Gradient-Based Learning Applied to Document Recognitionの和訳

著者

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

概説

逆誤差伝搬法で学習した多層ニューラルネットワークは, 勾配に基づく学習法の最たる成功例である. 適切なネットワークアーキテクチャがあれば, 勾配に基づく学習アルゴリズムは手書き文字などの高次元パターンを分類するような複雑な決定超平面を, 最小限の前処理で 作成することができる. 本論文では, 手書き文字の認識に適用される様々な方法を振り返り, 標準的な手書き整数認識タスクを用いて比較する. その結果, 二次元構造の多様性を取り扱えるように設計された CNN が他の手法に比べて良い性能を示した.

実際の文章認識システムは, フィールド抽出, 分割, 認識, 言語モデリングなど複数のモジュールにより構成されている. Graph Transformer Networks (GTN) と呼ばれる新しい学習の枠組みにより, このような多くのモジュールからなるシステムを勾配に基づく学習アルゴリズムを用いてパフォーマンス指標を最小化するように大域的に学習することが可能となった.

本論文では, オンライン手書き文字認識の 2 つのシステムを紹介する. 実験により, 大域的な学習の利点と GTN の柔軟性が示された.

また, 本論文では銀行小切手を読み取る GTN も紹介する. この手法では, CNN の文字認識と, 商業や個人利用の小切手の正確さを記録するための大域的な学習方法を組み合わせている. 商業的に利用されており,1 日あたりに数百万の小切手の認識をしている.

キーワード – ニューラルネットワーク, OCR, 文章認識, 機械学習, 勾配に基づく学習, CNN, Graph Transformer Networks, Finite State Tranducers.

略称

- GT Graph transformer
- GTN Graph transformer network.
- HMM Hidden Markov model.
- HOS Heuristic oversegmentation
- K-NN K-nearest neighbor
- NN Neural network.
- OCR Optical character recogniton.
- PCA Principal component analysis.
- RBF Radial basis function.
- RS-SVM Reduced-set support vector method.
- SDNN Space displacement neural network.
- SVM Support vector method.
- TDNN Time delay neural network.
- V-SVM Virtual support vector method.

1　1. 序論

ここ数年, 機械学習の技術, それらを適用したニューラルネットワークがパターン認識システムの設計において, 非常に重要な役割を果たしている. 実際に, 近年の連続的な音声の認識, 手書き文字の認識といったパターン認識アプリケーションの成功にはこのような学習法の利用が重要な要因であると言える.

本論文の主な主張は, 人間の経験的な知識に頼らず, 自動的な学習に依存することで, より良いパターン認識システムを作成することができるということだ. これは, 近年の機械学習や情報工学の進歩により可能となった. 文字認識を例として, 本論文ではこれまでの手作業の特徴抽出が, ピクセル画像を直接操作する注意深い学習機械に置き換えることができることを示している. また文章認識を例として, 本論文では個々に設計されたモジュールを組み合わせることで認識システムを作る従来の方法は, 大域的な性能基準を最適化するために全てのモジュールを学習する GTN と呼ばれる統一的で原理的な設計の枠組みに置き換えることができることを示している.

パターン認識の研究初期から, 音声, 文字パターンなどといった自然界のパターンの豊富さと多様さに

より手作業で完全に正確なパターン認識システムを作成することは難しいことが知られている. そのため多くのパターン認識システムは自動的な学習と人間が考案したアルゴリズムを組み合わせて作られている. 個々のパターンを認識するための一般的な方法は, 図 1 に示すようにシステムを 2 つの主なモジュールに分けて構成する. 一方は特徴抽出器と呼ばれるモジュールである. 特徴抽出器では入力パターンを,(a) 照合・比較しやすく,(b) 入力パターンの変形や歪みに対して比較的頑強である低次元のベクトルや短い配列で表現できるように変形する. 特徴抽出器は予備知識の多くを有しており, タスクに特化している. また, ほとんど手作業で作られるため, 設計の際の作業の大半を占める. もう一方は分類機と呼ばれるモジュールであり, 汎用的で学習可能である. このアプローチの大きな問題の 1 つは, 設計者の適切な特徴量のセットを考える能力に認識精度が大きく依存することだ. このことは不幸にもなにか問題が発生した時は, 気が遠くなるような設計作業を再び行わなければならないことを意味している. パターン認識に関する多くの文献は, 特定のタスクに関する異なる特徴量のセットの相対的な長所の比較に費やされている.

歴史的に, 分類器が用いる学習方法が要因にクラス分類可能な低次元空間に限定されていたため特徴抽出器が必要とされた [1]. この 10 年間で 3 つの要因が重なりこの見解は変化してきた. 1 つ目の要因は, 高速の演算装置を持つ低価格の計算機が登場し理論的な方法より強引な数値解析に頼るようになったこと.2 つ目の要因は手書き文字の認識のような市場規模が大きく関心の高い問題に関する大規模なデータベースが使用可能になり, 認識システムの設計において手作業での特徴抽出より, 実データに依存することができるようになったこと.3 つ目の非常に重要な要因は高次元の入力に対応し, 大規模なデータを与えられても複雑な決定関数を生成できる高性能な機械学習方法が生み出されたことである. 近年の音声認識や手書き文字認識システムの精度向上には, 学習技術と大規模なデータセットの進化が大きく起因している. この証拠として最近の商用の OCR システムの大部分は誤差逆伝搬法で学習された多層ニューラルネットワークを使用している.

本論文では手書き文字認識における課題点を検討し (1 節,2 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方法の性能を比較した (3 節). さらなる自動的な学習は有益であるが, タスクに関する最小限の予備知識無しで成功する学習手法は存在しない. 多層ニューラルネットワークの場合, 予備知識を組み込む良い方法はタスクに応じてアーキテクチャを調整する方法である.2 節で紹介する CNN[2] は, 局所的な接続パターンを用いたり重みに制約を与えることで 2 次元形状の不変性に対する予備知識を取り入れた特殊なニューラルネットワークの 1 つである. 分離された手書き整数認識タスクに対するいくつかの方法の比較を 3 章で示す.4 章では個々の文字認識から文書中の単語や文の認識に至るまで, 全体の誤差を減らすため学習した多くのモジュールを組み合わせる方法を紹介する. モジュールが有向グラフを操作可能の場合, マルチモジュールシステムを用いた手書きの単語といった可変長のオブジェクトの認識することが最適である. これは 4 章で述べる学習可能な GTN の概念につながる.5 章では, 単語や文字列を認識するための古典的なヒューリスティックなオーバーセグメンテーションを紹介する.6 章では人力でのセグメンテーションやラベリングを必要としない単語レベルでの識別機の学習を可能にするような識別的, 非識別的な勾配に基づく学習法を紹介する.7 章では, 入力における全ての位置において認識器を作用させることで, 分割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ニューラルネットワークのアプローチを示す. 8 節では, 学習可能な GTN 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として定式化できることを示している. また, 音声認識でよく用いられる隠れマルコフモデルと GTN の関連も示す. 第 9 節は, 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大域的に学習された GTN システムについて説明する. ユーザーが書いた文字を計算機が即座に応答を返さなければならないため, この問題は”オンライン”手書き文字認識として知られている.GTN システムの中核は CNN である. 本研究で得られた結果は, 認識器を事前に領域分割された手書き文字で学習するのではなく単語レベルで学習することの利点を示している. 10 節は, 手書きと機械印刷の銀行小切手を認識する GTN に基づいた完全なシステムを説明する. このシステムの中核は 2 節で述べた LeNet-5 と呼ばれる CNN である. このシステムは銀行産業向けの小切手認識システムとして NCR 社で商業的に利用されている. 全米のいくつかの銀行で一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

機械学習を自動化するための様々なアプローチがあるが, 最も成功したアプローチの 1 つは, 勾配に基づく学習と呼ばれる近年ニューラルネットワークコミュニティで人気のアプローチである. 学習する計算機は関数 $Y^p = F(Z^p, W)$ を計算する. ここで,Z^p は p 番目の入力パターンであり,W はシステムにおける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では, 出力 Y^p はパターン Z^p の認識可能なクラスラベル, あるいは各クラスに関係する確率やスコアとして解釈できる. 損失関数 $E^p = \mathcal{D}(D^p, F(W, Z^p))$ は, パターン Z^p における正しいまたは望ましい出力 D^p とシステムの出力との不一致さを測定する. 平均損失関数 $E_{train}(W)$ は学習セット (Z^1,D^1),...,(Z^P,D^P) と呼ばれるラベル付きの学習例の集合上の誤差 E^p の平均値である. 最も単純な設定において, 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, 学習セットにおける性能はあまり重要ではない. より重要な指標は実際に使用される領域での誤差率である. この指標はテストセットと呼ばれる学習セットから切り離されたサンプル集合に対する正解率を計算することで推定できる. 多くの理論的・実験的研究 [3, 4, 5] により, テストセットで予測されるエラー率と学習セットで期待されるエラー率との間の差は学習サンプルの数によっておおよそ式 1 のように減少することが分かっている.

$$E_{test} - E_{train} = k(h/P)^{\alpha} \tag{1}$$

ここで　P は学習サンプルの数であり, h は「有効容量」, 機械の複雑さの推定値, α は 0.5　1.0 の値で, k は定数である. 学習サンプルの数が増加した時は差は常に減少する. その上 h が増加すると, E_{train} が減少する. 従って, h が増えた時, 最小の汎化誤差 E_{test} を達成する容量 h の最適な値があれば,E_{train} の減少量と差の増加はトレードオフの関係となる. 多くの学習アルゴリズムは E_{train} だけでなく差の増加も最小化しようと試みる. この正式な呼び名は構造的リスク最小化と呼ばれ, それぞれのサブネットが以前のサブネ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系列に対応する容量に増加した学習機械の系列を定義することに基づいている. 実際には, 構造的リスク最適化は $E_{train} + \beta H(W)$ を最小化することを意味している. ここで関数 $H(W)$ は正規化関数と呼ばれ, β は定数である. $H(W)$ はパラメータ空間の高容量のサブセットに所属するパラメータ W に関して大きな値を取るように選択される. $H(W)$ を最小化することで, パラメータ空間におけるアクセス可能なサブセットの容量を制限する. それにより学習セットにおける誤差の最小化と学習セットとテストセットの誤差の差の最小化のトレードオフを制御する.

B. 勾配に基づく学習

パラメータ集合に関する関数の最小化の一般的な問題は, コンピュータサイエンスにおける多くの問題の根底となっている. 勾配に基づく学習は, 一般的に離散関数よりも滑らかな連続関数を最小化するほうがはるかに簡単という事実に基づいている. , 一般的に離散関数よりも滑らかな連続関数を最小化するほうがはるかに簡単という事実に基づいている. 損失関数はパラメータの値の変動が及ぼす影響を推定することで最小化できる. この推定はパラメータに対する損失関数の勾配によって行われる. 摂動函数による数値的な解析とは対称的に分析的に勾配ベクトルが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うる. このことが連続値のパラメータを持つ多くの勾配に基づく学習アルゴリズムの基礎となっている. 本論文で示す手順では, パラメータ集合 W は実数値のベクトルであり $E(W)$ は連続であり, ほぼ全て微分可能である. このような問題設定における最も単純な最小化の手順は勾配降下法であり W は式 2 のように再帰的に調整される.

$$W_k = W_{k-1} - \epsilon \frac{\partial E(W)}{\partial W} \tag{2}$$

もっとも簡単なケースでは ϵ は定数である. より洗練された手順では変数として ϵ を用いるか, ϵ を対角行列に置き換えるか, Newton 法や Quasi-Newton 法のようにヘッシアン行列の逆行列の推定値に置き換える. 共役勾配法も用いられる. しかし付録 B に示すように, 文献では多くの方法が示されているが, これらの 2 次元上の方法は大規模な学習の有用性は非常に限られていることが確認されている.

一般的な最小化手法はオンライン更新とも呼ばれる確率的勾配アルゴリズムがある. これは勾配平均のノイズを含んだもの, または近似したものを用いてパラメータベクトルを更新する手法である. 確率的勾配アルゴリズムのもっとも一般的な例では, W はシングルサンプルの基礎に基づいて更新されます.

$$W_k = W_{k-1} - \frac{\partial E^{p_k}(W)}{\partial W} \tag{3}$$

この手順では, パラメータベクトルは平均の系列を中心に変動する. しかし, 音声や文字認識に見られる冗長な大規模な学習データにおいては通常の勾配降下法や 2 次元の勾配アルゴリズムに比べてかなり高速に収束する. この理由は付録 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いた. しかし, 80 年代半ばまでは非自明な問題に対する実用的な成功例は起こっていなかった.

