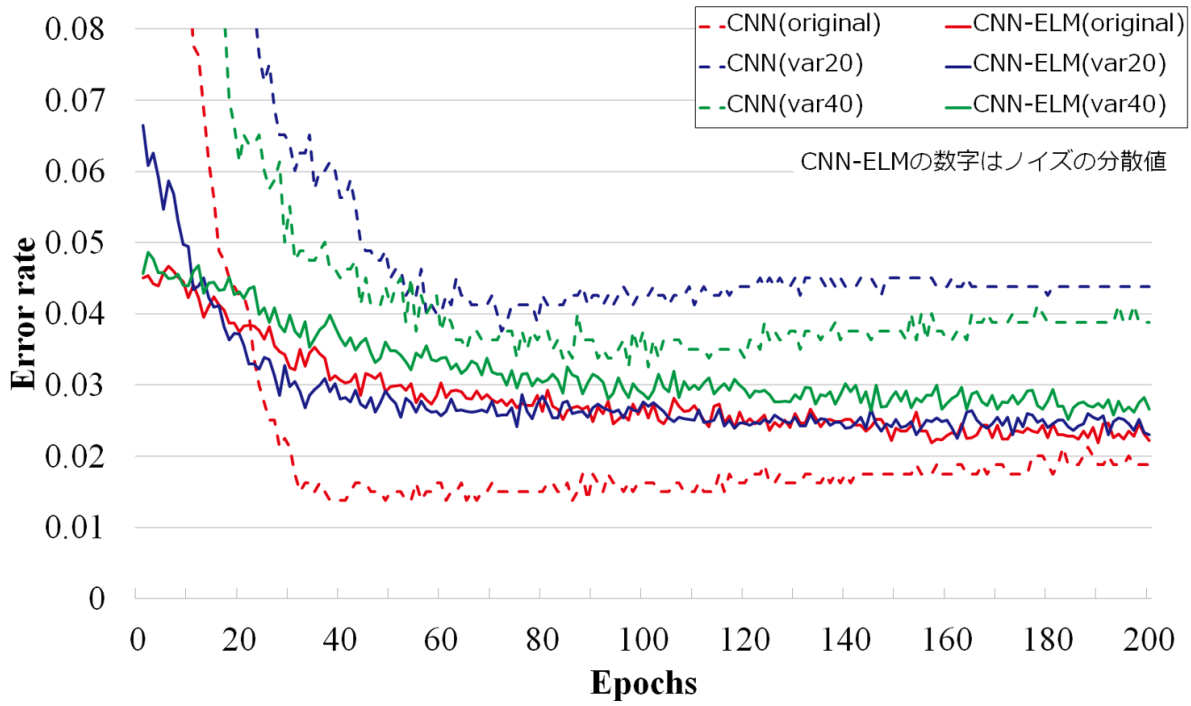

図 5.7　Caltech 101 にノイズを加えた際の CNN-ELM と CNN の判別誤差の推移

ゴリズムの精度の差は小さく，CNN 単体で十分に高い精度を示している．したがって，グレースケール画像を用いて手書き文字認識を行う MNIST と画素が 2 値の値をとる画像を用いて四角形の大小判別を行う Rectangles のタスクは Caltech 101 などのカラー画像を用いた一般物体認識タスクに比べて単純なタスクであり，CNN 単体で十分に高い精度が得られるため，CNN-ELM を用いても精度が向上しなかったと考えられる．

CNN-ELM と CNN の学習過程における精度の比較実験では，図 5.4 に示す通り，オリジナルの Caltech 101 において CNN-ELM は CNN より高い精度を実現することが出来ていないが，CNN の学習が収束しているのに対して，CNN-ELM の判別誤差が向上していることからエポック数を重ねることで CNN-ELM にとって判別しやすい特徴の設計が行われていると考えられる．また，CNN-ELM の ELM 部は隠れ層のニューロン数を変化させるこ

