勾配型学習の文書認識への応用

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner

概要

バックプロパ学習アルゴリズムで学習させた多層ニューラルネットは、勾配型学習の成功例である。適切なネットワークアーキテクチャがあれば、勾配学習アルゴリズムを用いて、最小限の前処理で手書き文字のような高次元パターンを分類できる複雑な決定面を合成することが可能である。本論文では、手書き文字認識に適用される様々な手法をレビューし、標準的な手書き数字認識タスクで比較する。その結果、2次元形状の多様性に対応するために特別に設計された畳み込みニューラルネットワークが、他のすべての手法よりも優れていることが示された。

実際の文書認識システムは、分野抽出、分割、認識、言語モデリングなどの複数のモジュールで構成されている。グラフ変換ネットワーク(GTN)と呼ばれる新しい学習パラダイムは、このようなマルチモジュールシステムを勾配に基づく手法でグローバルに学習させ、全体のパフォーマンス指標を最小化することを可能にするものである。

オンライン手書き文字認識のための2つのシステムについて説明する。実験では、グローバルな学習の利点と、グラフ変換ネットワークの柔軟性が示された。また、銀行小切手を読み取るためのGraph Transformer Networkについても述べる。このシステムでは、畳み込みニューラルネットワークの文字認識器と、グローバルな学習技術を組み合わせて使用している。は、ビジネスおよび個人向け小切手の記録精度を提供します。商業的に展開されており、1日あたり数百万件のチェックを読み取ることができます。

キーワード：ニューラルネットワーク、OCR、文書認識、機械学習、勾配学習、会話型ニューラルネットワーク、グラフ変換器ネットワーク、有限状態変換器。

呼称 ₑGT

グラフトランス。
ₑGTN Graph transformer network.
ₑHMM Hidden Markov model.
ₑHOS Heuristic oversegmentation
ₑK-NN K-nearest neighbor.
ₑNN ニューラルネットワーク。
ₑ OCR 光学的文字認識 ₑ PCA
主成分分析 ₑRBF 半径基底関数
ₑRS-SVM 還元集合サポートベクトル法 ₑ
SDNN
空間変位ニューラルネットワーク。
ₑSVM サポートベクトル法。
ₑTDNN Time delay neural
network（時間遅延型ニューラルネットワーク）。
ₑV-SVM 仮想サポートベクトル法。

著者らは、AT&T Labs- Research, 100 Schulz Drive Red Bank, NJ 07701のSpeech and Image Pro- cessing Services Research Laboratoryに所属している。電子メール：fyann,leonb,yoshua,haffnerg@research.att.com. また、Yoshua Bengioは、Département d'Informatique et de Recherche Opérationelle, Université de Montréal, C.P. 6128 Succ.Centre-Ville, 2920 Chemin de la Tour, Montréal, Québec, Canada H3C 3J7に在籍。

I.はじめに

ここ数年1、機械学習技術1、特にニューラルネットワーク1が、パターン認識システムの設計においてますます重要な役割を果たすようになってきた。実際1、連続音声認識や手書き文字認識などのパターン認識アプリケーションの近年の成功には、学習技術の利用が不可欠であったと言える。

本論文の主旨は、自動的な学習1に依存することで、より優れたパターン認識システムを構築することができ、手作業で設計したヒューリスティックを減らすことができるというものである。これは、近年の機械学習とコンピュータ技術の進歩により可能となった。文字認識を例にとり、手作業で行っていた特徴抽 出 を 、 ピクセル画像を直接操作する注意深く設計された学習機械に置き換えることで、有利になることを示す1。また、文書理解を事例として、個別に設計されたモジュールを手動で統合して認識システムを構築する従来の方法を、グローバルな性能基準を最適化するためにすべてのモジュールを学習できるグラフ変換ネットワーク1という統一的かつ原理的な設計パラダイムに置き換えることができることを示す1。

パターン認識の初期段階から、自然界に存在するデータ1（音声、文字、その他のパターン）の多様性と豊かさにより、人手で正確な認識システムを構築することはほとんど不可能であることが知られている1。そのため、ほとんどのパターン認識システムは、自動学習技術と手作業で作成されたアルゴリズムを組み合わせて構築されている1。通常、個々のパターンを認識する方法は、システムを図1のような2つの主要なモジュールに分割して構成する。最初のモジュール1は特徴抽出器と呼ばれ、入力パターンを低次元のベクトルや短い記号列で表現できるように変換し、（a）容易に照合・比較でき1、（b）入力パターンの変形や歪みに対して、その性質を変えずに比較的不変であることを特徴とする。特徴抽出器は、事前知識のほとんどを含み、タスクに特化したものである。また，多くの場合，完全に手作業で作成されるため，設計労力1の大半を占める。一方，分類器1は汎用的で学習可能です．このアプローチの主な問題点は、認識精度が、設計者が適切な特徴量のセットを考え出す能力によって大きく左右されることである。これは非常に困難な作業であり、残念ながら1新しい問題のたびにやり直さねばならない。パターン認識に関する多くの文献は、特徴量の相対的な比較に費やされ ている。

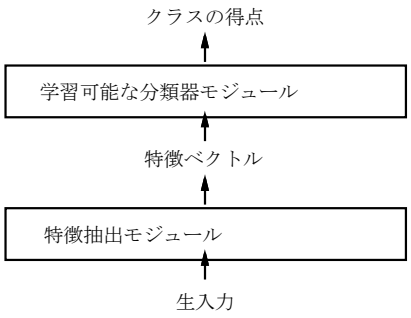

図1.従来のパターン認識は、固定特徴抽出器と学習可能な分類器という2つのモジュールで行われる。

特定のタスクに対するさまざまな機能セットのメリット。

歴史的に1、適切な特徴抽出器の必要性は、分類器が使用する学習技術が、容易に分離可能なクラスを持つ低次元空間に限定されていたことに起因する1]。この10年間、3つの要因が重なり、このビジョンは変化してきた。1つ目は、高速な演算装置を備えた低価格のマシンが登場し、アルゴリズムの改良よりも総当り的な「数値的」手法に頼ることができるようになったことである。第2に、手書き文字認識など、市場規模が大きく関心も高い問題1について、大規模なデータベースが利用可能になったことで、設計者は認識システムの構築にあたって、手作業による特徴抽出を減らし、実データに依存することができるようになったことである。第三に、高次元の入力に対応し、大規模データから複雑な決定関数を生成できる強力なマ-チン学習技術が利用可能になったことが非常に重要である。近年の音声・手書き文字認識システムの精度の向上は、学習技術と大規模な学習データセットへの依存度が高まったことに大きく起因していると言える。この事実を裏付けるように、最近の商用OCRシステムの多くは、バックプロパゲーションで学習させた多層ニューラルネットを使用しています1。