C．勾配逆伝搬法

勾配に基づく学習法は 1950 年代後半から用いられていた. しかし, それらは線形的なシステムに限定されていた. 複雑な機械学習タスクに対する単純な勾配降下法の驚くべき有用性は以下の 3 つの出来事が起こるまで広く知られていなかった. 1 つ目は当初に問題視されていたにもかかわらず, 損失関数の局所的最小値が実際には大きな問題にならないことが認識されたことである. これはボルツマンマシンといった初期の非線形の勾配に基づく学習の成功に局所的最小値が大きな障害になっていないと考えられたことで認識された. 2 つ目の出来事は数層の処理からなる非線形システムに行ける勾配を計算するための逆伝搬法が, RumelHart, Hinton, Williams らにより一般化されたことである. 3 つ目の出来事はシグモイド関数を有したユニットからなる多層ニューラルネットワークに逆伝搬法を適用することで複雑な学習タスクを解決できることが示されたことである. 逆伝搬法の基本的なアイデアは出力から入力への伝搬により勾配を効率的に計算することである. このアイデアは 60 年代初期の制御理論の文献に示されている. しかし, 機械学習への応用は一般に認識されていなかった. 興味深いことに, 深層学習の文脈における初期の逆伝搬法の導出は, 中間層のユニットや妨害要素の最小化に, 勾配ではなく「仮想ターゲット」を用いていた. 制御理論の文献で用いられていたラグランジュ形式は, 逆伝搬法や RNN, 異種のモジュールからなるネットワークへの逆伝搬法の一般化の導出においておそらく最も厳密な方法を提供した. 一般的な多層システムにおける簡潔な導出は 1-E 説で述べる.

多層ニューラルネットワークで局所最適解が問題にならないという事実は理論的にはやや謎である. タスクに対してネットワークが大きすぎる場合 (実用上この場合が一般的) , パラメータ空間の「追加の次元」の存在が到達不可能な領域の危険性を減らしていると推測されている, 逆伝搬法は, これまで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, 任意の形式の学習アルゴリズムで最も広く用いられている.

D. 手書き文字認識システムにおける学習

分離された手書き文字の認識は文献において集中的に研究されており, ニューラルネットワークの初期の応用において最も成功した例の一つである. 手書き数字認識における比較実験は第 3 節で示す. 比較実験では, 同じデータに対して, 勾配に基づく学習アルゴリズムを用いて学習した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た手法を記録した. CNN 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ために学習を行うように設計されている.

手書き文字の認識において最も困難な問題の 1 つは個々の文字を認識することではなく, セグメンテーションとして知られる, 単語や文中で隣り合う文字を分離する処理である. セグメンテーションのための技術として「ヒューリスティックオーバーセグメンテーション」と呼ばれる手法が標準となっている. この手法では, ヒューリスティックな画像処理手法を用いて多数の文字間の切れ目の候補を作成し, その後認識輝により各文字候補に与えられたスコアに基づいて最適な切れ目の候補の組み合わせを見つける. このようなモデルでは, システムの精度はヒューリスティックな知識を用いて作成した切れ目の候補の質と, 認識器が文字の集合から正しくセグメントされた文字, 多くの文字の集まり, 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気の学習は, 正しくセグメントされていない文字列のラベル付きデータセットを作成することが困難であるため難題となっている. もっとも簡単な解決法は文字列の画像をセグメンターに通して, すべての文字候補に手動でラベリングすることである. しかし, この作業は非常に面倒で労力がかかるだけでなく, 一貫したラベリングを行うことが困難という問題も有している. 例えば, 切り分けた　4 の右半分を 1 としてラベリングするのか文字ではないとラベリングべきだろうか? また, 切り分けた 8 の右半分を 3 とラベリングすべきだろうか？ といった問題が生じる.

第 5 節で述べる 1 つ目の解決策は, 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. このアイデアには勾配に基づいた学習の概念を使用することができる. このシステムは誤答の可能性を計算する全体的な誤差関数の値を最小化するように学習を行う. 第 5 節では誤差関数が微分可能であり, 勾配に基づく学習方法に適しているということを様々な方法を用いて調査する. 第 5 節では代替候補を表現する方法として, 枝が数値情報を持つ有向グラフの利用を紹介し GTN の概念を紹介する.

第 7 節で述べる 2 つ目の解決策は, セグメンテーションを行わないことである. このアイデアでは, 入力画像上の可能なすべての位置に識別器を置き, 認識器の「文字スポッティング」機能, すなわち入力画像内に他の文字があっても中心にある文字を正しく認識できる機能に頼る. 認識器を入力画像上に敷き詰めることで得られる認識器の出力の系列は, 言語的制約を考慮に入れて最終的に最もらしい解釈を抽出するグラフ変換ネットワークに送られる. この GTN は隠れマルコフモデルに類似しており, 古典的な音声認識を連想させるアプローチである. この解決策は一般的には非常にコストがかかるが, CNN を用いることで計算のコストを遥かに節約できるため特に魅力的です.

E. 大域的に学習可能なシステム

前節で述べた通り, 実用的なパターン認識システムは複数のモジュールから構成される. 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ー, 入力画像を文字候補の画像へ切り出す領域分割器, それぞれの文字候補を分類, 採点する認識器, 認識器によって生成された仮説から文法的に正しい答えを選択する確率的文法に基づく文脈後処理器から構成される. ほとんどの場合, モジュール間で伝搬する情報は辺に数値が付与された非巡回グラフとして表現されるのが最適である. 例えば, 認識器モジュールの出力は各枝が候補文字のラベルとスコアを含み, 各パスが入力文字列の解釈を表す非巡回グラフとして表現される. 典型的には各モジュールは手動で最適化されるが, 時には認識システムの流れを離れて学習される. 例えば, 文字認識システムはあらかじめ分離された文字のラベル付き画像で学習される. その後, システム全体を組み合わせ, 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を手動で調整する. この手動で調整する段階は非常に面倒であり時間がかかり, さらにはほとんどの場合確実に最適とは言えない.

よりよい代替案として, 何らかの方法で, 文書レベルでの文字の御分類の確率といった大域的な誤差を最小化するようにシステム全体を学習させることである. 理想的にはシステムのすべてのパラメータについて大域的な損失関数の最小値を見つけたい. もしシステムの性能を測定する損失関数 E がシステム内の調整可能なパラメータ W に対して微分可能であれば, 勾配に基づく学習により損失関数 E の局所的最小値を求めることができる. しかし, 一見するとシステムの規模の大きさや複雑さから実現不可能に思える.

大域的な損失関数 $E^p(Z^p, W)$ が微分可能であることを保証するために, システム全体は微分可能なモジュールの FFN として構成されている. 各モジュールにより実装される関数は, モジュールのパラメータ (例えば, 文字認識モジュールにおけるニューラルネットワーク文字認識器の重み) とモジュールの入力に対して連続的であり, 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, 良く知られている誤差逆伝搬法の単純な一般化を用いてシステムの全てのパラメータに対する損失関数の勾配を効率的に計算することができる. 例えば, システムがモジュールの連鎖として構成されていると考えると, 関数 $X_n = F_n(W_n, X_{n-1})$ として解釈される. ここで, X_n はモジュールの出力を表すベクトル, 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) , X_{n-1} はモジュールの入力ベクトル (直前のモジュールの出力ベクトル) である. 最初のモジュールの入力 X_0 は, パターン系列 Z^p である. X_n に関する E^p の偏導関数が既知であれば, 後退型回帰を用いて, W_n と X_{n-1} に関する E^p の偏導関数を計算することができる.

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

ここで, $\frac{\partial F}{\partial W}(W_n, X_{n-1})$ は (W_n, X_{n-1}) において評価された W に関する F のヤコビアン, $\frac{\partial F}{\partial X}(W_n, X_{n-1})$ は X に関する 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関するすべての出力の偏導関数を含んだ行列である. 1 つめの数式は $E^p(W)$ の勾配を計算し, 2 番目の式は良く知られているニューラルネットワークにおける逆誤差伝

搬法の手順のように後退型回帰を計算する. 学習パターンに沿って計算された勾配を平均化すると, 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ことが興味深い. 上で示した式はヤコビアンと偏微分のベクトルの積を用いるが, ヤコビアンを前もって直接計算するよりもこの席を直接計算する方が簡単なことが多い. 一般的な多層ニューラルネットワークネットワークとの類似性から最後のモジュール以外のすべてのモジュールは出力が外部から観測できないため隠れ層と呼ばれる. 上で述べた単純に連稀有されたモジュール群に比べてより複雑な場合は, 偏微分の表記が曖昧なものとなる. 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われる.

多層ニューラルネットワークは, 情報 X_n が固定長のベクトルで表される上で述べた例の特殊な場合であり, モジュールは行列積 (重み) とシグモイド関数 (ニューロン) の代替層となる. しかし, 前に述べた通り, 複雑な認識システムにおける状態情報は枝に数値情報が付与された非巡回グラフで表現されることが最もである. このような場合, グラフ変換器と呼ばれる各モジュールは, 1 つかそれ以上のグラフを入力として, 1 つの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 4, 6, 8 節では GTN の概念を発展し勾配に基づく学習を用いてすべてのモジュール内のパラメータを大域的な損失関数を最小化するように学習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データ構造で表現されるときに勾配が計算できるのは逆説的に思えるが, この困難は後に示す通り回避できる.

2　2. 分離された文字認識への畳み込みニューラルネットワーク

勾配降下法により学習された多層ネットワークは, 大量の教師データから複雑な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当然である. パターン認識の伝統的なモデルでは手作業で設計された特徴抽出器が入力から関連する情報を集め, 無関係な変数を除去する. 学習可能な分類器は得られた特徴ベクトルをクラスに分類する. この手法では, 標準的な完全連結型多層ネットワークが分類器として用いられる. 潜在的により興味深い手法は特徴抽出を可能な限り学習に頼ることである. 文字認識の場合, ネットワークはほぼ前処理が施されていないデータ (例えば, サイズが正規化された画像) を与えられる. 普遍的な完全連結 FFN を用いることで文字認識などのタスクでいくつかの成功を収めているが, 問題点もある.

第 1 に, 一般的な画像は数百もの変数 (ピクセル) を持つ大きなデータである. 例えば 100 個の隠れ層を持つ完全連結多層ネットワークの第 1 層は, 数万個の重みが含まれる. このような多数のパラメータはシステムの規模を増加させ, 結果としてより大規模な学習データを必要とする. また, 多くの重みを記録しておくためのメモリが必要であるため, ハードウェアによってはメモリ制限を超えてしまう. しかし, 画像や音声アプリケーションへの非構造化ネットワークの主な欠点として入力の変換や局所的な歪みに対しての不変性が組み込まれていない. ニューラルネットワークの入力層に固定長のベクトルを渡す場絵は, 文字画像や他の 2 次元, 1 次元信号はサイズが正規化され入力領域の真ん中に置かれなければならない. 不幸なことに, このような前処理を完璧に行うことはできない. 手書き文字は単語レベルで正規化されることが多く, それぞれの文字に対してサイズや傾き, 位置の相違が生じる. さらに書き方の違いまで組み合わさると, 入力の特徴的な位置にばらつきが生じる. 原理的に, 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する出力を生成するように学習できる. しかし, そのような学習をすると, 入力のどこに特徴があるか検出できるように入力の様々な位置に類似の重みパターンを持つ複数のユニットが配置されることになる. このような重みパターンを学習する際にはあり得る入力のばらつきに対処するために非常に多くの学習データが必要になる. 畳み込みニューラルネットワークでは, 後述するように, 重みの配置を強制的に複製することで自動的に不変性を得ている.

2 つ目の問題点として, 完全連結ネットワークの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. 入力変数は学習の結果に影響を与えることなく, どのような順序で提示されてもよい. 一方で, 画像 (あるいは, 音声の時間周波数表現) には空間的, 時間的に近接した変数が高い相関を持つという強い 2 次元局所的構造を有する. 局所的な相関は, 空間的であれ時間的であれ認識対象のオブジェクトを認識する前に局所的な特徴を特

徴・結合できる利点として良く知られている. なぜなら, 隣接変数の構成は少数のカテゴリ (例えば, エッジ, 角) に分類することができるからである. 畳み込みニューラルネットワークでは, 隠れ層の受信可能な領域を局所的に制限することで局所的な特徴抽出を強制している.