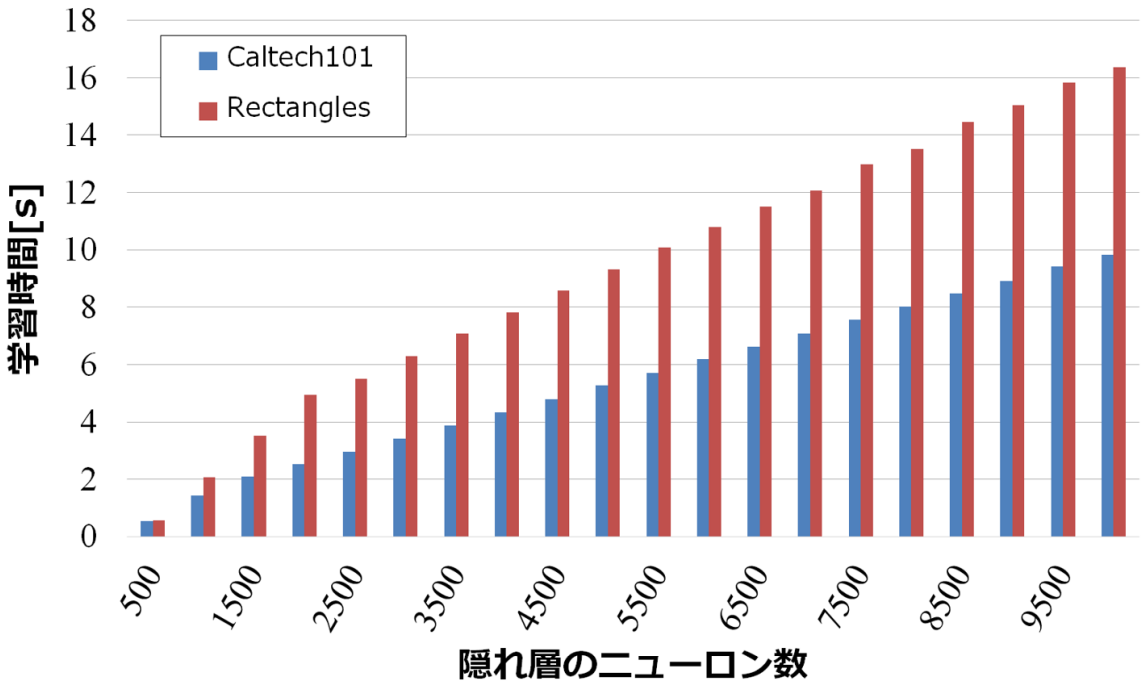

図 5.8　Rectangles と Caltech 101 における ELM 部の学習時間の推移

とで，判別精度が変化する．そのため，500 から 10000 まで 500 区切りとしていた隠れ層のニューロン数を最適な値とすることで，CNN-ELM は CNN より優れた判別誤差を実現できると考える．ノイズを含んだ Caltech 101 において CNN-ELM は CNN より高い精度を実現しているが，オリジナルの Caltech 101 と同様に CNN の判別誤差が収束しているのに対して，CNN-ELM は ELM 部の隠れ層のニューロン数を調整することでさらに精度を向上させることが出来ると考える．また，CNN の判別誤差がノイズに大きく影響を受けているのに対して，CNN-ELM 部があまり影響を受けていないことから，ELM 部のパラメータ調整により，CNN-ELM と CNN の判別誤差の変化の差が大きくなり，CNN-ELM はより顕著なノイズに対するロバスト性を示すと考えられる．

CNN-ELM と CNN の学習時間の比較では，判別誤差 0.05 を実現するための学習時間

を CNN-ELM と CNN で比較し，ELM 部の隠れ層のニューロン数を 10000 とした CNN-ELM が CNN より短い時間で学習可能であることを示した．これにより，最高精度ではなく，それより劣るある一定の判別精度を短い時間で実現することが求められる場合に，CNN-ELM は CNN より短い学習時間で目標の判別精度を達成できることを示している．また，CNN-ELM の学習時間は CNN 部の学習時間に ELM 部の学習時間を加算したものになり，ELM 部は隠れ層のニューロン数が少ないほど学習時間が短くなるため，ELM の隠れ層のニューロン数を調整することで，さらに学習時間を短くできると考えられる．CNN-ELM のパラメータ調整による学習時間の短縮とは別に，ELM 部で生成される出力値行列に着目し，一般か逆行列を算出する際に，その行列の特性を活かした計算の省略を行うことで，さらに学習時間を短縮できると考える．