本研究では、手書き文字認識の課題を検討し（第I節、第II節）、手書き数字認識のベンチマークデータセットにおいて、いくつかの学習技術の性能を比較する（第III節）。より自動的な学習は有益であるが1、どのような学習手法もタスクに関する最小限の事前知識なしには成功し得ない。多層神経回路網1の場合、知識を取り入れる良い方法は、そのアーキテクチャをタスクに合わせて調整することである。セクションIIで紹介した畳み込みニューラルネットワーク2]は、局所的な接続パターン1や重みに制約を与えることによって、2次元形状の不変性に関する知識を取り入れた特殊なニューラルネットワークアーキテクチャの一例である。第III部では、孤立手書き文字認識のためのいくつかの手法の比較を示す。第IV部では、個々の文字認識から文書中の単語や文の認識1に至るまで、複数のモジュールを組み合わせて全体の誤差を小さくする考え方を紹介している。手書き文字のような可変長のオブジェクトをマルチモジュールで認識する場合、各モジュールが以下の条件を満たすことが重要である。

は有向グラフを操作することができる。これは、セクション IV で紹介する学習可能なグラフ変換ネットワーク (GTN) の概念につながるものである。セクションVでは、単語や文字列を認識するためのヒューリスティックなオーバーセグメンテーションという古典的な方法について述べる。セクションVIでは、手動によるセグメンテーションやラベリングを必要とせず、単語レベルで認識器を学習するための判別型および非判別型の勾配ベース技術が紹介されている。セクションVIIでは、入力上のすべての可能な位置で認識器をスキャンすることにより、セグメンテーションヒューリスティックの必要性を排除する、有望な空間置換ネットワーク（Space-Displacement Neu-Ral）アプローチを示す。セクションVIII1では、学習可能なグラフ変換ネットワークが、一般的なグラフ構成アルゴリズムに基づく複数の一般化変換1として偽造できることが示されている。また、音声認識でよく用いられる隠れマルコフモデル1 とGTNとの関連も扱う。第IX節では、ペン型コンピュータに入力された手書き文字を認識するためのグローバルに学習されたGTNシステムについて説明する。この問題は「オンライン」手書き認識1 として知られており、ユーザーが書いたものを機械が即座にフィードバックしなければならないからである。このシステムの中核は、畳み込みニューラルネットワークである。この結果は、認識器をあらかじめセグメント化1 された手書きラベル1 の孤立文字でトレーニングするのではなく、単語レベル1 でトレーニングすることの利点を明確に示している。セクションXでは、手書きと機械印刷の銀行小切手を読むためのGTNベースの完全なシステムについて説明する。このシステムの中核は、セクションIIで説明したLeNet-5と呼ばれる畳み込みニューラルネットワークである。このシステムは、NCR社の銀行向け小切手認識システムとして商業的に利用されている。このシステムは、全米のいくつかの銀行で、毎月数百万枚の小切手を読み取っている。

A. データからの学習

機械学習の自動化にはいくつかのアプローチ1 があるが、近年ニューラルネットのコミュニティ1 で盛んに行われている最も成功したアプローチ1 は、「数値的」あるいは勾配に基づく学習と呼ぶことができる。学習機械は関数 $Y^p = F(Z^p, W)$ を計算し、Z^p は p 番目の入力パターン1、W はシステムの調整可能なパラメータの集合を表す。パターン認識の設定1 では、出力Y^p は、パターンZ^p 1の認識されたクラスラベルとして、または各クラスに関連するスコアまたは確率として相互予見されるかもしれません。損失関数 $E^p = D(D^p, F(W, Z^p))$1 は、D^p 1 パターン Z^p 1 に対する「正しい」または望ましい出力と、システムによって生成された出力との間の不一致を測定するものである。平均損失関数 $E_{train}(W)$ は、学習セット $f(Z^1, D^1), ...(Z^P, D^P)g$ と呼ばれるラベル付き例集合上の誤差 E^p の平均値である。最も単純な設定1 では，学習問題は $E_{train}(W)$ を最小化する W の値を見つけることである．実際には，学習セットに対するシステムの性能はあまり重要ではない．より重要な指標は，システムが実際に使用される現場1 でのエラー率である．この性能は，テスト集合と呼ばれる訓練集合から切り離されたサンプル集合1上での精度を測定することによって推定される．多くの理論的・実験的研究3]1 4]1 5]により，以下のことが示されている．

テストセット E_{test} で期待されるエラー率とトレーニングセット E_{train} で期待されるエラー率との間のギャップは，トレーニングサンプルの数によって，およそ次のように減少することがわかる．

$$E_{test} - E_{train} \; k(h/P)^{a} \quad (1)$$

ここで，P は学習サンプルの数1 h は「有効容量」または機械の複雑さの尺度6]1 7]1 o: は0.5 から 1.01 の間の数で，k は定数である。このギャップは、学習サンプル数が増加すると必ず減少します。さらに1 容量hが増加すると1 E_{train} は減少する。したがって、容量 h1 を増やすと、E_{train} の減少量と学習サンプル数の減少量との間にトレードオフが生じます。

は、最小の汎化誤差 E_{test} を達成する容量 h の最適値で、gap1 を増加させる。

学習アルゴリズムは、E_{train} と、ギャップの推定値を最小化しようとする。この正式なバージョンは構造的リスク最小化6]1 7]1 と呼ばれ、各サブセットが前のサブセットのスーパーセットであるようなパラメータ空間のサブセットのシーケンスに対応する、容量増加1 の学習マシンのシーケンスを定義することに基づいています。構造的リスク最小化は、E_{train} + ;3H(W)1 を最小化することによって実行される。ここで、関数 H(W)は正則化関数1 と呼ばれ、 ;3 は定数と呼ばれる。H(W) は、パラメータ空間の高容量部分集合に属するパラメータ W に対して大きな値を取るように選択される。H(W) を最小化することにより，アクセス可能なパラメータ空間の部分集合の容量を制限し1，学習誤差の最小化と，学習誤差とテスト誤差の期待値のギャップの最小化との間のトレードオフを制御する．

B. 勾配に基づく学習

コンピュータサイエンスにおける多くの問題の根底には、パラメータのセットに対して再スペクトを持つ関数を最小化する一般的な問題がある。勾配に基づく学習は、一般に離散（組み合わせ）関数よりも合理的に滑らかな1連続関数を最小化する方がはるかに簡単であるという事実に基づいている。損失関数は、パラメータ値の小さな変動が損失関数に与える影響を推定することで最小化できる。これはパラメータ値に対する損失関数の勾配によって測定される。勾配ベクトルが解析的に（摂動による数値計算と同様に）計算できるようになると、効率的な学習アルゴリズムが考案される。これが連続値のパラメータを持つ多くの勾配学習アルゴリズムの基礎となっている。本稿で述べる手順1 では、パラメータW Newton法やQuasi-Newton法と同様にHessian行列を使用します。共役勾配法8]も使用可能である。しかし1 付録B によれば、文献上では多くの主張がなされているが1 、これらの2次法の大規模学習機に対する有用性は非常に限られている。

一般的な最小化手法として、確率的勾配アルゴリズム1 があり、オンライン更新とも呼ばれる。これは，平均勾配のノイズ版1 または近似版1 を用いて，パラメータベクトルを更新するものである．このアルゴリズムの最も一般的な例1 では，W は1つのサンプルに基づいて更新される．

$$W_k \; W_{k-} \; - \; E \; \frac{8E^{p\,k}(W)}{8W} \quad (3)$$

の集合は実数値のベクトル1 であり、E(W) は連続1であり、かつほぼどこでも微分可能である。このような設定における最も単純な最小化手順は、勾配降下アルゴリズムであり、Wは以下のように反復的に調整される。

$$\frac{8E(W)}{}$$

この手順では、パラメータベクトルが変動する
しかし、音声認識や文字認識のような冗長なサンプルを含む大規模な学習セットでは、通常の勾配降下法や2次勾配法よりもかなり高速に収束する。この理由については付録B で説明する。このようなアルゴリズムの学習への応用は1960年代から理論的に研究されてきた9]1 10]1 11]1 が、非自明な課題に対する実用的な成功は80年代中頃までなかった。