2.1 A. 畳み込みネットワーク

畳み込みネットワークは, ある程度のずれ, 拡大縮小, 歪みの不変性を確保するために局所的受信可能領域, 共有される重み (重み複製) , 時間的または空間的サブサンプリング の 3 つのアイデアをアーキテクチャに取り入れた. 図 2 に, LeNet-5 と呼ばれる文字認識のための一般的な畳み込みネットワークを示す. 入力では, おおよそサイズが正規化され文字が中央に配置された画像を入力する. 層の各ユニットは直前の層の小規模な近傍に位置するユニット群から入力を受け取る. 入力の局所的受信可能領域にユニットを接続するアイデアは, 60 年代初頭のパーセプトロンまで遡り, Hubel と Wiesel が猫の視覚システムのおいて局所的に感度が高く, 適応的選択ニューロンを発見したのとほぼ同時である. 局所的結合はこれまで, 視覚学習のニューラルネットワークモデルに多く用いられている. 局所的受信可能領域を持つニューロンは, 向きのあるエッジ, 終点, 角といった様々な初歩的な視覚的特徴を抽出することができる. これらの特徴は, より高次の特徴を抽出するために後続する層によって結合される. 先述したように, 入力の歪みやずれにより重要度の高い特徴の位置が変化する場合がある. それに加えて画像の一部分で有効な初歩的な特徴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, 受信可能領域を持つユニットの集合に画像上の異なる場所に同一の重みベクトルをもたせることで応用できる. 層の中のユニットは, すべてのユニットが同じ重みの組を共有する平面で構成される. 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動作をするように制約を設けられている. 完全な畳み込み層はいくつかの特徴マップ (異なる重みベクトル) から構成され, それぞれの位置において多くの特徴を抽出できる. 具体例としては, 図 2 に示す LeNet-5 の 1 層目である. 最初の隠れ層のユニットは 6 つの特徴マップで構成される. 特徴マップのユニットは 25 個の入力を持ち, 入力の 5 × 5 の領域をユニットの受信可能領域と呼ぶ. それぞれのユニットは 25 個の入力を持つため結果として, 25 個の学習可能な係数と学習可能な 1 つのバイアスを持つ. 特徴マップの連続するユニットの受信可能領域は, 直前の層の対応する連続するユニットの中央に配置される. そのため, 隣接するユニットの受信可能領域は重なり合う. 例えば, LeNet-5 の最初の隠れ層は, 水平方向に連続するユニットの受信可能領域は, 4 列 5 行にわたって重なっている. 先述したように, 1 つの特徴マップに存在するすべてのユニットは同じ 25 個の重みと同じバイアスを共有するため入力上のすべての可能な位置の同じ特徴を検出可能である. 層内の同じ特徴マップは異なる重みとバイアスを用いており, 異なる種類の局所的特徴を抽出する. LeNet-5 の場合, 6 つの特徴マップの同じ場所に存在する 6 つのユニットにより 6 つの異なる特徴が各入力位置で抽出される. 特徴マップの築地的な解釈により, 局所的受信可能領域を持つ 1 つのユニットで入力画像を精査し, 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, 畳み込みとそれに付随する加算バイアスおよび活性化関数に相当するため, 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により用いられる接続の重みの集合である. 畳み込み層の興味深い特性は, 入力画像にずれが生じると, 特徴マップの出力も同程度ずれ, それ以外は変更されないことである. この特性は, 入力の歪みやずれに対しての畳み込みニューラルネットワークの頑健性の基礎となる.

一度特徴が検出されると, その抽出個所の位置はあまり重要ではなくなる. 他の特徴量とのおおよその相対的な位置のみが重要である. 例えば, 入力画像が左上にやや水平なセグメントの終点, 右上に角, 下部にやや垂直なセグメントの終点があれば, 入力画像は 7 であるとわかる. これらの特徴の正確な位置はパターン認識において無関係だけでなく, 文字の異なる学習データにおいて絶対的な位置は異な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量の位置の精度を減少させる簡潔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, サブサンプリングを実行するいわゆるサブサンプリング層で実現でき, 特徴マップの解像度を下げ, ずれや歪みへの出力の感度を下げることができる. LeNet-5 の 2 つ目の隠れ層はサブサンプリング層である. この層は 6 つの特徴マップからなり, それぞれは前の層の特徴マップに対応して

いる. それぞれの受信可能領域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 つの出力の平均を計算し, 学習可能な係数を乗算し, 学習可能なバイアスを加算し, 結果をシグモイド関数に渡す. 連続するユニットは重なりがない連続した受信可能領域を持つ. 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減する. 学習可能な係数とバイアスは, シグモイドの非線形性の影響を制御する. 係数が小さい場合, ユニットは疑似線形モードで動作し, 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, サブサンプリングユニットはバイアスの値によって, 「ノイズのある OR 」,「ノイズのある AND 」関数を実行すると捉えられる. サブサンプリング層と畳み込み層の繰り返しは, 一般的には交互に起こり, 「バイピラミッド」となる. 各層において, 特徴マップの数は空間解像度が減少するにつれ増加する. 図 2 の 3 つ目の隠れ層のそれぞれのユニットは, 直前の層のいくつかの特徴マップからの入力接続を持つ. 畳み込み / サブサンプリングの組み合わせは, Hubel と Wiesel の「単純な」細胞と「複雑な」細胞という概念に着想を受け, 福島の Neocognition で実装されたが, 当時は逆誤差伝搬法といった大域的に学習可能な手法は存在しなかった. 表現の豊かさ (特徴マップの数) を徐々に増加させ, 空間解像度を徐々に減らすことで, 入力の幾何学的変換に対して高い不変性を持たせることが可能である.

すべての重みは逆誤差伝搬法により学習されるため, 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みなすことができる. 重みの共有技術は, 自由なパラメータを減らすという興味深い側面の効果を持ち, それにより計算機の容量を減らし学習時のエラーと検証時のエラーの差を減らすことができる. 図 2 のネットワークは 340908 個の接続を持つが, 重みの共有により学習可能な自由パラメータは 60000 個である.

サイズが固定された畳み込みネットワークは手書き文字, 機会印刷文字, オンライン手書き文字, 顔 など様々な認識アプリケーションに採用されてきた. 1 つの時間次元に沿って重みを共有するサイズ固定畳み込みネットワークは Time-Dealy Neural Network (TDNNs) として知られている. TDNN は音声認識 (サブサンプリング無し) , 会話中単語認識 (サブサンプリングあり) , 分離された手書き文字のオンライン認識, 署名検証に用いられてきた.

2.2 B. LeNet-5

本節では, 実験で用いた畳み込みニューラルネットワークである LeNet-5 も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 層から構成され, それらすべての学習可能なパラメータ (重み) が設定されている. 入力は 32 × 32 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 (28 × 28 の領域を中心とした最大 20 × 20 のサイズ) よりはるかに大きい. その理由として, ストロークの終点や角といった潜在的な特徴量が特徴抽出器の受信可能領域の中心に現れることが望ましいからだ. LeNet-5 では, 最後の畳み込み層 (後述する C3) の受信領域の中心の集合は, 32 × 32 の入力の中心に 20 × 20 の領域を形成する. 入力画素の値は, 背景レベル (白) が -0.1 に, 前景レベル (黒) が 1.175 に一致するように標準化されている. この標準化により, 入力の平均はおおよそ 0 , 標準偏差はおおよそ 1 になり高速に学習が可能となる.

以下, 畳み込み層は Cx , サブサンプリング層は Sx , 全結合層は Fx とする. ここで, x は層のインデックスである.

C1 は 6 つの特徴マップを持つ. それぞれの特徴マップ内のユニットは, 入力の 5 × 5 近傍に接続されている. 特徴マップのサイズは 28 × 28 であり, 入力の接続が境界から外れることを防いでいる. C1 は 156 個の学習可能なパラメータを持ち, 122304 個の接続を持つ.

S2 は 14 × 14 サイズの特徴マップを 6 個持つ. それぞれの特徴マップ内のユニットは, C1 の対応する 2 × 2 近傍に接続される. S2 内のユニットの 4 つの入力は加算され, 学習可能な係数が乗算され, 学習可能なバイアスが加算され, 結果をシグモイド関数に渡す. S2 の特徴マップは C1 に比べ 行と列の数が半減する. S2 は 12 個の学習可能なパラメータ, 5880 の接続を持つ.

C3 は 16 個の特徴マップを持つ. それぞれの特徴マップのユニットは, S2 の特徴マップの一部分と同位置の 5 × 5 近傍に接続される. 表 1 に C3 の各特徴マップと接続している S2 の特徴マップの集合を示す. なぜすべての S2 の特徴マップと C3 の特徴マップを接続しないのか? その理由は 2 つある. 1 つ目に, 完全に接続しない方式により接続する数を現実的な領域に落とし込むことができる. より重要なことはネットワークにおいて対称性を崩すことができる. 異なる特徴マップは, 異なる入力群を得るた

めそれぞれ異なる特徴 (できれば補完的な) を抽出する. 表 1 に示した接続の根拠を以下に示す. C3 の最初の 6 つの特徴マップは 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 つの特徴マップは S2 の 4 つの連続する特徴マップの部分集合を入力として得る. 次の 3 つの特徴マップは, S2 の 4 つの 不連続な特徴マップの部分集合を入力として得る. 最後の 1 つは S2 のすべての特徴マップから入力を得る. C3 は 1516 個の学習可能なパラメータと, 2000 個の接続を持つ.

S4 は 5 × 5 のサイズの特徴マップを 16 個持つ. 各特徴マップのユニットは, C1 , S2 と同様に, C3 の対応する特徴マップの 2 × 2 近傍に接続される. S4 は 32 個の学習可能なパラメータと 2000 個の接続を持つ.

C5 は 120 個の特徴マップを持つ. 各ユニットは S4 の特徴マップ 16 個すべての 5 × 5 近傍に接続される. ここで, S4 のサイズが 5 × 5 であるため, C5 の特徴マップは 1 × 1 である. これは S4 と C5 が完全に接続されていることに相当する. LeNet-5 の入力を大きくして他すべてを一定にすると, 特徴マップの次元は 1 × 1 より大きくなるため, C5 は全結合層ではなく畳み込み層としている. この畳み込みネットワークの次元を動的に増加させる方法は, 第 7 節で述べる. C5 は 48120 個の学習可能な接続を持つ.

F6 は 84 個のユニット (84 と決定した理由は出力層の設計による) を含み, C5 と完全に接続している. F6 は 10164 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と同様に, F6 までの層のユニットは, 入力のベクトルと重みのベクトルの内積を計算し, バイアスを加算する. このユニット i の重み付き和 a_i を, シグモイド型活性化関数に通し, x_i を得る.

$$x_i = f(a_i) \tag{5}$$

活性化関数は, 双曲線正接関数で標準化される.

$$f(a) = A\tanh(Sa) \tag{6}$$

ここで, A は関数の振幅, S は原点における傾きを表す. 関数 f 破棄関数であり, 水平方向における漸近線は $+A$, $-A$ である. 定数 A は 1.71519 と決まっている. この活性化関数を選んだ根拠は付録 A に示している.

最後に出力層は, ユークリッド基底関数ユニット (RBF) から構成され, 各クラスごとに 84 個の入力を持つ. それぞれの RBF ユニットの出力 y_i は次のように計算される.

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

つまり, 各 RBF ユニットの出力は, 出力ベクトルとパラメータベクトルとのユークリッド距離を計算する. 入力がパラメータベクトルから離れれば離れるほど, RBF の出力の値も大きくなる. 特定の RBF ユニットの出力は, RBF に関連するクラスのモデルと入力パターンとの合致度合いを測定するペナルティ項と解釈できる. 確率論的な用語では, RBF の出力は, F6 の 構成空間におけるガウス分布の非正規化負対数尤度と解釈できる. 入力パターンが与えられた時, 損失関数は F6 の構成について, パターンに対応する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づくように設計するべきである. 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. パラメータベクトルの構成要素は -1 から +1 の範囲で設定された. -1 と +1 を等確率でランダムに選ぶ, また [47] で提案されたように誤り訂正符号を構成するように選ぶこともできたが, その代わりに 7 × 12 のビットマップに描かれた対応する文字クラスに一致させるように様式化された画像を表現するように表現した (このため入力は 84 個となった) . このような表現は分離された数字の認識において時に有用というわけではないが, 出力可能な ASCII データから取り出せる文字列を認識するにはとても有用である. 大文字の O と小文字の o と数字の 0 や, 小文字の l , 数字の 1 , 大文字の I といった似ていて混同しやすい文字は出力コードは似てくるため. このような混同を修正する言語的な後処理層装置と組み合わせた際に特に有効である. なぜなら, 混同可能なクラスの符号が類似しているため, 曖昧な文字に対応する RBF の出力も類似し, 後勝利装置が適切な解釈を選択できるようになる. 図 3 に ASCII セットに対する出力を示す.

出力に一般的な "1 of N" 符号 (プレイス符号, グランドマザーセル符号とも呼ばれる) ではなく, このような分散的な符号を用いるもう一つの理由として, クラスの数が数十といったオーダーより大きくなると, 非分散符号の動作が悪くなる傾向があるからである. 出力ユニット内の非分散符号は, ほとんどの時間非活性にならなければならない. 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, 分類器は文字だけでなく文字以外の認

識にも良く用いられるからである. シグモイドと異なり, 分散的な符号を持つ RBF は非典型パターンが外部に落ちやすいようによく囲われた入力の領域で活性化されるためこの目的に適している.

RBF のパラメータベクトルは F6 のターゲットベクトルの役割を担う. これらのベクトルの成分は +1 あるいは -1 であり, F6 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. 実際, +1 と -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 のユニットは最大限の非線形の範囲で動作する. シグモイドの飽和は, 損失関数の悪い条件付け, 収束が遅くなることにつながるため避けなければならない.