第 6 章

結論

本研究では，画像認識の分野で高い判別精度を有していることが知られている CNN を事前学習として捉え，CNN の全結合層に 3 層に限定してニューラルネットワークの学習を高速化し，適切なニューロン数で高い汎化性能を示す ELM を接続したハイブリッドアーキテクチャニューラルネットワークである CNN-ELM モデルについて判別精度の向上と特性を明らかにするため，画像データセットを用いた画像認識実験を行った．その結果，判別精度に関しては，MNIST，Rectangles，オリジナルの Caltech 101 において CNN と同程度な判別精度を示し，$\sigma^2 = 20$ と $\sigma^2 = 40$ の 2 種類のノイズを含んだ Caltech 101 においてそれぞれ CNN より 2.1×10^{-2} ポイント 1.2×10^{-2} ポイント高い判別精度を示した．MNIST と Rectangles では，タスク自体が単純であるため，CNN 単体で十分に高い精度を実現できたと考えられ，Caltech 101 については ELM の隠れ層のニューロン数を最適化することでさらに精度を向上することが出来ると考えられる．また，ノイズが加えられることにより，CNN の判別精度が 2.0×10^{-2} ポイント悪化するのに対し，CNN の判別精度は，4.5×10^{-3} ポイントの悪化となり，CNN よりノイズの影響を受けにくいことが分かった．学習時間に関しては，少ないエポック数において CNN より高い判別精度を実現しており，$\sigma^2 = 20$ のノイズを含んだ Caltech 101 で判別精度 0.05 を実現するための学習時間として，CNN では 49.05s 掛かるのに対して，CNN-ELM は 19.64s と半分以下の学習時間で実現可能であることを示した．また，ELM の隠れ層のニューロン数を最適な値にすることや ELM の出力値行列に適した計算削減手法を考案することで，さらなる学習時間の短縮が可能であると考えられる．これらのことから，CNN-ELM は一般物体認識のような複雑なタスクでノイズが含まれることが分かっているようなデータを用いた判別に適していると言える．また，ある

一定の精度を短い学習時間で実現する際にも有用であると言える.

謝辞

本研究を行うにあたり，終始丁寧なご指導をしていただいた高知工科大学情報学群吉田真一准教授には心から感謝致します．学部生の頃に修士課程に進学したいと言った私のために多くの文献や勉強する機会を与えていただき，質問に伺った際は丁寧にご指導くださいました．また，2 回も連れて行ってくださった海外での発表を含めた国際会議では，英語での論文の執筆や発表などの大変貴重な経験をさせていただきました．計算知能以外の分野についても知見を広めるためのセミナーである ITNews では，新しい技術に関する技術のみならず，社会に出てからのエンジニアとしての大事な情報収集スキルを得られたと思います．吉田准教授の幅広い知識と研究者としての考え方は公私ともに学ぶところが多く，学士課程からの 4 年間で多くのことを学ばせていただき，一人間として成長できたと感じています．ありがとうございます．よく吉田准教授は,「40 歳が一番の働き時だから僕の年齢の人が頑張らないといけない」と仰っていましたが，くれぐれも無理をなさらないようにご自愛ください．吉田研究室で 4 年間を過ごせたことを深く感謝致します．また，本研究の副査を引き受けていただき，的確なご指摘をしてくださった高知工科大学情報学群の岩田誠教授と福本昌弘教授に感謝いたします．岩田教授には，研究の副査はもちろん，普段のセミナーでも大変お世話になりました．梗概や発表においてのご指摘では，研究について深く理解していただいた上で，方向性に関して助言をいただき，大変助かりました．ありがとうございます．福本教授は，研究室合同の行事に加えて，ネットワーク防災訓練など多くのイベントでお世話になりました．福本教授の鋭いご指摘に適切な回答が出来た自身はありませんが，頷きながらお話を聞いていただき，課題点や応用についてお話しいただいたおかげで，私一人では気づくことができなかった考察点に気づくことが出来ました．ありがとうございます．

吉田研究室の皆様にも大変お世話になりました．4 回生の皆さんは，卒業研究の時期が一緒だったことから結果が出なくて苦しい時や談笑した時など多くの時間を研究室で過ごしました．同じ時期に就職することになりますが，共に頑張りましょう．3 回生の皆さんは，互