C. 勾配逆伝播法

勾配ベース学習法は1950年代後半から使われているが1,
そのほとんどはリニアイヤーシステムに限定されていた1]。このような単純な勾配降下法が複雑な機械学習課題に対して驚くほど有用であることは、次の3つの出来事が起こるまで広く認識されることはなかった。最初の出来事は、初期の警告にもかかわらず、損失関数のローカルミニマムの存在は、実際には大きな問題ではないようであることがわかったことで

$$W_k W_{k-} - E\frac{}{8W}\cdot \qquad (2)$$

最も単純な場合1 E はスカラー定数である。より高度な手続きでは、変数 E1 を使用するか、対角行列1 に置き換えるか、逆行列の推定値に置き換えます。

ある[12]1。これは、ボルツマンマシン13]1 14]のような初期の非線形勾配ベースの学習技術の成功に、ローカルミニマムが大きな障害になっていないように思われることに気づいたときに明らかになった。第二の出来事は、Rumelhart1 Hinton and Williams 15] などによって、数層の処理からなる非線形システムにおいて勾配を計算するためのシンプルで効率の良い手順1 back-propagation al- gorithm1 が一般化されたことである。第3の出来事は、バックプロパゲーション法をシグモイド単位を持つ多層ニューラルネットに適用することで、複雑な学習課題を解決できることを示したことである。バックプロパゲーションの基本的な考え方は、出力から入力への伝搬によって勾配を効率的に計算することである。この考え方は60年代初頭の制御理論の文献に記載されていたが16]1、機械学習への応用は当時は一般に認識されていなかった。興味深いことに1、ニューラルネットワーク学習の文脈におけるバックプロパゲーションの初期の導出は、勾配1 ではなく、「仮想タ-」を用いた。

中間層17]1 18]1 または最小層のユニットを「get」。擾乱の議論 19]。使用されるラグランジュ形式は制御理論の文献にあるバックプロパゲーション20]1 を導き出し、バックプロパゲーションのリカレントへの一般化を導き出すための、おそらく最も厳密な方法を提供します。

ネットワーク21]1
や異種モジュールのネットワーク22]がある。一般的な多層システムに対する簡単な導出は第I-
E節で行う。

多層ニューラルネットでローカルミニマムが問題にならないのは、理論的にやや謎である。これは、ネットワークがタスクに対して大きすぎる場合（実際には通常そうである）1、パラメータ空間に「余分な次元」が存在することで、到達不可能な領域のリスクが減少すると推測されている。バックプロパゲーションは、ニューラルネットワークの学習アルゴリズムとして最も広く用いられており1、おそらくあらゆる形式の学習アルゴリズムの中で最も広く用いられているものである。

D.実際の手書き文字認識システムにおける学習

分離型手書き文字認識は、従来から手書き文字認識

システムとして開発されてきた。
を参照)1
、ニューロネットの初期の成功例の一つである25]
。手書き数字の認識に関する比較実験をセクションIII
で報告する。その結果、勾配に基づく学習で学習させたニューラルネットワークは、同じデータでテストした他のすべての手法よりも優れた性能を発揮することがわかった。畳み込みニューラルネットワーク1と呼ばれる最も優れたニューラルネットワークは、ピクセル画像から直接関連する特徴を抽出するよう学習するように設計されている（セクションIIを参照）。

しかし、手書き認識における最も難しい問題の1つは、個々の文字1
を認識するだけでなく、単語や文の中の隣接する文字1
を分離することである。このための手法として、「ヒューリスティック・オーバーセグメンテーション」と呼ばれるものが「標準」となっている。これは、ヒューリスティックな画像処理技術1
を用いて文字間のカットの候補を多数生成し、その後、認識器によって各候補文字に与えられたスコアに基づいて、カットの最適な組み合わせを選択するものである。このようなモデル1では、システムの精度は、ヒューリスティック1により生成されたカットの品質と、認識器が正しくセグメントされた文字と文字片1、複数文字1、あるいは正しくセグメントされていない文字を区別する能力に依存する。このタスクを実行するための認識器のトレーニングは、正しく分割されていない文字のラベル付きデータベースを作成することが困難であるため、大きな課題となっている。最も単純な解決策は、文字列の画像をセグメンテーション機能1
に通し、すべての文字仮説を手動でラベル付けすることである。しかし、この作業は非常に面倒でコストがかかるだけでなく1、一貫したラベリングを行うことが困難である。例えば、切り分けた4の右半分は1とラベル付けすべきか、それとも非文字とラベル付けすべきか、切り分けた8の右半分は3とラベル付けすべきか、などである。

セクションVで説明した最初の解決策1は、文字レベルではなく、文字列全体のレベルでシステムを学習させることである1。この目的のためには、勾配学習（Gradient-Based
Learning）の概念を利用することができる。システムは、誤答の確率を測定する全体的な損失関数を最小化するように学習される。セクションVでは、この損失関数を最小化するための様々な方法について検討する。

そのため、勾配に基づく学習法（Gradient-Based Learning）の使用に適している。セクションVでは、代替仮説を表現する方法として、弧が数値情報を持つ有向無サイクルグラフの利用を紹介し1、GTNの考え方を紹介する。

セクション VII で説明する 2 つ目の解決策は、セグメンテーションを完全に排除することである。このアイデアは、入力画像1上のすべての可能な場所に認識器を掃引し、認識器の「文字スポッティング」特性1、すなわち、中心がない文字を含む画像を拒否しながら、入力フィールド1内に他の文字があっても、中心がある文字を正しく認識する能力に頼ることである26]1 27]。認識器を入力フィールド上でスイープすることによって得られる認識器出力のシーケンスは、次に、言語的制約を考慮に入れ、最終的に最も可能性の高い解釈を抽出するグラフ変換ネットワークに供給される。このGTNは隠れマルコフモデル（HMM）1に類似しており、古典的な音声認識を彷彿とさせるアプローチである28]1。
29].この技術は一般的なケースでは非常に高価ですが1、Convolutional Neural Networksを使用することで、計算コストを大幅に削減することができるため、特に魅力的です。

E.グローバル・トレイナブル・システム

前述したように、実用的なパターン認識システムの多くは、複数のモジュールで構成されている。例えば、文書認識システムは、注目領域を抽出するフィールドロケータ1、入力画像を文字候補画像に切り出すフィールドセグメンタ1、文字候補を分類・採点する認識器1、認識器が生成した仮説の中から文法的に正しい答えを選択する確率文法1に基づく文脈後処理1、から構成されている。ほとんどの場合1、モジュールからモジュールに伝達される情報は、円弧に数値が付加されたグラフとして表現されるのが最適である。例えば1、認識モジュールの出力は、各アークが候補文字のラベルとスコアを含み1、各パスが入力文字列の代替解釈を表す非循環グラフとして表現することができる。通常、各モジュールは手動で最適化され1、時にはコンテキストに依存しない形で学習され1る。例えば1
文字認識システムは、あらかじめセグメント化された文字のラベル付き画像で学習される。次に、システム全体を組み立て1、全体の性能を最大にするために、モジュールのパラメータのサブセットを手動で調整する。この最後のステップは非常に面倒であり1、時間がかかるため1、ほぼ確実に最適とは言えない。