2.3 C. 損失関数

上記のネットワークで使用できる最も簡潔な出力損失関数は, 最尤推定 (MLE) であり, 本研究では 最小平均二乗誤差 (MSE) と等価である. 学習サンプルの集合に対する基準は簡単である.

$$E(W) = frac1P \sum_{p=1}^{P} y_{D^p}(Z^p, W) \tag{8}$$

ここで, y_{D^p} は D^p 番目の RBF ユニットの出力, 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, 3 つの重要な特性が抜けている. 第 1 に, RBF のパラメータを適応させる場合, $E(W)$ は些細で到底受け入れがたい解となる. この解では, すべての RBF のパラメータベクトルが等しく, F6 の状態が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, すべての RBF の出力は 0 に等しくなる. この破壊的な現象は RBF の重みを適用させようとしなければ起こらない. 2 つ目の問題は, クラス間で競合が起こらないことである. このような競合は, HMM の学習に時々使用される最大相互情報量に類似した MAP (maximun a posteriori) と呼ばれるより識別性能が高い学習基準を使用することで得ることができる. これは, 入力画像がクラスのうち 1 つからくる, あるいは背景を表す 「ゴミ」 クラスラベルから来るとすると, 正しいクラス D^p の事後確率を最大化 (あるいは正しいクラスの確率の対数を最小化) に相当する. ペナルティの観点からは, MSE のように正しいクラスのぺねるてぃを下げるだけでなく, 異なるクラスのペナルティを引き上げることができる.

$$E(W) = frac1P \sum_{p=1}^{P} (y_{D^p}(Z^p, W) + \log(e^{-j} + \sum_i e^{-y_i(Z^p, W)})) \tag{9}$$

第 2 項の負は競合の役割を果たす. 第 2 項は, 第 1 項　に比べ小さい (あるいは等しい) ことが必要であるため, この損失関数は正である. 定数 j が正の値であり, すでにペネルティがかなり大きいクラスのペネルティをさらに引き上げるのを防ぐ. 「ゴミ」 クラスラベルの事後確率は, $\log(e^{-j} + \sum_i e^{-y_i(Z^p, W)})$ の割合である. この基準では RBF パラメータを学習するときに, RBF の中心を互いに離すことで, 先述した破壊的な影響を防ぐことができる. 第 6 節では, 入力中の複数の物体 (例えば, 文書や単語中の文字) を分類するよう学習するシステムに対するこの基準の一般化を紹介する.

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, 誤差逆伝搬法を使用する. 標準的なアルゴリズムは重みの共有を考慮するためにわずかに修正する必要がある. これを実装する簡単な方法は, ネットワークが重みの共有がない従来の多層ネットワークであるかのように各接続に対して損失関数の偏導関数を計算することである. 次に, 同じ重み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, そのパラメータに対しての導関数を形成する.

このような大規模なアーキテクチャはかなり効率的に学習することができるが, そのためには付録で示すいくつかの技術を使用する必要がある. 付録の第 A 節では, 使用したシグモイドや重みパラメータの初期化などの詳細を示している. また, 第 B, C 節では使用した最小化の方法を示す.

3　3. 結果と他手法との比較

12 / 29 要約でよいといわれたので要約にします. 数字の認識タスクは, 形状認識手法を比較するためのベンチマークとして優れている. 本論文ではサイズ正則化された画像に対して直接的に動作する適応的な手法に焦点を当てている.

3.1 A. データベース: 修正した NIST セット

本論文でシステムの学習, 検証で用いるデータベースは NIST の Special Database 3 と Special Database 1 の手書き整数の 2 値画像である. SD-1 と SD-3 には手書き文字の書き手やデータのスクランブルといった差があり, 学習データとテストデータの選択に実験結果が依存しないようにするために, NIST のデータセットを混合して新たなデータセット Modified NIST , MNIST を構築した. 実験では, 10000 個のテスト画像, 60000 個の学習データ画像を用いた.

元の 2 値画像は縦横比を保ったまま 20 × 20 ピクセルに収まるように標準化され, 標準化の過程でグレーレベルが含まれる. データセットは以下の 3 つのバージョンを用いた. 1 つ目は, ピクセルの重心を計算し画像を平行移動することで 28 × 28 の領域の中心に配置した. これをレギュラーデータセットと呼ぶ. 2 つ目は, 文字画像の画角を補正し 20 × 20 ピクセルの画像にトリミングする. 画角補正では, ピクセルの 2 次慣性モーメントを計算し, 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. 画角補正済データセットと呼ぶ. 初期の実験で用いられていた 3 つ目のデータセットは, 画像は 16 × 16 に縮小される. 図 4 にテストデータからランダムに抽出した例を示す.

3.2 B. 結果

レギュラーデータセットを用いて, 複数バージョンの LeNet-5 を学習させた. 各学習段階において学習データを 20 回反復した. 学習率 η は以下に示すように減少していく. 最初の 2 回は 0.0005 , 次の 3 回は 0.0002, 次の 3 回は 0.0001, その次の 4 回は 0.00005, それ以降は 0.00001 とした. 実験により過学習は起こらなかったが, 事前に計算した有効学習率の範囲内で学習率を比較的大きな値にしていたことが要因と考えられている.

学習データの大きさの影響を 15000, 30000, 60000 として測定した. 図 6 に学習時のエラーとテスト時のエラーを示す. LeNet-5 のような特殊なアーキテクチャでも, 学習データを増やせば精度が向上することがわかる.

この仮説を検証するために, 学習用画像をランダムに歪ませることでデータ拡張を行い, 元の 60000 個の画像に加え, 540000 個の学習データを追加した. データ拡張は図 7 に示すようにアフィン変換を適用した. 学習パラメータを変更せず, データ拡張を行った場合, テスト時のエラーは 0.95% から 0.8% に減少する. この 20 回の学習の繰り返しで, ネットワークは各サンプルを 2 回しか観測していない. 図 8 に誤分類されたテスト例を示している. 例の中では, 人間から見ても曖昧なものもあれば, 容易に判別可能な例もある.

3.3 C. 他の分類器との比較

比較のために同じデータセットに対して様々な分類器を学習し, テストした.

3.3.1 C.1 線形分類器とペアワイズ分類器

最も単純な分類器として考えられるのが, 各入力画素値は, 各出力ユニットの重み付き和に寄与し, 最大値を持つ出力ユニットが入力文字のクラスを示す線形分類器である. レギュラーデータセットでは, テスト時のエラーは, 12 % , ネットワークは 7850 個の自由パラメータを持つ. 画角補正済みデータセットでは, テスト時のエラーが 8.4 % でネットワークは 4010 個の自由パラメータを持つ.

線形分類器を単純に改良したものが, ペアワイズ分類器である. ペアワイズ分類器は各クラスを他のクラスから分離するために各ユニットを個別にラベル付けして学習したものであり, レギュラーデータセットではテスト時のエラーは 7.8 % まで減少している.

3.3.2 C.2 ベースライン : 近傍分類器

もう 1 つの単純な分類器として, 入力画像間のユークリッド距離を用いた K 近傍分類器がある. この分類器は, 学習や設計が必要ではないという長所があるが, メモリ容量と認識にかかる時間が大きくなる欠点がある. レギュラーデータセットにおいてはテスト時のエラーは 5.0 % であった. 画角補正済みデータセットの場合, $k = 3$ の時にテスト時のエラーは 2.4 % となった. 本研究で示す他のシステムはすべて画素を直接処理しているため, この分類器はベースラインとして適している.

3.3.3 C.3 主成分分析 (PCA) と 多項式分類器

入力パターンを学習ベクトル集合の 40 個の主成分に投影するように前処理し, 主成分を計算した. 得られた 40 次元の特徴ベクトルは多項式分類器の入力として使用し, 結果としてこの分類器は前もって入力変数の組の積を計算しておく 821 個の入力を持つ線形分類器とみなすことができる. レギュラーデータセットにおけるテスト時のエラーは 3.3 % であった.

3.3.4 C.4 放射基底関数 (RBF) ネットワーク

第 1 層が 28 × 28 入力の 1000 個のガウス型 RBF ユニットからなり, 第 2 層が単純な 1000 個の入力, 10 個の出力を持つ線形分類器となる RBF ネットワークを構築した, RBF ユニットは 100 個ずつ 10 グループに分けられ, 適応的な k-means 法を用いて 10 クラスのうちの 1 つずつすべての学習データについて RBF のユニットをグループごと学習した. レギュラーデータセットにおいてテスト時のエラーは 3.6 % であった.

3.3.5 C.5 1 つの隠れ層を持つ完全結合多層ニューラルネットワーク

誤差逆伝搬法を用いて学習した 1 層の隠れ層を持つ多層ニューラルネットワークを構成し, 検証した. レギュラーデータセットにおいてテスト時のエラーは 300 個の隠れユニットの場合は 4.7 % , 1000 個の各ユニットの場合は 4.5 % となった. データ拡張を行った場合, 300 個の隠れユニットでは 3.6 %, 1000 個の隠れユニットの場合は 3.8 % とわずかな改善しか得られなかった. 画角補正済みデータセットにおいてテスト時のエラーは 300 個の隠れユニットで 1.6 % に減少した. このような多くの自由なパラメータを持つネットワークで低いテスト時の誤差が得られる原因として, 筆者たちは多層ニューラルネットワークにおける勾配降下法の挙動に「自己正則化」の効果があると推測しているが, まだ理論的な理解や実証的な根拠が必要であるとしている.

3.3.6 C.6 2 つの隠れ層を持つ完全結合多層ニューラルネットワーク

28 × 28 - 300 - 100 - 10 ネットワークにおけるテスト時のエラーが 3.05 % となり 1 つの隠れ層の場合と比べかなり良い結果を得られた. さらに 28 × 28 - 1000 - 150 - 10 の場合では 2.95 % とわずかに改善されただけだった. データ拡張を行った場合, 28 × 28 - 300 - 100 - 10 ネットワークでは 2.50 % , 28 × 28 - 1000 - 150 - 10 の場合では 2.45 % と若干の改善が見られた.

3.3.7 C.8 小規模な畳み込みネットワーク : LeNet-1

畳み込みネットワークは, 学習データを十分に学習できない小さなネットワークと過度にパラメータ化された大きいネットワークのジレンマを解決するための試みである. 入力画像は 16 × 16 ピクセルに縮小され, 28 × 28 の入力層の中央に配置された. LeNet-1 の評価には 100000 ステップの乗算加算が必要になるが, 畳み込み処理の性質上, 自由パラメータは約 2600 個に留まる. LeNet-1 ではテスト時のエラーが 1.7 % となり, パラメータが少ないネットワークで良い結果を残せていることは LeNet-1 のアーキテクチャがタスクに適していることを示している.

3.3.8 C.8 LeNet-4

LeNet-1 と比較して大規模な学習データを最適に利用するために設計されたのが LeNet-4 である. LeNet-4 では, 4 つの特徴マップとそれに続く 8 つのサブサンプリングマップが第 1 層の各特徴マップに対として結合し, 次に 16 個の特徴マップと 16 個のサブサンプリングが続き, 17000 個の自由パラメータを持っている. テスト時のエラーは 1.1 % であった.

3.3.9 C.9 ブーストした LeNet-4

複数の分類器を結合する「ブースティング」の手法がある. 1 つ目のネットワークは通常通り学習し, 2 つ目は 1 つ目のネットワークによって 1 番目のネットワークが正解したパターンと間違えたパターンが 50 % づつ混在するようにフィルタリングされたパターンを学習する. 3 つ目のネットワークは 2 つのネットワークが間違えた新しいパターンで学習される. データ拡張し実験した結果, テスト時のエラーは 0.7 % とどの分類器よりも優れていた. また, 計算コストも 1 つのネットワークの時に比べ約 1.75 倍であった.

3.3.10 C.10 タンジェント距離分類器 (TDC)

TDC は入力画像の歪みや変換に敏感な距離関数を配置した最近傍法である. 画像を高次元の画素空間の点と考えると, 歪みは空間内の多様体を意味し, この多様体はタンジェント平面と呼ばれる平面で近似できる. 文字画像の「近さ」は平面の距離で表される. 16 × 16 の画像を用いた学習では, テスト時のエラーは 1.1 % となった.

3.3.11 C.11 サポートベクターマシン (SVM)

SVM は高次元区間の複雑な局面を表現する際に非常に優れた方法である. 通常の SVM を用いた場合, レギュラーデータセットを用いた学習におけるテスト時のエラーは 1.4 % であった. 通常の SVM を用いた時, Burges と Schölkopf によってレギュラーデータセットを用いてテスト時のエラーは 1.4 % という結果が得られた. その後, Schölkopf は V-SVM という SVM の改良版を用いて 0.8 % という結果を得た. V-SVM は非常に計算コストが高いため, Burges は RS=SVM という手法を提案しレギュラーデータセットで 1.1 % を記録した.

D. 考察

図 9　12 に分類器の性能の概要を示す. 図 9 では 10000 件のテストデータに対するエラーを示しており, ブースティングを行った LeNet-4 が 0.7 % と最も良い結果を残し, LeNet-5 の 0.8 % がそれに続いた.

図 10 では, エラー 0.5 % を達成するために棄却しなければならないデータ数を示している. 多くのアプリケーションは テスト時のエラーよりもこの指標の方が重要である. ここでも ブースティングされた LeNet-4 が最も良い性能を残した. LeNet-4 の改良版はテスト時のエラーはほぼ同一だったものの, この指標では LeNet-4 より優れた結果を残した.