いに支え合いながらソフトウェア工学の工程を進めている様子を見て，私も頑張ろうと思わせてくれました．とても頼りがいのある先輩になられると思います．特に，大学院進学する佐々木君と笹谷君，領内さんは，あと 2 年の猶予があると思いますので，多くのことを吸収して欲しいと思います．吉田研究室での 4 年間は既に卒業された先輩や同期生，後輩と一緒に過ごせたことで大変有意義であったと思います．深く感謝致します．ありがとうございました．

最後に，学士課程から含めて 6 年もの間，生活面で支えてくださった家族に心から感謝致します．

参考文献

[1] David E. Rumelhart, Geoffrey E. Hinton and Ronald J. Williams, "Learning representations by back-propagating errors," Nature, vol.323, pp.533-536, 1986.

[2] David H. Hubel and Torsten N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," The Journal of Physiology, vol.160, pp.106-154, 1962.

[3] Frank Rosenblatt, "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain," Psychological Review, vol.65, no.6, pp.386-408, 1958.

[4] Guang-Bin Huang, Lei Chen, and Chee-Kheong Siew, "Universal Approximation Using Incremental Constructive Feedforward Networks With Random Hidden Nodes," IEEE Trans. Neural Networks, vol.17, no.4, pp.879-892, 2006.

[5] Guang-Bin Huang, Qin-Yu Zhu and Chee-Kheong Siew, "Extreme Learning Machine: A New Learning Scheme of Feedforward Neural Networks," Proceedings of International Joint Conference on Neural Networks, vol.2, pp.985-990, 2004.

[6] Guang-Bin Huang, Qin-Yu Zhu and Chee-Kheong Siew, "Extreme Learning Machine: Theory and applications," Neurocomputing, vol.70, issues1-3, pp.489-501, 2006.

[7] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng and Trevor Darrell, "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition," arXiv 1310.1531v1 [cs.CV], 2013.

[8] Kunihiko Fukushima and Sei Miyake, "Neocognitron: A New Algorithm for Pattern Recognition Tolerant of Deformations and Shifts in Position," Pattern Recognition, vol.15, no.6, pp.445-469, 1982.

[9] Lili Guo and Shifei Ding, “A Hybrid Deep Learning CNN-ELM Model and Its Application in Handwritten Numeral Recognition,” Journal of Computational Information Systems, vol.11, no.7, pp.2673-2680, 2015.

[10] Marvin Minsky and Seymour A. Papert, “Perceptrons, Expanded Edition,” The MIT Press, 1969.

[11] Matthew D. Zeiler and Rob Fergus, “Visualizing and Understanding Convolutional Networks,” Lecture Notes in Computer Science(ECCV 2014), vol.8689, pp.818-833, 2014.

[12] Quoc B. Le, Marc’Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg S. Corrado, Jeff Dean and Andrew Y. Ng, “Building High-level Features Using Large Scale Unsupervised Learning,” Proceedings of 2013 IEEE International Conference on Acoustics, Speech, and Signal Processing(ICASSP 2013), 2013.

[13] Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner, “Gradient-Based Learning Applied to Document Recognition,” Proceedings of the IEEE, vol.86, issue 11, pp.2278-2324, 1998.

[14] 熊沢逸夫, “学習とニューラルネットワーク,” pp.21-34, 森北出版, 1998.

[15] 渡辺慧, “認識とパタン,” pp.90-105, 岩波新書, 1978.

[16] “Scikit-learn: machine learning in Python – scikit-learn 0.14 documentation,” `http://scikit-learn.org/stable/`.

[17] “Caffe,” `http://caffe.berkeleyvision.org`.

[18] “Extreme Learning Machines,” `http://www.extreme-learning-machines.org`.

[19] L. Fei-Fei, R. Fergus and P. Perona, “One-Shot Learning of object categories,” IEEE Trans. Pattern Recognition and Machine Intelligence, vol.28, no.4, pp.594-611, 2006.

[20] Yann LeCun, Corinna Cortes and Christopher J.C. Burges, “THE MNIST DATABASE of handwritten digits,” `http://yann.lecun.com/exdb/mnist/`.

[21] “Rectangles and Rectangles-images data,” http://www.iro.umontreal.ca/lisa/twiki/bin/view.cgi/Public/RectanglesData.