より良い代替案は、文書レベルでの文字の誤分類の確率のようなグローバルなエラー測定を最小化するように、何らかの方法でエンタイヤ・システムを学習させることである。理想的には1、システムのすべてのパラメータに関して、このグローバルな損失関数の良い最小値を見つけたいものである。性能を測定する損失関数Eがシステムの調整可能なパラメータW 1に対して微分可能であれば、勾配に基づく学習を用いてEの局所最小値を求めることができる。しかし、1

一見すると1、システムの規模が大きく、複雑であるため、実現不可能なように見えます。

グローバルな損失関数 $E^p(Z^p, W)$ が微分可能であることを保証するために1、システム全体は微分可能なモジュールのフィードフォワードネットとして構築されている。各モジュールが実装する関数は、モジュールの内部パラメータ（例えば、文字認識モジュールの場合、ニューラルネット文字認識器の重み）1
およびモジュールの入力に対して、連続的で、かつ、ほとんどの場所で微分可能である必要がある。この場合1、よく知られたバックプロパゲーション法の単純な一般化を使って、システムのすべてのパラメータに関する損失関数の勾配を効率的に計算することができる22]。X_n
はモジュールの出力を表すベクトル1 W_n
はモジュールの調整可能なパラメータのベクトル（W のサブセット）1 X_{n-1}
はモジュールの入力ベクトル（前のモジュールの出力ベクトルも同様）である。最初のモジュールへの入力 X_0 は、入力パターン Z^p です。 X_n に関する E^p の偏導関数が既知1 であれば、W_n と X_{n-1} に関する E^p
の偏導関数は、後方回帰を使って計算することが可能です。

$$\frac{\partial E^p}{\partial W_n} = \frac{\partial F}{\partial W}(W_n, X_{n-1}) \frac{\partial E^p}{\partial X_n} \qquad (4)$$

$$\frac{\partial E^p}{\partial X_{n-1}} = \frac{\partial F}{\partial X}(W_n, X_{n-1}) \frac{\partial E^p}{\partial X_n}$$

ここで $\frac{\partial F}{\partial W}(W, X)$ は F のヤコビアンである。
ベクトル関数のヤコビアンは、すべての入力に対するすべての出力の偏導関数を含む行列であり、点 (W, X)1 で評価された W と $\frac{\partial F}{\partial X}(W, X)$ は X に関する F のヤコビアンです。最初の式は $E^p(W)$1 の勾配のいくつかの項を計算し、2番目の式はニューラルネットワークのよく知られたバックプロパゲーション手順のようにバックワード再帰1
を生成する。この勾配を学習パターンにわたって平均化すると、完全な勾配を得ることができる。多くの場合、ヤコビアン行列を明示的に計算する必要がないのは興味深いことです。上式はヤコビアンと偏微分のベクトルとの積を用いるが1、ヤコビアンを計算せずにこの積を直接計算する方が簡単な場合が多い。通常の多層ニューラルネット1との類似性から、最後のモジュール以外は出力が外部から観測できないため隠れ層と呼ばれる。より一般的な場合における完全に厳密な導出は，ラグランジュ関数を用いて行うことができる

ーションシステムは、円弧に数値情報が付加されたグラフで表現するのが最も適している。この場合、グラフ変換器と呼ばれる各モジュール1 は、1つ以上のグラフを入力1
として受け取り、グラフを出力として生成する。このようなモジュールのネットワークをグラフトランスフォーマーネットワーク(GTN)と呼ぶ。第 IV1 章、第 VI 章、第 VIII 章では、GTN1
の概念を展開し、勾配に基づく学習により、全モジュールのパラメータを学習し、グローバル損失関数を最小にすることができることを示す。状態情報がグラフのような本質的に離散的なオブジェクトで表現されるときに勾配が計算できることは逆説的に見えるかもしれないが1、その困難は後で示すように回避できる1。

II. 孤立文字認識のための畳み込みニューラルネットワーク

勾配降下法を用いて学習された多層ネットワークは、大量の例から複雑な1
高次元非線形マッピングを学習できるため、画像認識タスクの候補となるのは明らかである。パターン認識の伝統的なモデル1
では、手作業で設計された特徴抽出器
が入力から関連情報を収集し、無関係な変数を除去する。そして、学習可能な分類器は、結果として得られる特徴ベクトルをクラスに分類する。この方式では、標準的な1つの完全連結多層網が使用される。
を分類器として使用することができる。より相互作用が強い可能性がある
エスティングスキームは、可能な限り学習に依存することです。
を特徴抽出器自体に組み込む。 文字の場合
認識1
ネットワークにほぼ生のデータ（例えば、サイズ正規化された画像）を供給することができる。これは、以下のように行うことができる。
20]1 21]1 22].

従来の多層ニューラルネットは、状態情報 X_n が固定サイズのベクトルで表現され1、モジュールが行列の乗算（重み）と成分単位のシグモイド関数（ニューロン）のアルターネート層である上記の特殊なケースであった。しかし、前述したように1、複雑な認識系では、状態情報Xを固定サイズのベクトルで表現する。

は、通常の完全連結フィードフォワードネットワークを用いることで、文字認識などのタスクで一定の成功を収めているが1、問題点もある。

まず、一般的な画像は数百の変数（画素）を持つ大きなものである。例えば100個の隠れユニットを持つ完全連結の第1層1には、数万個の重みが含まれることになる。このような多数のパラメータはシステムの容量を増大させるため、より大きな学習セットを必要とする。また、多くの重みを保存するためのメモリが必要なため、ハードウェアの実装ができない場合もある1。しかし、画像や音声のアプリケーションにおける非構造化ネットの主な欠点は、入力の並進1や局所的な歪みに対する不変性が組み込まれていないことである。ニューラルネットの固定サイズ入力層に送られる前に、文字画像1やその他の2次元または1次元信号1は、ほぼサイズ正規化され、入力フィールドの中央に配置される必要がある。残念ながら、このような前処理を完璧に行うことはできません。手書き文字は単語レベルで正規化されることが多いため1、個々の文字にサイズや傾き、位置のばらつきが生じる可能性があります。この1　　と書き方の違い1が組み合わさると、入力オブジェクトの特徴的な位置のばらつきが生じる。原理的には1、十分な大きさの完全連結ネットワークがあれば、このようなばらつきに対応した出力を生成するように学習することができる。しかし、そのような学習をすると、入力のどこに特徴があっても検出できるように、類似の重みパターンを持つ複数のユニットが入力の様々な位置に配置されることになるだろう。このようなウェイトパターンを学習することで

の場合、可能なバリエーション空間をカバーするために非常に多くの学習インスタンスが必要となる。後述の畳み込みネットワーク1
では、空間的に重みの配置を強制的に複製することで、自動的にシフト不変性を得ることができる。