図 11 は各手法について 1 枚のサイズが標準化された画像を認識するために必要な積和演算の数を示したものである. ニューラルネットワークはメモリベースの方式に比べてはるかに負担が少なく、畳み込みニューラルネットワークはその規則的な構造と重みのための必要なメモリが少ないことから積和演算の数も少ないことがわかる.

学習時間も計測した. k 最近傍法と TDC はほぼ学習時間は 0 であった. 一方で, 1 つの層からなるネットワーク, ペアワイズネットワーク, PCA + 2 次ネットワークは 1 時間未満で計算できる一方で, 多層ニューラルネットワークはより長い時間かかると予想されましたが, 実際には学習セットを 10　20 回繰り返すだけでした. 学習時間という指標は, 開発者にとっては重要ですが, システムのユーザーにはほとんど意味がありません.

図 12 には様々な分類器における記憶する必要のある変数の数で測定したメモリ要件を示している. ほとんどの手法は妥当な性能を得るために 1 変数当たり約 1 バイトしか必要としない. しかし, 最近傍法では 1 画素当たり 4 ビットのメモリで十分な性能を得ることができる. 当然, ニューラルネットワークはメモリベースの方法よりもはるかに少ないメモリしか必要としない.

分類器の全体的な性能は, 精度, 実行時間, 必要なメモリなど多くの要因に依存する. 1989 年の段階では, LeNet-5 のような複雑な分類器は数週間物学習が必要であり, 検討されていなかった. LeNet-1 から様々な分類器が考案されたものの, 様々な学習機の性能の見積もりからより優れたニューラルネットワークアーキテクチャの期待が高まり, LeNet-4 や LeNet-5 が開発された.

また, ブースティングによりメモリと計算機のコストを抑えながら精度を向上させることができることがわかった. また, データ拡張により多くの元学習データが無くてもデータセットのサイズを増やすことができる.

サポートベクターマシンは問題に対する事前知識を含んでいないため優れた制度を実現しているが, 畳み込みニューラルネットワークに匹敵する性能に達するにはメモリと計算機にかなりのコストがかかる. 比較的新しい手法である縮小版 SVM はコストが畳み込みニューラルネットワークの 2 倍程度ではあるが, エラーの値は非常に近い.

多くの量のデータが利用可能な場合, 多くの手法で十分な精度を残すことができる. ニューラルネットワークを用いた手法はメモリベースの手法に比べて実行速度がかなり早く, コストも少ない. ニューラルネットワークを用いた手法の優位性は学習データの規模が増えるほどより顕著になる.

3.4 E. 不変性とノイズ耐性

畳み込みニューラルネットワークは実世界の文字認識システムにおけるヒューリスティックなセグメ

ンテーションによって生成されるサイズや位置が大きく変化する形状を認識するのに特に適している.

上記の実験ではノイズ体制や歪み不変性の重要性は明らかではない. 実際のアプリケーションにおいては全く異なる. 一般に文字は認識過程の前に分割する必要がある. しかし, セグメンテーションのアルゴリズムは通常完璧にはいかず, 文字画像に余計なマークが残ったり, 不完全な文字ができたりする. このような画像はサイズの正規化やセンタリングといった前処理ができない. そのため, 多くのシステムでは領域や単語のレベルで画像を正規化する. 本研究では, 上下のプロファイルを検出し, 一定の高さに正規化することで文字画像に余計なマークが浮き出ることが無くなるが文字の大きさや縦方向の位置のばらつきが大きくなる. そのため, このようなばらつきに強い認識器を用いることが望ましい. 図 13 に LeNet-5 が正しく認識した歪んだ文字の例を示す. スケールの変動は約 2 分の 1 まで, 縦方向のシフト変動は文字の高さの約半分まで, 角度は約 30 度まで正しく認識できると推測できる. このことは畳み込みニューラルネットワークが幾何学的な歪みに対する頑健性を持つ部分的な根拠になりうる.

図 13 には非常にノイズの多い状況下での LeNet-5 のロバスト性が示された例を含んでいる. LeNet-5 はこれらの乱雑な画像から顕著な特徴を抽出できるようである.

4. マルチモジュールシステムと GTN

先述した古典的な誤差逆伝搬法は, 勾配に基づく学習の単純な形態である. しかし, (4) 式で示される勾配逆伝搬アルゴリズムは, 線形層とシグモイド関数の交互の配置からなる単純な多層 FFN よりも一般的な状況を示していることは明らかである. 理論的には, 関数モジュールのヤコビアンと任意のベクトルとの積を計算できる限り, どのような配列の関数モジュールを通しても導出することができる. しかし, 大規模で複雑な学習システムは, 特化したモジュールから構築される必要がある. 単純な例は 畳み込み層やサブサンプリング層, 完全連結層, RBF 層からなる LeNet-5 がある. また, あまり一般的ではない例として正しいセグメンテーションが与えられずに単語の分割と認識を同時に行える単語認識システムがある.

図 14 は訓練可能なマルチモジュールシステムの例である. マルチモジュールシステムは, 各モジュールが実装する機能とモジュール間の相互接続のグラフによって定義される. グラフはモジュールが更新されなければならない順序を示している. 最も単純なケースでは損失関数は, 望まれる出力を得られるような画部の入力を受け取る.

A. オブジェクト指向のアプローチ

マルチモジュールシステムを実装する斉井, オブジェクト指向プログラミングは便利な方法である. 各モジュールはクラスのインスタンスであり, モジュールクラスは 「順伝搬」メソッドを持っている. 複雑なモジュールは抽象的なモジュールから新しいクラスを定義することにより構築される. このクラスの fprop メソッドは, 適切な中間状態変数や外部入出力を引数としてメンバモジュールの fprop めそっどをよびだすだけである. ここでは, 有向非巡回グラフの場合に限定して述べる.

マルチモジュールシステムにおける導関数の計算は簡単である. それぞれのモジュールのクラスに bprop と呼ばれる「逆伝搬」メソッドを定義できる. モジュールの bprop メソッドは fprop メソッドと同じ構成要素を持つ.

システムのすべての微分は, すべてのモジュールに対して bprop メソッドに対して順伝搬と逆順に呼び出すことで計算できる. 逆伝搬によりシステムのすべての状態変数とパラメータに関する損失関数 E の偏導関数が効果的に計算される. 順伝搬と逆伝搬の間には興味深い二元性がある.

導関数が逆伝搬により計算できることは直感的に理解しやすい. 理論的に正当化するには, ラグランジュ関数を用いる方法がある. 再帰的な接続を持つネットワークに拡張するために使用される.

B. 特殊モジュール

ニューラルネットワークや他の標準的なパターン認識手法は, 勾配に基づく学習で学習したマルチモジュールシステムとして定式化できる. 一般的に使用されるモジュールには, 行列積やシグモイド関数などがあり, これらを組み合わせることで従来のニューラルネットワークを構築できる, その他も, 畳み込み層, サブサンプリング層, RBF 層などがある. 損失関数も 1 つのモジュールとして表現され, 一般的に

使用されるモジュールは bprop メソッドを持つ. 一般的には, 関数 F の bprop は, F のヤコビアンとの乗算である. 興味深いことに, ある種の微分不可能なモジュールはマルチモジュールシステムに悪影響を与えることなく挿入することが可能である. 例として, マルチプレクサモジュール, min モジュールがあり, これらはある条件下においては微分可能であるため, 勾配に基づく学習アルゴリズムでも収束が保証されている.

オブジェクト指向でのマルチモジュール実装は 2 次導関数のガウス-ニュートン近似を伝搬する bbprop メソッドを含むように簡単に解決できる.

マルチプレクサモジュールは, 一般的なシステムのアーキテクチャが入力データに応じて動的に変化する特殊なケースであり, 新しい入力パターンごとにアーキテクチャを再構成するために使用することができる.

3.5 C. GTN

マルチモジュールシステムは大規模な学習可能システムを構築するための非常に柔軟な道具である. しかし, 前節まではパラメータと状態情報が組となって固定サイズのベクトルでモジュール間を通信していることが前提となっていた. 固定長のベクトルでデータを表現する場合には柔軟性に制約がかかり, このことは多くのアプリケーション, 特に連続的な音声認識や手書き単語認識といった入力長が変化するタスクや, 数や性質が変化する物体や形状を符号化する必要がある場面解析や複合物認識といったタスクで深刻な欠点となっている.

より一般的には, 固定サイズのベクトルは, ベクトルやシンボルの系列に対する確率分布を符号化する必要があるタスクには柔軟性が欠ける. このような系列の分布は, 確率的文法, より一般的な枝にベクトルが含まれる有向グラフで表現される. グラフの各パスは異なるベクトル列を表す. 各枝に関連するデータの要素を確率分布のパラメータとして解釈することで系列に対する分布を表現できる. 系列上の分布は音声認識システムや手書き文字認識システムにおける言語知識のモデリングにおいて特に便利である. グラフ内の各パスは入力の代替解釈を表す.

本研究では大規模な手書きシステムを構築する際に, システムを 1 つ以上のグラフとして受け取り出力としてグラフを生成するモジュールのネットワークとすることで簡単かつ迅速に開発や設計できることを発見した. このようなモジュールをグラフトランスフォーマーと呼び, 出来上がった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.

統計的な観点から見ると, 従来のネットワークにおける固定サイズの状態ベクトルは状態空間における分布の平均を表している. 状態が可変長である固定サイズのベクトルは, 固定サイズのベクトルの可変長の系列に対する確率分布の平均とみることができる. GTN では状態はグラフとして表現され, 構造化されたベクトルの系列の確率分布の混合として見ることができる.

勾配に基づく学習法は, 固定サイズのベクトルで通信する単純なモジュールのネットワークに限定されないが, GTN に一般化することができる. グラフトランスフォーマーによる勾配逆伝搬は, 出力グラフの数値情報に対して勾配を取り, 入力グラフとモジュール内部のパラメータの数値情報に対して勾配を計算する. 勾配に基づく学習法は勾配計算の際の関数が, 微分可能であれば適用可能である.

また, 一般的に混合して用いる文書処理システムなどのシステムの多くのモジュールで実装されている関数がその内部パラメータや入力に対して微分可能であり, 大域的に学習可能なシステムとして使用できる.

この 2 つのことを以下の説で示す. その際, あえて確率論への言及を避ける.

5. 複数物体認識: ヒューリスティックオーバーセグメンテーション

手書き文字認識における最も難しい問題の 1 つは, 分離した文字だけでなく, 郵便番号や小切手の金額, 単語といった文字列も認識することである. ほとんどの分類器は 1 度に 1 つの文字しか扱うことができないため文字列を個々の文字画像を分割する必要があるが, 自然に書かれた文字列を確実に成形された文字に分割する画像解析技術を発案するのはほぼ不可能である.

認識器は個々の文字を認識するためだけでなく, 誤って分割された文字を拒否することもでき全体の誤差を最小にできる.

本節と次節では, 単語や小切手などの文字列を読み取るための GTN の簡単な例について説明する.

この方法ではセグメンテーションの結果を確認する作業を避けることができる.

A. セグメンテーショングラフ

単語のセグメンテーションと認識のための古典的な方法は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手法である. 他の手法と比べて, 多数の異なるセグメンテーションを考慮することでセグメンテーションに関する難しい決定を避けることができるという利点がある. この手法では, ヒューリスティックな画像処理技術を用いて単語, 文字列のカットの候補を見つけ次に認識器を使用して生成されたセグメンテーションを採点する. カットの候補では「正しい」カットが含まれることを期待して必要以上の候補を生成する. セグメンテーショングラフは, 開始ノードと終了ノードを持つ有向非巡回グラフであり, 各ノードはカット候補に関連付けられており, 枝はカット間の画像に対応している. グラフを通る完全なパスは, 文字列を形成するためのセグメンテーションの断片を関連付ける方法に対応している.

B. 認識変換器とビタビ変換器

文字列を認識するための簡単な GTN を 図 17 に示す. これは認識器 T_{rec} , T_{vit} の 2 つのグラフトランスフォーマーからから構成されている. 認識変換器は, 解釈グラフあるいは認識グラフ G_{int} を生成することが目標であり, これは入力のすべての可能な分割すべての解釈を含むものである. G_{int} の各パスは入力におけるう　1 つの可能な解釈を示す. ビタビ変換器の役割は解釈グラフから最適な解釈を抽出することである.

認識変換器 T_{rec} は, セグメンテーショングラフ G_{seg} を入力としてセグメンテーショングラフの各枝に関連する画像に単一文字用の認識器を適用する. 解釈グラフ G_{int} は, 各枝が同じノードからの枝と同じノードへの枝のセットに置き換えられることを除いて, G_{seg} とほとんど同じ構造を持っている. G_{int} の枝は G_{seg} 内の枝に関連する画像の可能なクラスごとの数値を持つ. 図 18 を示すように各枝にはクラスラベルと, 認識器が示すように画像がクラスラベルへが示すクラスへ属するという制約が設けられている. セグメンテーション機能によって分割候補に対する制約が計算されると, これらの制約は文字認識機能によって計算された制約と組み合わせて解釈グラフの枝に対する制約が求められる. 異なる性質の制約を組み合わせることは非常にヒューリスティックであるが, GTN の学習過程では, 制約を調整し, 制約の組み合わせを利用する. 解釈グラフの各パスは, 入力された単語の解釈の可能性に対応する. ある分割に対応する特定の解釈の制約は解釈グラフの対応するパスに沿った枝の制約値の合計で与えられる. ある解釈の制約値を分割とは無関係に計算するなら, その解釈を持つすべてのパスの制約を結合する必要がある. 並列パスの制約を結合するための適切な規則は第 7 - C 節で述べる.

ビタビ変換器は 1 つのパスをもつグラフ G_{vit} を生成する. このパスは解釈グラフの累積制約値が最小のパスである. 認識結果はビタビ変換器によって抽出されたグラフ G_{vit} に沿った枝のラベルを読み取ることで生成することができる. ビタビ変換器は, グラフ内の最短経路を効率的に求める動的計画法の原理を応用した有名なビタビアルゴリズムから名付けられている. ソースノード s_i とデスティネーションノード d_i を持つ枝 i に関連するペナルティを c_i とする (2 つのノード間に複数の枝が存在することに留意する). 解釈グラフでは, 枝もラベル l_i を持つ. ビタビアルゴリズムは以下のように進行する. 各ノード n にはビタビ制約の累積値 v_n が設定される. これらの累積制約値は $v_{start} = 0$ で初期化される. 他のノードの累積制約値 v_n はその親ノードの v 値から上流の枝 $U_n = \mathrm{arc}i\mathrm{withdestination}d_i = n$: を介して再帰的に計算される.

$$v_n = ¥\, undersetmini \in U_n(c_i + v_{s_i}). \qquad (10)$$

そのうえ, 右辺を最小化する各ノード n の i の値は最小化する有向枝を m_n と示す. 終点ノードに到達したとき v_{end} で, 制約値の合計が最小となる経路の制約値の合計を求める. この制約値をビタビペナルティと呼び, この枝とノードの系列をビタビパスと呼ぶ. ノード $n_1...n_T$ と枝 $i_1...i_T$ を持つビタビパスを得るには, これらのノードと枝を次のように辿る. 終点ノード n_T から始めて開始ノードに達するまで最小の枝: $i_t = m_{n_t+1} and n_t = s_{i_t}$ を再帰的に辿る. そしてビタビパスの枝からラベルの列を読み取ることができる.

4 グラフ変換ネットワークのための大域的な学習

本節では正しく分割された文字列における正しいクラスラベルには低い制約値を, 違うクラスラベルには高い制約値を, 正しく分割されなかった文字列に対してはすべてのクラスラベルについて高い制約値を課すような文字列レベルでの認識システムの学習方法について説明する, 多くのアプリケーションでは各モジュールを別々に学習させるためにヒューリステックオーバーセグメンテーションのような多くの事前知識が用いられるが, これらを用いた個別学習は最適ではないことが知られている. 以下の節では GTN ベースの手書き文字認識器を文字列レベルで学習するための 3 つの異なる勾配に基づく学習法について説明する. 各手法はそれぞれビタビ学習, 判別ビタビ学習, フォワード学習である. また, 判別フォワード学習もあるがこれは第 2 - C 商で紹介した MAP をグラフシステムに一般化したものである. 本研究では確率的な解釈に頼らず, 勾配に基づく学習における識別学習は広範な誤差訂正学習の原理の単純な一例であることを示す.

HMM のようなグラフベースの系列認識システムのための学習方法は音声認識の分野で広く研究されており, これらの方法はシステムがデータの確率的生成モデルに従い, 可能な入力系列の空間において正規化された尤度を提供する. 一般的な HMM 学習はこの正規化に依存しているため, ニューラルネットワークのような非生成モデルを組み込むと正規化を維持できない. この場合には識別的学習法など他の手法を用いてニューラルネットワーク/ HMM 音声認識器を単語や文章レベルに適用する方法が提案されている.

他の大域的な学習可能な系列認識システムはグラフベースに頼らずに統計的モデリングの難しさを避けている. その最たる例が RNN であるが, 勾配に基づく学習を用いた RNN の学習は非常に困難であることが判明している. 以下に示す GTN 技術は音声認識のために開発された大域的な学習法を一般化したものである.

A. ビタビ学習

認識時にはビタビアルゴリズムによりグラフの制約値の最も低い経路が正しいラベル列と関連していることが望ましい. よって最小化すべき損失関数は最も低い制約値を持つ正しいラベル列の関連するパスの学習データセットの平均となる. 学習ではこの制約値の平均を最小化するような認識器のパラメータの集合を見つけることが目的となり, 損失関数の勾配は図 19 に示す GTN アーキテクチャを介した逆伝搬法により計算される. 図 19 におけるパスセレクタと呼ばれるグラフ変換器は, 解釈グラフと最適なラベル列を入力として, 解釈グラフから相関のあるラベル列を含むパスを抽出する. その出力である制約付き解釈グラフ G_c は正しいラベル列に対応するすべてのパスを含み, ビタビ変換器の入力となり 1 つのパスをもつ G_{cvit} となる. 最後にパススコア変換器が G_{cvit} を入力として累積制約値 C_{cvit} を計算する. この GTN の出力は現在のパターンに対する損失関数である.

$$E_{vit} = C_{cvit} \tag{11}$$

このシステムでは希望するラベルの並びのみ必要な情報であり, 正しいセグメンテーションに関する知識は必要ない.

第 6 節で説明したように, GTN のアーキテクチャで勾配を逆伝搬する過程において先攻するモジュールで勾配を計算したのちに勾配は GTN のすべてのモジュールを通して逆伝搬させる必要がある. パススコア変換器においては, G_{vit} 上の個々の制約に関する損失関数の偏導関数は損失関数が制約の和であることから 1 に等しいため単純である. ビタビ変換器においては, G_c の枝の制約に関する E_{vit} の偏導関数は, G_{cvit} に現れる枝については 1, そうでなければ 0 となる. ビタビ変換器のような本質的に不連続な関数を逆伝搬させても良い理由としてはビタビ変換器が min 関数と加算器をまとめたものであるからである. 第 6 節で min 関数で勾配を逆伝搬しても悪影響がないことを述べた. パスセレクタにおいては, ビタビ変換器と同様である. 認識変換器においては, 順伝搬ではセグメンテーショングラフ内の各枝に 1 つずつインスタンスが生成され, G_{int} の各枝の制約値はインスタンスの出力により生成され, インスタンスは各出力に対する勾配を持つ. 逆伝搬では各インスタンスを通して勾配を逆伝搬することができる. 各インスタンスにおいてパラメータに関する損失関数の偏微分のベクトルを得る. すべての認識器インスタンスは同じパラメータベクトルを共有するため, 認識器の完全な勾配は単純に各インスタンスによって生成される勾配ベクトルの和となる.

一見シンプルに見えるが, 致命的な欠陥がある. 第

2 - C 節で述べたように認識器がシグモイド出力ユニットを持つ単純なニューラルネットワークであるときには損失関数の最小値は, 入力を無視して出力をすべての成分について小さな値を持つ一定のベクトルに設定することで達成される. この崩壊問題において, 厳密には上記のような完全な崩壊は起こらないがより緩やかな崩壊は完全に防ぐことができない, RBF のパラメータが適用可能であるなら, ニューラルネットワークが緩やかな崩壊を起こすベクトルを生成することを学習してしまう. このような崩壊はニューラルネットワークのような学習可能なモジュールが　RBF に入力した場合のみ発生する. また, ビタビ学習では制約値が低い競合回答が考慮されないため, 解答の性能として信頼性が低くなってしまう問題もある.

B. 判別型ビタビ学習

危険なほど低い制約値をもつようなおそらく間違えているパスのペナルティを上げるといったように学習基準を修正することにより, 上記の崩壊問題を回避すると同時により信頼性の高い評価値を生成できる. このような基準は, 識別的と呼ばれ個々のクラスを独立にモデル化するのではなく, クラス間に適切な分離面を構築しようとするものである. 識別基準の一例として, 所望の出力を満たすグラフにおけるビタビパスの制約値と, 解釈グラフにおけるビタビパスとの制約値の差, すなわち最良の正しいパスの制約値と, 最良の正しいかどうか不明なパスの制約値の差が挙げられる. 図 20 に対応する GTN アーキテクチャを示す. 非識別学習では図 20 の左半分で計算された所望の出力と図 20 の右半分で計算された実際の出力の差を最小化する.

判別ビタビ損失関数を E_{dvit} とし, 所望の出力を得るグラフのビタビパスの制約値を C_{cvit}, 解釈グラフのビタビパスの制約を C_{vit} と呼ぶことにする.

$$E_{dvit} = C_{cvit} - C_{vit} \tag{12}$$

なお, E_{dvit} は常に正であり理想的なケースでは 0 となる.

判別ビタビ GTN における勾配の逆伝搬は, 図 20 の右側はビタビ学習 GTN と同様で, G_{int} の枝状の勾配では図 20 の左側から負の寄与を受ける. G_{int} の枝で G_{vit} にも G_{cvit} にも現れない枝は勾配は 0 となる. また, G_{vit} と G_{cvit} 両方に現れる枝の勾配も 0 となる. 言い換えれば, 枝が正解のパスに含まれているなら勾配は 0 となる. G_{cvit} に存在して G_{vit} に存在しない枝の勾配は +1 となりこの枝は G_{vit} に含まれるようにより低い制約値を持つべきであるといえる. その反対の場合, 枝の勾配は -1 となりこの枝は望まれる解に含まれないためより高い制約値を持つべきといえる.

判別ビタビ学習ではビタビ学習のような目立った欠陥はないが, 残存する問題としてはクラス間にはっきりとしたマージンを築いていないことである. そのため間違ったパスの制約値が望ましいパスの制約値に近づいたときに押し上げることができれば望ましいといえる.

C. フォワードスコアリング, フォワード学習

ビタビパスの制約は認識という目的に適しているが, 状況の部分的な判断材料にしかなりえない. 同じ分割に対応する複数の最小制約値パスが同じラベル系列を生成する場合, 1 つのパスのみがその解釈を生成する場合よりも, そのラベル列が正しいという証拠になるため全体の制約はより小さくなるといえる. 確率論的な枠組みでは, 解釈の事後確率はその解釈を生み出すパスの事後確率の総和であるべきで, 制約においては解釈に対応する制約値は個々のパスの制約の負の指数和の負の対数であるべきである. このような確率論的な枠組みでは, 全体のペナルティは個々のパスのすべてのパスより小さくなる.

解釈が与えられた時に上記の計算を効率的に計算するために フォワードアルゴリズムが知られている. また, これにより計算される特定の解釈に対する制約値をフォワード制約値と呼ぶ. 各ラベル系列に対して 1 つの制約グラフが存在し, ある解釈が与えられた時対応する制約グラフ上でフォワードをあるゴリ李イズムを実行すると解釈に対応したフォワード制約値を得る. フォワードアルゴリズムはビタビアルゴリズムと非常によく似た形で進行するが, 累積制約値を組み合わせる際の演算が min 演算ではなく以下に示すいわゆる logadd 演算であることが違いである.

$$f_n = \mathrm{logadd}_{i \in U_n}(c_i + f_{s_i}) \tag{13}$$

ここで, $f_{\mathrm{start}} = 0$, U_n はノード n の常駐の枝の集合, c_i は枝 i の制約値であり,

$$\mathrm{logadd}(x_1, x_2, ..., x_n) = -\log(\sum_{i=1}^{n} e^{-x_i}) \tag{14}$$

である． 枝に加算される制約値: score = exp(-penalty) を考えると, ビタビアルゴリズムでは累積スコアが最大の経路を選択し, スコアはその経路に沿って乗算される. 一方, フォワードアルゴリズムにおけるスコアは開始ノードから終了ノードまでの各経路に関連する累積スコアの合計である. フォワード制約値は常に他のパスの累積の制約値よりも低い. ただし, 制約値が他の経路に比べてかなり低い「支配的な」経路が存在する場合, その制約値はフォワード制約値とほぼ等しくなる.

フォワード制約値を用いることで, ビタビアルゴリズムと比較して最も低い制約値を持つ方法だけでなく, 答えを生成するすべての異なる方法を考慮することができ分割に曖昧さを含む場合に特に有効である.

フォワード学習 GTN は先述したビタビ学習 GTN におけるビタビ変換器を, 解釈グラフを入力としてそのグラフのフォワード制約値を出力とするフォワードスコア計算機に置き換え, 最良の経路 1 つではなく正解を含むすべてのパスの制約値を低くするように変更した GTN といえる.