第二に、完全連結型アーキテクチャの欠点は、入力のトポロジーが完全に無視されることである。入力される変数は、学習の結果に影響を与えることなく、どのような（固定された）順序で提示されてもよい。これに対して、画像（あるいは音声の時間周波数表現）は、空間的あるいは時間的に近接した変数（あるいはピクセル）が高い相関を持つという、強い2次元局所構造を持っている1。局所的な相関は、空間的または時間的なオブジェクトを認識する前に局所的な特徴を抽出・結合することの利点としてよく知られている理由です1
なぜなら、隣接する変数の構成は少数のカテゴリ（例えば、エッジ1コーナー...）に分類することができるからです。畳み込み網は、隠れユニットの受容野を局所的に制限することで、局所特徴の抽出を強制している。

A.畳み込みネットワーク

畳み込みネットワークは、ある程度のシフト1スケール1および歪み不変性を確保するために、局所受容野1共有重み（または重み複製）1および空間または時間サブサンプリングという3つのアーキテクチャ上のアイデアを組み合わせたものである。文字認識のための典型的な畳み込みネットワーク1（LeNet-51と呼ばれる）を図2に示す。入力面には、ほぼサイズ正規化され、中央に配置された文字の画像が入力される。ある層の各ユニットは、前の層の小さな近傍に位置するユニット群から入力される。入力の局所受容野にユニットを接続するというアイデアは60年代初頭のパーセプトロンに遡る1が、HubelとWieselが猫の視覚系で局所的に感度を持つ1方向選択性ニューロンを発見したのとほぼ同時である30]。局所結合は、視覚学習の神経モデルで何度も用いられてきた31]1 32]1 18]1 33]1 34]1 2]。局所受容野1を持つニューロンは、向きを変えたエッジ、端点、角といった初歩的な視覚的特徴（あるいは音声スペクトログラムなど他の信号における同様の特徴）を抽出することができる。これらの特徴は、高次の特徴を抽出するために、後続の層によって結合される。前述したように、入力の歪みやずれにより、顕著な特徴の位置が変化することがある。また、画像の一部分で有効な素性検出器は、画像全体でも有効である可能性が高い。この知見は、画像上の異なる場所に受容野を持つユニット1
の集合に同一の重みベクトルを持たせることで応用できる32]1。
15]1
34].層内のユニットは、すべてのユニットが同じ重みのセットを共有する平面で編成される。このような平面におけるユニットの出力の集合を特徴マップと呼ぶ。特徴マップのユニットはすべて、画像の異なる部分に対して同じ操作を行うように制約されている。畳み込み層は、複数の特徴マップ（ウェイトベクトルが異なる）1
から構成され、各位置で複数の特徴を抽出することが可能である。具体的には、図 2 に示す LeNet-5 の第 1 階層がそうである。LeNet-5 の第 1 隠れ層のユニットは、6 つの層で構成されている。

プレーン1
がそれぞれ特徴マップである。特徴マップのユニットは 25 個の入力を持ち，入力の 5 × 5 の領域1
をそのユニットの受容野と呼ぶ．各ユニットは25個の入力1
を持つので、25個の学習可能な係数と学習可能なバイアスを持つ。特徴マップの連続するユニットの受容野は、前の層の対応する連続するユニットを中心とする。そのため、隣接するユニットの受容野は重なり合う。例えば1、LeNet-
51の第1隠れ層では、水平方向に連続するユニットの受容野は4列5行に渡って重なっている。前述したように1つの特徴マップに含まれる全てのユニットが、同じセットの

2
5
の重みとバイアスが同じであるため、入力上のすべての可能な位置で同じ特徴を検出することができる。層内の他の特徴マップは異なる重みとバイアス1
を使用し、異なるタイプの局所的特徴を抽出する。LeNet-
51の場合、6つの特徴マップの同じ場所にある6つのユニットによって、各入力位置で6種類の特徴が抽出される。特徴マップを逐次実装する場合、局所受容野1
を持つ1つのユニットで入力画像を走査し、そのユニットの状態を特徴マップの対応する位置に格納する。この操作は、畳み込み1
と、それに続く加算バイアスおよびスカッシング関数1
に相当するため、畳み込みネットワークと呼ばれている。畳み込みのカーネルは、特徴マップのユニットが使用する接続重みの集合である。畳み込み層の興味深い特性は、入力画像がシフトした場合1、特徴マップの出力も同じだけシフトするが1、それ以外は変更されないことである。この性質は、入力のずれや歪みに対する畳み込みネットワークの頑健性の基礎となる。

一度検出された特徴量1
は、その正確な位置はあまり重要ではなくなります。他の特徴量との相対的なおおよその位置だけが重要である。例えば、入力画像に、左上のほぼ水平なセグメントの終点、右上の角、画像の下部のほぼ垂直なセグメントの終点があることが分かれば、入力画像は7であることが分かる。これらの特徴の正確な位置はパターンの識別に関係ないだけでなく、その位置が文字の異なるインスタンスで異なる可能性があるため有害である可能性もある。特徴マップの中で識別可能な特徴の位置の精度を下げる簡単な方法は、特徴マップの空間解像度を下げることである。これは、局所平均とサブサンプリング1
を行ういわゆるサブサンプリング層で実現でき、特徴マップ1
の解像度を下げ、シフトや歪みに対する出力の感度を下げることができる。LeNetの第2隠れ層は
5
はサブサンプリング層である。この層は、前の層の各特徴地図に対応する6つの特徴地図1
で構成される。各ユニットの受容野は、前の層の特徴マップの2×2の領域である。各ユニットは4つの入力の平均を計算し1、それに学習可能な係数を掛け1、学習可能なバイアスを加え1、その結果をシグモイド関数に通す。連続したユニットは重複しない連続した受容野を持つ。その結果、サブサンプリング層の特徴マップは、行と列の数が半分になる1。

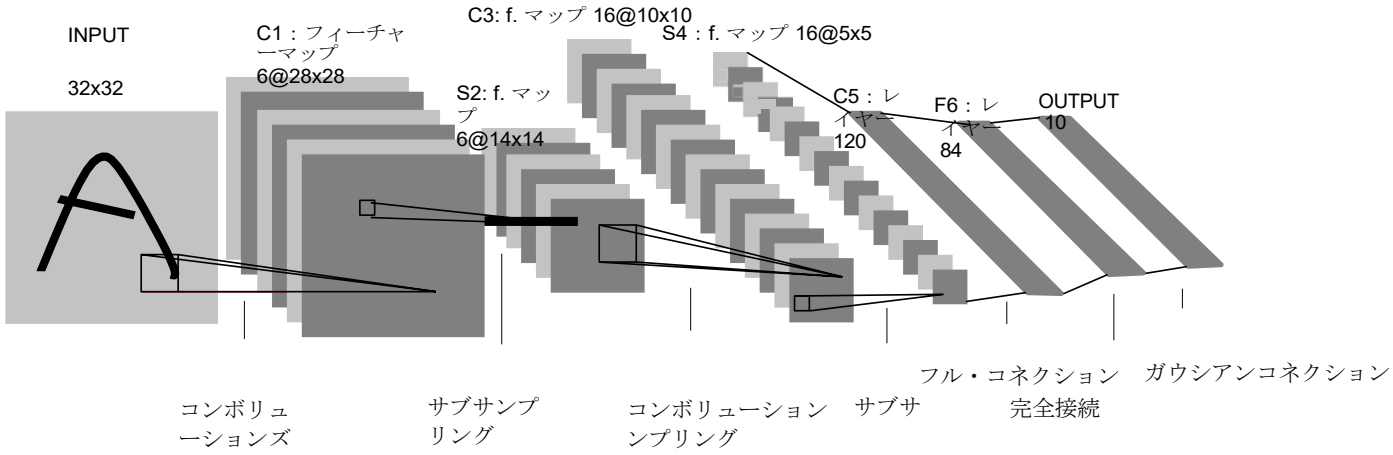

図2.畳み込みニューラルネットワークLeNet-5の構成。各平面は特徴マップ、すなわち重みが同一であることが制約されたユニットの集合である。

を前の層の特徴マップとして用いる。学習可能な係数とバイアスは、シグモイドの非直線性の効果を制御する。係数が小さい場合1、ユニットは準線形モード1で動作し、サブサンプリング層は単に入力をぼかすだけである。係数が大きい場合1、サブサンプリングユニットは、バイアスの値によって「ノイズの多いOR」または「ノイズの多いAND」関数を実行すると見ることができる。畳み込みとサブサンプリングの連続した層は通常交互に繰り返され1、「バイピラミッド」となる。各層1において、空間解像度が低下するにつれて特徴マップの数が増加する。図 2 の第 3 隠れ層の各ユニットは、前の層の複数の特徴マップから入力接続することができる。Hubel と Wiesel の「単純な」細胞と「複雑な」細胞という概念に着想を得た畳み込み／サブサンプリングの組み合わせ1は、福島の Neocognitron 32]1で実装されたが、当時はバックプロパゲーションのようなグローバルな教師付き学習手法は存在しなかった。このように空間分解能を徐々に低下させながら、表現の豊かさ（特徴マップの数）を徐々に増加させることにより、入力の幾何学的変換に対して大きな不変性を達成することができる。

すべての重みはバックプロパゲーションで学習されるため1、畳み込みネットワークは、それ自身の特徴抽出器を合成していると見ることができる。この重み共有技術は、自由パラメータ1の数を減らすという興味深い副次的効果を持ち、それによってマシーンの「容量」を減らし、テストエラーと学習エラーの間のギャップを減らすことができる[34]。図 2 のネットワークは 3401908 個の接続1を持つが、ウェイトシェアリングにより、学習可能な自由パラメータは 601000 個にとどまる。

固定サイズ畳み込みネットワークは、手書き認識 35]1 36]1 機械印字文字認識 37]1 オンライン手書き認識 38]1 および顔認識 39]などの多くのアプリケーション1に適用されている。1つの時間次元に沿って重みを共有する固定サイズの畳み込みネットワークは、時間遅延ニューラルネットワーク（TDNN）として知られている。TDNNは音素認識（サブサンプリングなし）に用いられてきた40]1 41]1 音声単語認識（サブサンプリングあり） 42]1 43] 1 孤立した手書き文字のオンライン認識 44] 1 および署名検証 45]。

B.LeNet-5

本節では、実験に用いた畳み込みニューラルネットワーク LeNet-51 のアーキテクチャをより詳細に説明する。LeNet-5 は入力層を除く7層1 から構成され、その全てに学習可能なパラメータ（重み）が設定されている。入力は32x32ピクセルの画像である。これは、データベース中の最大の文字（28x28フィールドを中心とした最大20x20ピクセル）よりもかなり大きい。その理由は、ストロークの終点や角などの潜在的な特徴量が、最高レベルの特徴検出器の受光野の中心に現れることが望ましいからである。LeNet-5では、最後の畳み込み層（後述のC31）の受容野の中心の集合は、32x32の入力の中心に20x20の領域を形成する。入力画素の値は、背景レベル（白）が-0.1、前景（黒）が1.175に対応するようにノルマライズされている。これにより、平均入力はおおよそ01 、分散はおおよそ1になり、学習が加速される46]。

以下、畳み込み層はCx1、サブサンプリング層はSx1 、完全連結層はFx1と表記し、xは層のインデックスとする。

層C1は6つの特徴マップを持つ畳み込み層である。各特徴マップの各ユニットは、入力の5x5近傍に接続されている。特徴マップのサイズは28x28であり、入力からの接続が境界から外れることを防いでいる。C1には156の学習可能なパラメータ1 と1221304の接続がある。

層 S2 は、サイズ 14x14 の 6 つの特徴マップを持つサブサンプリング層である。各特徴マップの各ユニットは、C1の対応する特徴マップの2x2近傍に接続される。S2のユニットへの4つの入力は、加算1された後、学習可能な係数1が掛けられ、学習可能なバイアスに加えられる。その結果はシグモイド関数に通される。2x2 の受容野は非重複であるため1 、S2 の特徴マップは C1 の特徴マップの半分の行数、列数になる。S2層は12個の学習可能なパラメータと51880個の接続を持つ。

層C3は16の特徴マップを持つ畳み込み層である。各特徴マップの各ユニットは、S2の特徴マップの部分集合の同一位置にある複数の5x5近傍に接続される。 表IにS2の特徴マップの集合を示す

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

表1
各列は、S2
のどのフィーチャーマップを合成したかを示す。

C3の特定のフィーチャーマップに含まれるユニットによって

C3フィーチャーマップは、それぞれのC3フィーチャーマップによって結合される。なぜ、すべてのS2フィーチャーマップとすべてのC3フィーチャーマップを接続しないのか？その理由は2つある。第一に、非完全接続方式は、接続数を合理的な範囲に抑えることができる。より重要なのは、1.ネットの対称性を崩すことである。異なる特徴マップは、異なる入力セットを得るため、異なる（できれば相補的な）特徴を抽出することを余儀なくされる。表 I の接続方式の論理的根拠は以下の通りである。最初の 6 つの C3 特徴マップは、S2 の 3 つの特徴マップの連続する部分集合のすべてから入力を得る。次の 6 個は、4 個の連続する部分集合から入力を得る。次の3つは、4つの不連続な部分集合から入力を取る。最後に、最後の1つはS2のすべての特徴マップから入力を取る。層C3は11516の学習可能なパラメータと151160の接続を持つ。

レイヤ S4 は、サイズ 5x5 の 16 個の特徴マップからなるサブサンプリング層である。各特徴マップの各ユニットは、C1、S2と同様に、C31の対応する特徴マップの2x2近傍に接続される。S4層は 32 個の学習可能なパラメータと 21000 個の接続を持つ。

層C5は120の特徴マップを持つ畳み込み層である。各ユニットは5x5の近傍領域に接続され、その近傍領域にはすべての 16 であり、S4 の特徴量マップのここで1、S4のサイズも5x51であるから、C5の特徴マップのサイズは1x1であり、これはS4とC5が完全に接続されていることに相当する。C5 が完全結合層1 ではなく畳み込み層1 とされているのは、LeNet-5 の入力を大きくして他を一定にすると1x1 よりも特徴マップの次元が大きくなってしまうからである1。この畳み込みネットワークのサイズを動的に増加させるプロセスについては、第 VII