フォワードスコア計算機における逆伝搬はビタビ変換器とは異なる. グラフの各ノード n で計算されたフォワード制約値 f_n に対する微分はグラフ G_C を介した逆伝搬により計算される.

$$\frac{\partial E}{\partial f_n} = e^{-f_n} \sum_{i \in D_n} \frac{\partial E}{\partial f_{d_i}} e^{f_{d_i} - c_i} \tag{15}$$

ここで, D_n はノード n の下流の枝の集合である. 上記の導関数から枝の制約に関する導関数が得られる.

$$\frac{\partial E}{\partial c_i} = \frac{\partial E}{\partial f_{d_i}} e^{-c_i - f_{s_i} + f_{d_i}} \tag{16}$$

G_C のすべての枝が損失関数に影響を与え, 低い制約値を持つパスに属する枝ほど大きな影響力を持つ.

D. 判別型フォワード学習

フォワード制約値に含まれる情報は識別的フォワード基準と呼ぶべき別の識別的学習基準に利用することができ, この基準は正しい解釈に関連する経路を選択する事後確率を最大化することに相当する. 理想的には, 制限付きグラフのフォワード制約値が完全な解釈グラフのフォワード制約値と等価になっていることが望ましい. なぜならこの等価性は 正しい解釈を持つパスに関連する事後確率がほぼ 1 になる場合に達成されるためである. 対応する GTN アーキテクチャの概要を図 21 に示す.

差分を E_{dforw}, 制限付きグラフ, 完全な解釈グラフのフォワード制約値をそれぞれ C_{cforw}, C_{forw} とする.

$$E_{dforw} = C_{cforw} - Cforw \tag{17}$$

E_{dforw} は, 解釈グラフと制限付きグラフの包含関係から常に正である. 先述した理想的な場合には $E_{dforw} = 0$ となる.

判別可能なフォワード GTN で導関数を逆伝搬すると, ビタビの場合よりも勾配が均等に分散される. 導関数は図 21 の左半分から解釈グラフまで逆伝搬され, また負となり右半分へも逆伝搬され左半分の結果に加えられる. 正しいパスの 1 部となる枝は正の導関数を持ち, この導関数は不正確なパスがすべての正しいパスよりも低い制約値を持つ場合に非常に大きくなる. 同様に, 低い制約値を持つ不正確なパスの 1 部となる枝に関する導関数は大きな負の導関数を持つ. 一方で, 正しい解釈に関連するパスの制約値が他のすべてのパスよりはるかに小さい場合, 損失関数の値は 0 にほとんど近くなり勾配はほとんど逆伝搬されない. よって学習は分類誤りをもたらすデータに集中し, さらに誤りを引き起こす画像の断片に集中する. 一般的には学習機が離散的な代替解釈を選択ならなければならない状況で同じ考えを用いることができるエレガントな手法である.

4.1 E. 識別学習における備考

先述した議論において大域的な学習基準に確率的な解釈を与えたが, グラフの枝の制約値については確率的な解釈を与えなかった. これには理由があり, 例えば和が 1 にならなければいけない, あるいは入力領域上で積分して 1 にならなければいけないなどの制約が異なるクラスラベルに関連する場合に問題が発生するためである.

クラスラベルの和が 1 にならなければいけないケース (クラスの正規化) においては, 画像の 1 部が有効なクラスに対応しない場合に分割候補が間違っている可能性があるとして局所的にすべてのクラスに対応しないために重要な情報を排除する可能性がある. Baum-Welsh アルゴリズムと Expectation-Maximization 法の組み合わせでは個々の変数の確率的な解釈が重要である. しかし, これらの方法は識別的な学習基準に適しておらず, 勾配に基づく学習においても非効率になりうる.

また, 入力領域上で積分して 1 にならなければいけないケース (入力の生成モデルを使用) について, 生成モデルは各クラスについての独立した密度モデルを構築しそのモデルに基づいて独立した分類の決定をすることで間接的な境界を構築する. これは分類判定面を学習するという学習における最終目標とは直接関係していないため識別的なアプローチではない.

分類のための内部変数が確率論的な解釈を持っていなくても, システム全体はクラスの事後確率を生成していると見なすことができる. 例えば先述した図 21 においてあるラベル系列が「望ましい列」として与えられるとすると仮定すると, $-E_{dforw}$ の指数はラベル列の事後確率の推定と相互的に予測可能である. 誤分類の数の近似値を直接最小化するといったアプローチがあるが, 本研究では最適化の際の数値的な問題が小さくなる, 分類モデルに適切と思うパラメータを自由に選択することができるという利点で判別可能なフォワード損失関数を用いている.

5 複数物体認識 : 空間変位ニューラルネットワーク

文字列の画像に対してヒューリスティックな分割を用いるかわりに, 正規化された画像のすべての可能な位置で認識器を適用する方法がある. しかし, この方法は大きく 3 つの問題がある. 1 つ目は計算コストが非常に大きいこと. 2 つ目は認識器が認識すべき文字の中心にある時, 認識器は近傍に他の文字があったとしても必ず正確に文字を認識しなければならないということ. 3 つ目は認識器がずれやサイズの変動に対してけんろうであることである. これらの問題は入力領域上で CNN を複製すればエレガントに回避できる. 第 3 章で述べたように CNN は入力画像のずれや大きさの変化, ノイズに対して非常に頑健である. このことで 2 つ目, 3 つ目の問題を回避できる. また, CNN は Space Displacement Neural Network (SDNN) と呼ばれる複製した CNN を用いる手法で大きな入力領域上で計算量を大幅に削減することができる. 入力領域上の CNN のインスタンスと近い場所にある CNN を考えると, CNN の性質上同じ出力を持つため, 共有されていない「スライス」だけが再計算される. その結果特徴マップが水平方向に大きくなっていることを除いて元のネットワークと同じ構造を持つ. SDNN は信頼できる分割方法が存在しない筆記体の認識タスクにおいて非常に魅力的である. SDNN のアイデアは非常に古く先述したように認識器への要求性能が高いため最近までは注目されていなかった.

A. GTN による SDNN の出力の解釈

SDNN の出力は入力の対応の位置で特定のクラスラベルを持つ文字を見つける尤度や制約値, あるいはスコアを符号化したベクトルの系列である. このベクトル列から最適なラベル列を抽出するために後処理が必要となる. SDNN においては高頻度で個々の文字が複数の隣接する認識器のインスタンスに発見される, 文字の一部しか見ていない認識器インスタンスによって誤って検出されることが起こる. 出力列からこういった文字を防ぐには 図 24 のような 2 つの入力グラフを持つグラフ変換器を用いる. SDNN が生じるベクトル列は隣接ノード間で複数の枝を持つグラフに変換される. 各枝はクラスのラベルと SDNN が生成する制約値が含まれる. これを SDNN 出力グラフと呼ぶ. 2 番目の入力グラフは文字モデル変換器と呼ばれクラスラベルの文字列と認識された文字列の対応する出力文字列間の関係を符号化する. この変換器は重み付き記号系列を他の重み付き記号系列に変換する. 図 24 のグラフ変換器は SDNN 出力グラフのすべてのパスに対応する系列を文字モデル変換器とマッチングさせることで SDNN 出力グラフと文字モデル変換器を合成変換器に通して解釈グラフを生成する. 解釈グラフには対応する出力ラベル列のパスが含まれる.

B. SDNN を用いた実験

本節で示す実験では LeNet-5 が分割なしに複数の文字を認識するように複製されることを目標に学習を進めた. データセットは先述した修正データセットにおいて画像処理を施したものを用いた. 図 25, 26 は LeNet-5 SDNN が複数の文字認識に成功した例である, LeNet-5 SDNN は顕著な不変性と耐ノイズ性があることが示された. また, 文字が密接に絡み合っている場合でも文字を区別することができている. さらに図 26 の左上の例のように文字を形成する切断されたインクの断片から文字のグループ化に成功している. 図 26 の例では連続する 1 を幾何学的な外部情報なしで認識しており, 最後の 4 についても文字モデル変換器によって 1 と誤識別された

結果が取り除かれて正しく認識されている. SDNN はこのような頑健性だけでなく, その「簡単さ」も重要な利点である. 簡単であるため並列ハードウェアに実装することができる.

C. SDNN の大域的な学習

上記の実験では, 文字列画像は人工的に生成された画像であるため重要な文字の位置とラベルがあらかじめわかっていることになる. 実用上は文字列に対するラベルの正確な系列は入手可能であるが, 入力画像中の対応する各文字の正確な位置は不明となる. SDNN の大域的な学習では, 第 6 節で述べたようなアーキテクチャに配置された図 27 のようなグラフ変換器を介して勾配を逆伝搬することによって可能となる. これは SDNN の出力を隠れマルコフモデルでモデル化するのと等価である. 大域的に学習された可変サイズの TDNN/HMM ハイブリッドは様々な分野に用いられている. 図 27 は SDNN/HMM ハイブリッドを識別的フォワード基準で学習するためのグラフ変換器アーキテクチャである. 図 27 の右側では SDNN 出力系列と文字モデル変換器の合成によりすべての可能な解釈を示す解釈グラフを得る. 左側はさらに所望のラベル系列を持つパスのみを含む文法と合成する. 損失関数は左半分から得られたフォワードスコアと右半分から得られたフォワードスコアの差である. 合成変換器を逆伝搬するには, SDNN 出力グラフのどの枝が解釈グラフのどの枝を発生させたか記録する必要がある. SDNN 出力グラフ内の枝に関する導関数は, その枝を始点とする解釈グラフのすべての枝に関する導関数の和に等しい. 同様に文字モデル認識器の制約値についても導関数を計算できる. 先述したようにネットワークの出力 RBF が適応的である場合には破綻が起こりえるため識別的基準を用いなければならない. SDNN は非常に有望な技術であるが, ヒューリスティックな分割よりも良い結果を残せていない. これは今後の課題である.

D. SDNN による物体検出と注目

SDNN と大規模な入力領域と親和性の良さが相まって大規模画像における「総当たり的な」物体の発見と検出に利用できることが示唆されている. 1 つの CNN を学習させ背景の画像から目的の物体の画像の画像を区別するといったことがアイデアとして考えられる. ネットワークは入力画像を分析するために画像全体を覆うように複製され結果的に 2 次元空間変異ニューラルネットワークが形成される, SDNN の出力は 2 次元平面となり活性化されたユニットがその中にあれば, 受信領域に注目する物体が存在することを意味する. 画像内の対象物体の大きさは未知であるため画像を複数の解像度でネットワークに通し, 複数の解像度の結果を結合することで結果を示すことが可能である. このアイデアは顔の位置検出などに応用されている.

画像中の顔検出の場合を考えると, 顔を含む画像を収集し, ラプラシアンフィルタで照明の変動と低い空間周波数の照明勾配が除去される. 次に, 手動でサンプルを抽出し顔の部分をサイズ正規化する. 背景画像の大きさはランダムに選択され, これらのサンプルに対して CNN が学習し, 顔部分画像と非顔部分画像を識別する.

画像を解析する際, ラプラシアンフィルタを通してから 2 のべき乗の解像度でサブサンプリングしネットワークは複数の解像度の画像に対してそれぞれ複製される. 結果の結合は単純な投票処理が用いられる.

前節の大域的な学習手法の 2 次元版を用いることで学習サンプルを作成する際に顔の位置を手動で特定する必要性を削減できる.

他の研究では顔検出に NN や SVM のようなクラス分類器を用いて大きな成功を収めている. これらは複数のスケールでネットワークに画像を通すというアイデアを含めて上記手法と非常に似ている. しかし CNN を用いていないため CNN の高速化が生かせず, 高速化のために他の手法を併用している. それに加えて　これらのクラス分類器は CNN に比べて眼鏡性が低いため分類器に通す画像を増やさなければならない.

6　GTN と変換器

本節では GTN を 一般化変換の枠組みで再解釈し強力なグラフ合成アルゴリズムを提案する

A. 先行研究

音声認識においては, グラフベースの統計モデルと音声人市区モジュールを統合する勾配型学習が用いられている. しかし, 多層グラフを用いた学習可

能なシステムにおいてのシステマチックなアプローチは提案されていない. グラフを他のグラフに変換するアイデアは CS の分野で大きな注目を集めており, 手書き文字のための提案もなされている. この研究では変換器かとグラフを組み合わせることによる代数的な側面について述べているが, 変換器から大域的に学習可能なシステムを構築するという点にはほとんど触れられていない. 本研究ではグラフを操作するシステムの自動的な学習のためのアプロ^値を提案する.

6.1 B. 標準的な変換器

有限状態変換器の枠組みでは, グラフの枝に離散的な記号がつけられている. 変換器グラフは入力記号と出力記号の 2 つを持ち, アクセプタグラフは各枝に 1 つの記号を持つ. この枠組みでは合成操作はアクセプタグラフと変換器グラフを入力として新たなアクセプタグラフを構築する. 合成操作はアクセプタグラフと変換器グラフを構築すると, 出力アクセプタグラフの各パス S_{out} は, 入力アクセプタグラフの 1 つのパス S_{in} と変換器グラフの入出力系列のペアと 1 つのパスに対応する. 出力アクセプタグラフの枝状の重みは入力アクセプタグラフと変換器グラフのマッチングからの重みを加算して得られる. 以降, このグラフ合成操作を変換操作と呼ぶ. 変換の例が図 28 に示す. 変換器の枝状の出力記号と入力記号は常に同一であり, このタイプの変換器グラフは文法グラフと呼ぶ. トークンが入力アクセプタグラフと変換器グラフの開始ノードにそれぞれいると仮定すると, 両方のトークンがグラフの終端ノードに到達したとき許容可能な軌跡を持つ. この軌跡はアクセプタグラフと変換器グラフの両方に準拠した入力記号の並びを示している. 次に, 軌跡の沿って対応する出力記号列を集めることができる.