ここで、Aは関数の振幅、Sは原点での傾きを決める。関数fは奇数1であり、水平方向の漸近線は+Aおよび−Aにある。定数Aは1.7159に選ばれている。このスカッシング関数の選択の根拠は付録Aに示されている。

最後に1 出力層はユークリッド基底関数ユニット（RBF）1 で構成され、各クラス1ごとに 84 をそれぞれ入力する。各RBFユニットy_i の出力は以下のように計算される。

章で説明する。層 C5 は 481120 個の学習可能な接続を持つ。

層F61は84ユニット（この数の理由は出力層1の設計に由来する）であり、C5に完全に接続されている。この層は101164個の学習可能なパラメータを持つ。

古典的なニューラルネットワーク1 と同様に、F6層までのユニットは入力ベクトルと重みベクトル1 の内積を計算し、それにバイアスを加える。このユニット i1 の重み付き和(a_i) をシグモイドスカッシュ関数に通すと、ユニット i1 の状態 (x_i) が得られる。

$$x_i f(a_i) \quad (5)$$

スカッシング関数は、スケーリングされたハイパーボリックタンジェントである。

$$f(a) A \tanh(Sa) \quad (6)$$

$$y_i = \sum_j (x_j - w_{ij})^2 . \qquad (7)$$

つまり、各出力RBFユニットは、その入力ベクトルとパラメータベクトルとの間のユークリッド距離を計算する1。入力がパラメータベクトル1から離れれば離れるほど、RBFの出力は大きくなる。特定のRBFの出力は、入力パターンとRBFに関連するクラスのモデルとの適合度を示すペナルティ項として解釈することができる。確率論的な用語1 では、RBF 出力は、層 F6 の構成の空間におけるガウス分布の非正規化負対数尤度と解釈することができる。入力パターン1が与えられたとき、損失関数は、F6の構成がパターンの望ましいクラスに対応するRBFのパラメータベクトルにできるだけ近くなるように設計する必要がある。これらのユニットのパラメータ・ベクトルは手作業で選ばれ、（少なくとも最初は）固定されている。これらのパラメータ・ベクトルの成分は-
1または+1に設定された。1と+11を同じ確率でランダムに選ぶこともできたし，47]1が提案するように誤り訂正符号を形成するように選ぶこともできたが，代わりに7x12のビットマップに描かれた対応する文字クラスの様式化された画像を表現するように設計した（それゆえ84という数字になった）．このような表現は、等比数列の数字を認識するには特に有用ではありませんが1、印刷可能なASCIIセットから取り出した文字列を認識するには非常に有用です。その理由は、大文字の O1 小文字の O1 とゼロ1 や、小文字の l1 数字の角括弧1 と大文字の I1 など、似ていて混同しやすい1 文字は、出力コードが似てくるからです。これは、このような混同を修正する言語的な後処理装置と組み合わせた場合に特に有効である。混同可能なクラスのコードが類似しているため1、曖昧な文字に対応するRBFの出力も類似し1、後処理装置が適切な解釈を選択できるようになるのである。図3は、ASCIIフルセットに対する出力コードである。

出力に一般的な "1 of N "コード（プレースコード1やグランドマザーセルコードとも呼ばれる）ではなく、このような分散コード1を用いるもう一つの理由は、クラスの数が数十より大きくなると非分散コードの動作が悪くなる傾向があるからである。その理由は、非分散コードの出力ユニットはほとんどの時間オフでなければならないからである。これはシグモイドユニットではかなり難しい。さらにもう一つの理由は、分類器は文字1の認識だけでなく、非文字の再分析にも使われることが多いからである。シグモイドと異なり，分布符号を持つRBFは，その内部でうまく囲われた領域で活性化されるため，この目的により適している1.

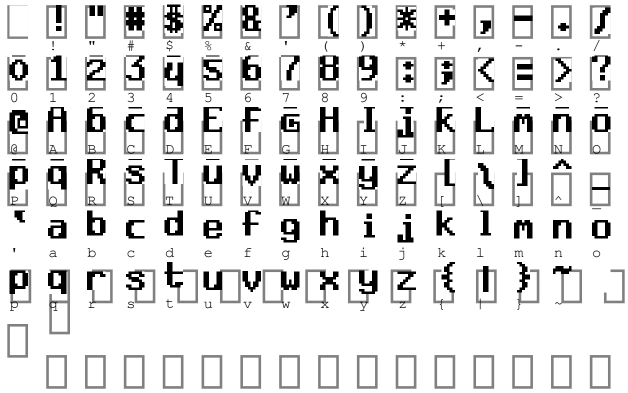

図3.
を認識するための出力RBFの初期パラメータ。ASCIIフルセット

非定型パターンが落ちやすいスペースを置くの外にある。

RBFのパラメータ・ベクトルは層F6のターゲット・ベクトルの役割を担っている。これらのベクトルの成分は+1または-11であり、F61のシグモイドの範囲内であるため、シグモイドが飽和するのを防ぐことができることは指摘に値する。実際、+1や-1はシグモイドの曲率が最大となる点である。これにより、F6ユニットは最大限の非線形の範囲で動作することになる。シグモイドが飽和すると、収束が遅くなり、損失関数の条件付けがおかしくなることが知られているため、シグモイドの飽和は避けなければならない。

C.損失関数

上記のネットワークで使用できる最も単純な出力損失関数は最尤推定基準（MLE）1であり、この場合、最小平均二乗誤差（MSE）と等価である。学習サンプルのセットに対する基準は単純である。

$$E(W) \quad \frac{1}{P}\sum_{p=1}^{P} y_{D^p}(Z^p, W) \qquad (8)$$

ここで，y_{D_p} は D_p -th RBF unit1の出力，すなわち入力パターン Z^p の正しいクラスに対応するものである．このコスト関数はほとんどの場合において適切であるが1，3つの重要な性質が欠けている．まず，RBFのパラメータを適応させる場合1 E(W) には，些細な1しかし全く受け入れがたい1解がある．この解1では、RBFのパラメータ・ベクトルはすべて等しく1、F6の状態は一定でそのパラメータ・ベクトルに等しくなる。この場合、ネットワークは入力を無視し1、すべてのRBF出力はゼロに等しくなる。RBFの重みに適応させない場合、このような崩壊現象は起こらない。第二の問題は、クラス間に競争

この基準は、MSE基準1のように正しいクラスのペナルティを押し下げるだけでなく、誤ったクラスのペナルティも引き上げることを意味します。

$$E(W) \quad \frac{1}{P}\sum_{p=1}^{P} \left(Y_{D^p}(Z^p, W) + \log\left(e^{-j} + \sum_i e^{-y_i(Z^p, W)}\right)\right) \qquad (9)$$

第2項の負は「競争」的な役割を果たす。したがって、この損失関数は正である。定数jは正の値1であり，al-al-al-al-al-al-al-al-al-のクラスがペナルティを受けるのを防ぐ。

をさらに押し上げることで、非常に大きな準備ができます。このポス
この屑クラスラベルの劣後確率は、屑クラスラベルの劣後確率となる。
の割合が　と $e^{-j} + \sum_i e^{-y_i(Z^p, W)}$. この差別的な
この基準は、RBFパラメータを学習する際に、RBFの中心を互いに離すことにより、先に述べた「崩壊効果」を防ぐことができる。項VI1では、入力中の複数のオブジェクト（例えば、単語や文書中の1文字）を分類するために学習するシステムに対するこの基準の一般化を提示する。

畳み込みネットワークの全層の重みに対する損失関数の勾配を計算するには、バックプロパゲーションを使用する。標準的なアルゴリズムは、重みの共有を考慮するために若干変更する必要がある。これを実装する簡単な方法は、まず、ネットワークが重み共有のない従来の多層ネットワークであるかのように、各接続1に関する損失関数の偏導関数を計算することである。次に、同じパラメータを共有するすべての接続の偏導関数を追加して、そのパラメータに関する導関数を構成する。

このような大規模なアーキテクチャは非常に効率的に学習することができますが1、そのためには、付録で説明するいくつかのテクニックを使用する必要があります。付録のセクションA

は、使用したシグモイド1や重みの初期化などの詳細を記述している。セクションB とC では、使用した最小化法1 について述べる。
がないことである。このような競合は，HMMの学習に使用されることのある最大相互情報量基準に類似した MAP (maximum a posteriori) 基準と呼ばれるより識別性の高い学習基準1を使用することによって得ることができる 48]1 49]1 50]．これは，入力画像がクラスの1つまたは背景の「ゴミ」クラスラベルから来る可能性がある場合，正しいクラスD_pの事後確率を最大化（または正しいクラスの確率の対数を最小化）1に相当します．の観点からは

Levenberg-Marquardt手順の対角近似の。

III.結果および他との比較メソッド

数字の認識は、実用的な認識システムの設計に関わる多くの問題の一つに過ぎないが1、形状認識手法を比較するための優れたベンチマークとなる。既存の多くの手法は、手作業で作成した特徴抽出器と学習可能なクラス分類器を組み合わせているが1、本研究では、サイズ正規化された画像に対して直接動作する適応的な手法に焦点を当てる。

A.データベース：Modified NISTセット

本論文で説明するシステムの学習と試験に用いるデータベースは、NISTのSpe-cial Database 3とSpecial Database 1の手書き数字2値画像から構成されるものである。NISTは当初、SD-3をトレーニングセット、SD-1をテストセットとした。しかし、SD-3はSD-1よりはるかにきれいで、認識しやすい。1.この理由は、SD-3が

は国勢調査局職員1 を対象に、SD-1 は高校生を対象に収集された。学習実験から感覚的に結論を導き出すには、学習セットとテストの選択に依存しないことが必要である。そのため、NISTのデータセットを混合して新たなデータベースを構築する必要があった。SD-1には500人の書き手が書いた581527桁の数字画像が含まれている。SD-31では各ライターのデータが順番に表示されるのに対し1、SD-1ではデータにスクランブルがかけられている。SD-1のライターの身元が判明しているため、この情報を用いてライターのスクランブルを解除した。SD-1を2つに分割し、最初の250人のライターが書いた文字を新しい学習セットに入れた。残りの250人のライターはテストセットとした。こうして2つのセットができあがり、それぞれ301000例近くとなった。新しい学習セットには、パターン#01から始まるSD-31の例を十分に加え、601000の学習パターンの完全なセットを作成することができた。同様に1、新しいテストセットは、パターン#351000から始まるSD-3の例で完成し、601000のテストパターンを持つフルセットになった。本実験では、101000枚のテスト画像の一部（SD-1の51000枚とSD-3の51000枚）1のみを使用し、601000枚の学習サンプルをフルに使用した。このようにして得られたデータベースをModified NIST1またはMNIST1データセット。

オリジナルの白黒（2値）画像は、縦横比を保ったまま20×20ピクセルのボックスに収まるようにサイズ正規化されました。得られた画像には、正規化アルゴリズムで使用されたアンチエイリアス（画像補間）技術の結果、グレーレベルが含まれている。データベースは3つのバージョンを使用した。
最初のバージョン1では、ピクセルの重心1
を計算し、この点が28x28フィールドの中心に位置するように画像を平行移動させることによって、画像を28x28画像の中央に配置しました。この28x28のフィールドを背景画素で32x32に拡張した場合もある1。このバージョンのデータベースを通常のデータベースと呼ぶことにする。第2バージョンでは、文字画像をデスランティングし、20×20ピクセルの画像にトリミングしている。デスランティングでは、画素の慣性2次モーメントを計算し（前景画素を1、背景画素を0と数える）1、主軸が垂直になるように線を水平方向にずらして画像を切り取る。このバージョンのデータベースをデスランテッドデータベースと呼ぶことにする。 初期の実験で用いられた第3バージョンのデータベース1 では、画像は16x16ピクセルに縮小されている。通常のデータベース（60 1000個の学習例1
101000個のテスト例、サイズは20x201に正規化、28x28フィールドの重心で中心化）は
http://www.research. att. com/~yann/ocr/mnist で利用可能である。図4は、テストセットからランダムに選んだ例である。

B.結果

通常の MNIST データベースを用いて，いくつかのバージョンの LeNet-5
を学習させた．各セッションにおいて、学習データ全体を 20 回反復した。グローバル学習率 rJ (定義については付録 C の式 21 を参照) の値は、以下のスケジュールで減少させた。最初の2パスで0.0005、次のパスで0.0002。

図4.MNISTデータベースからのサイズ正規化された例。

three1 次の0.0001 three1 次の0.00005 41
であり，それ以降は 0.00001
である．各反復の前に1、付録Cにあるように対角ヘシアン近似を500サンプルで再評価し、全反復の間、固定したままとした。パラメータμは0.02に設定された。その結果、1パス目の有効学習率は約7×10-5から0.016の間でパラメーターのセットに対して変化することがわかった。テスト誤差は訓練セットを10回ほど通過した時点で0.95%に安定する。トレーニングセットの誤差は19パス後に0.35%に達する。ニューラルネットワークやその他の適応的アルゴリズムを様々なタスクで学習させる際、多くの著者が過学習という一般的な現象を観察していると報告している。過学習が起こると1、学習誤差は時間とともに減少し続けるが1、テスト誤差は最小値を経て、ある反復回数の後に増加し始める。この現象は非常に一般的であるが1、図5の学習曲線が示すように、我々のケースでは観察されなかった。その理由として考えられるのは、学習率を比較的大きくしていたことである。その結果、重みはローカルミニマムで落ち着くことなく、ランダムに振動し続ける。このような揺らぎのために1、平均コストはより広い最小値でより低くなる。したがって、確率的勾配は、より広い最小値に有利な正則化項と同様の効果を持つ。より広い極小値は、パラメータ分布のエントロピー1
が大きい解に対応し、一般化誤差に有利となる。

学習セットの大きさの影響は、1510001 3010001 と601000
の試験片でネットワークを学習させることで測定した。その結果、学習誤差とテスト誤差が図6に示されている。LeNet-
51のような特殊なアーキテクチャであっても、学習データを増やせば精度が向上することがわかる1。

この仮説を検証するために、オリジナルの学習用画像をランダムに歪ませることで、より多くの学習用画像を人工的に生成した1。増加した学習セットは、元の601000パターンに加えて、54001000インスタンスの