この変換操作は非常に効率的であるが, 枝にラベル付けされている null と非 null 記号のすべての組み合わせの処理が複雑である. 重みが的確に正規化され確率として解釈される場合, アクセプタグラフはグラフ内のすべての可能なパスに関連するラベル系列集合によって定義される言語上の確率分布を示す. 変換操作の応用として単語などの文字列を認識する際に言語的制約を取り入れることが挙げられる. この例では各分割候補に関してニューラルネットワーク認識を適用してアクセプタグラフを作成する. このアクセプタグラフは文法の変換器グラフと一緒に構成される. この文法変換器には有効な記号列のパスが含まれ, 枝には同一の入力記号と出力記号が含まれる.

C. 一般化した変換

各枝に関連するデータが有限個の値しかとらない場合, 入力グラフを合成し変換器を使用することは妥当であるが, 画像認識といった場合グラフの枝のデータ構造はベクトルや画像, その他の高次元のオブジェクトとなる. そのためこれを解決する新しい合成操作を紹介する.

このような複雑なグラフを構成するには, さらに情報を追加する必要がある.

- 各入力のグラフから 1 組の枝を調べる時, 入力グラフの枝に付加された情報に基づいて出力グラフに対応する枝とノードを作るかどうかの基準が必要である. これにより枝, 複数の枝, あるいは複数のノードと枝からなるサブグラフ全体を構築することを決めることができる.
- この基準を満たした場合, 出力グラフに対応する枝とノードを作成し, 新たに作成された枝に付与する情報を計算する必要がある.

これらの機能はコンポジション変換と呼ばれるオブジェクトにカプセル化されている. このインスタンスは 3 つのメソッドを実装している.

- check(arc1, arc2)

  arc1 と arc2 が持つデータ構造を比較して対応する arc を出力グラフに作成すべきするかどうかを返す.

- fprop(ngraph, upnode, downnode, arc1, arc2)

  check が True を返すと呼び出される. 出力グラフ ngraph のノード upnode と downnode の間に新しい枝とノードを作成し, これらの新しく作成した枝に付属する情報を計算する.

- bprop(ngraph, upnode, downnode, arc1, arc2)

  arc1 と arc2 のデータ構造, また同じ引数で fprop を呼び出した際に使用したパラメータに関して upnode と downnode の出力部分グラフから勾配を伝搬させるために学習中に呼び出される. これは fprop の計算に用いる関数が微分可能であることを前提としている.

check 関数で動的なアークテクチャを構築し, fprop 関数でそのアーキテクチャを通して枝に付加された数値情報を計算する. bprop 関数ではアーキテクチャを逆伝搬して枝に付与された情報に対する損失関数の偏導関数を計算する. 図 29 は一般化されたグラフ合成アルゴリズムを簡略化したものである. このアルゴリズムは Null 遷移は扱われず, 両方のトークンが同時にそのグラフの端点に到達する. null 遷移を管理するには各トークンの null 繊維の可能性を再帰的にシミュレーションし, 最終的に fprop 関数を呼び出せばよい. 許容可能な軌跡を特定する最も安全な方法は終端ノード上の両方のトークンに到達可能なトークンの構成を特定するの副次的なパスを実行することであり, これは逆方向の軌跡を列挙することで容易に達成できる. 変換器を用いたグラフ合成は, 一般化された変換として簡単かつ効率的に実装される. check 関数は 2 つの枝状の入力記号を比較し, fprop 関数は変換器の枝状の出力記号を記号とする枝を生成する. グラフのペア間の合成は, 手書き文字認識装置に言語的制約を組み込む際に特に有効である. 本論文の残りの部分では, 複数のグラフの変換に基づくグラフ変換器を示す. これまでに紹介されてきた分割器や認識器といったグラフ変換器の多くは一般化された変換の観点から定式化できる. この場合, 変換の入力は 1 つのグラフとなり, (check, fprop) のペアそのものが手続き的に変換器を定義しているとみなすことができる. 実際には生成されるグラフは手続き的に表現され, 認識時に探索アルゴリズムが訪れるノードだけをインスタンス化する. このことでビームサーチに代表される刈込アルゴリズムの利点がグラフ変換ネットワーク全体に伝搬される.

D. グラフ構造に関する注意点

bprop 関数は一般的なグラフ変換器における逆伝搬アルゴリズムの基礎となるものである. check 関数が関係を確立すべきと判断すると, fprop 関関数が数値の計算を実行し, ネットワークの構造が確立される. fprop は微分可能であると仮定するので勾配はネットワークのアーキテクチャに沿って逆伝搬することができ m ほとんどのパラメータはシステムのグラフの枝に格納されたスコアに影響を及ぼす. グラフに枝が現れるか否かを決定できる閾値パラメータもあり, ここではそのパラメータについてのみ考察する. これまで述べてきたようなシステムでは, グラフ変換器によって生成されるグラフの構造に関してはグラフ変換器の性質によって決まるが, パラメータの値や入力に依存することもあり得る.

E. GTN と隠れマルコフモデル

GTN は HMM の一般化及び拡張とみなすことができる. 一方で, 確率的解釈は維持するか, 最終決定段階まで進めるか, 完全に落とすかのいずれかである. 一方でグラフ変換器ネットワークは複数のモジュールをフレームワークにより組み合わせることで HMM を拡張する.

HMM を展開すると, 解釈グラフと非常によく似たグラフが得られる. これはモデル内の各時間ステップ t と状態 i に関連するノード $n(t,i)$ を持つ. $n(t-1,i)$ から $n(t,i)$ への枝の制約値 c_i は時間空間において位置 t の観測データ o_t が放出されて状態 j から i に至る負の対数確率に相当する. 確率論的に解釈するとフォワードペナルティは観測データ列の尤度の負の対数である.

第 6 節では非識別損失関数を用いてニューラルネットワークと HMM のハイブリッドシステムを学習する際, 崩壊現象が起こりうる可能性を示した. 古典的な HMM では確率変数の確率の値の和や積分が 1 となるような制約が強制されるのでこの現象は発生しない. 一方で, HMM の確率的仮定が現実的でないときは第 6 節で述べる識別学習により性能を向上させることができる.

入力-出力 HMM モデル (IOHMM) はグラフ変換器と強く関連している. IOHMM は入力列が与えられた時の出力列の条件つき分布を表現する. IOHMM は出力変数の条件付き放出確率を計算する放出確率モジュールと, 入力値が与えられた時状態変数の値が変化する条件つき遷移確率を計算する遷移確率モジュールから構成される. グラフ変換器としてみると入力グラフの各パスに出力グラフを割り当てる. これらの出力グラフはすべて同じ構造を持ち, 枝の制約値は単純に加算され完全な出力グラフを得る. 放出確率モジュールと遷移確率モジュールの入力値は IOHMM の入力枝状のデータ構造から読み取れる.

7 オンライン手書き文字認識システム

手書き文字は様々な書体が存在している. これを認識できればペン型デバイスとの接続が大幅に向上するが実現にはまだ課題がある. 文字だけを見れば非常に曖昧だが単語全体の文脈を考慮すれば十分な情報が得られる. 本研究では, 単語構造に幾何学モデルを当てはめることで単語や単語群を正規化する前処理器, 正規化されたペンの軌跡から注釈つき画像を生成するモジュール, 文字を発見し認識する複製畳み込みニューラルネットワーク, 単語レベルの制約を考慮してネットワークの出力を解釈する GTN の 4 つの主要モジュールに基づいてペン型デバイス用の単語認識システムを構築した.

本研究では, 第 7 節で述べた SDNN に基づくシステムと, 第 5 章で述べたヒューリスティックオーバーセグメンテーションに基づくシステムを比較している. ペンの軌跡の情報は連続的であるため, ヒューリスティックオーバーセグメンテーションは非循環的な文字に対して適切な文字の分割候補を提案する際に非常に効率的な手法である.

7.1 A. 前処理

入力の正規化により文字内のばらつきを抑えて認識を単純化できる. 単語構造の幾何学的モデルのフィッテングに基づく単語正則化スキームを利用した. ペンの軌跡から手書き文字を認識する方法は, 時間領域で行われることが多い. また, 軌跡は正規化されることで局所的な特徴を抽出される. カーブマッチングや TDNN などの分類手法を用いて認識することができる. しかし書体に依存するため高い精度で文字の書き手に依存しない認識が困難となる.

書き手のストロークの順序や各速度に依存せず, あらゆるタイプの手書き文字に使用できるように AMAP という方式を提案した. AMAP は各画素が 5 画素の特徴ベクトルを持つ注釈付き画像とみなせる. AMAP は他の多くの表現と異なり, 分割を必要としない.

ネットワークアーキテクチャ

文字認識において最も優れたネットワークの 1 つは LeNet-5 にやや似た 5 層畳み込みニューラルネットワークである. 出力の分散符号は LeNet-5 と同じであるが, LeNet-5 と異なり適応的である. ヒューリスティックオーバーセグメンテーションシステムで使用する場合, ネットワークの入力は 5 平面, 20 行, 18 列の AMAP で構成される.

SDNN の場合では入力単語の幅に応じて列の数が異なり, サブサンプリング層の数とカーネルのサイズが決まればすべての層のサイズが一意に決定される. 本研究では接続の総数を制限するため, サブサンプリング率をできるだけ小さく (2 × 2), 最初の層のカーネルも可能な限り小さくした. このようなアークテクチャでは性能がいいとは言えず, 学習にかなりの時間を有した. 入力領域を半分にした小さなアーキテクチャは入力の解像度が不十分であるため性能が低下した. しかし, 各ピクセルに単一のグレーレベルを与えるよりも角度や曲率がより多くの情報を与えるため入力の解像度は光学的な文字認識と比べて遥かに小さくてよい.

ネットワークの学習

学習は 2 段階に分けて行った. まず, RBF の中心を固定し正しいクラスに対応する RBF ユニットの出力距離を最小にするようにネットワークの重みを学習させた. 学習は分割された文字に対して実行された. 第 2 段階では単語レベルでの識別基準を最小化するためにすべてのパラメータ, ネットワークの重み, RBF 中心を大域的に学習させた.

ヒューリスティックオーバーセグメンテーションにより GTN は主に 4 つのグラフ変換器から構成される.

1. **分割変換器** はヒューリスティックオーバーセグメンテーションを実行し, 分割グラフを出力する. このグラフの枝に付与された画像に対して AMAP が計算される.
2. **文字認識変換器** は分割候補に対して CNN 認識器を適用し, 各枝に制約値とクラスを付与した認識グラフを出力する.
3. **合成変換器** は認識グラフと語彙制約を組み込んだ文字モデルを表すグラフを合成する.
4. **ビームサーチ変換器** は解釈グラフから良好な解釈を抽出する.

SDNN のアプローチでは, 以下のようにグラフが変換される.

1. **SDNN 変換器** は単語画像全体に対して CNN を複製し, 認識グラフを出力する.
2. **文字レベル合成変換器** は認識グラフを文字クラスごとに左から合成する.
3. **単語レベル合成変換器** は前の変換器の出力と語彙制約を組み込んだ言語モデルと組み合わせ解釈グラフを生成する.
4. **ビームサーチ変換器** は解釈グラフから良好な解釈を抽出する.

このアプリケーションにおいて解釈グラフでは明示的ではなく手続き的に表現される.

本研究では, ネットワーク内のすべてのグラフ変換モジュールが単一の基準に対して同時に学習したことに重要な意味がある.

7.2 D. 実験結果

1 つ目のの実験ではニューラルネットワーク分類器と単語正則化前処理および AMAP 入力表現を組み合わせてその汎化性能を評価した. データセットは書き手非依存の手書き文字約 10 万文字 (大文字, 小文字, 数字, 句読点の 95 クラス) である. 分割された文字の学習を実施し, テストには大文字 (9122 パターン, 誤差 2.99 %), 小文字 (8201 パターン, 誤差 4.15 %), 整数 (2938 パターン, 誤差 1.4%), 句読点 (881 パターン, 誤差 4.3 %) について別々に実施した. 実験は上記のネットワークアーキテクチャを用いて, 認識器の頑健性を高めるために元の文字に局所的なアフィン変換を施してデータ拡張した.

2, 3 番目の実験は小文字の単語認識に関する実験である. データセットは 881 語のデータベースである. 単語の正則化によってもたらされる改善点を評価した. SDNN/HMM システムではネットワークが 1 度に 1 つの単語全体を見るため単語レベルの正規化を施さなければならない. ヒューリスティックオーバーセグメンテーション では単語レベルで学習する前に文字レベルで正規化すると, 25461 語の辞書内において, 単語と文字の誤り (挿入, 削除, 置換) はそれぞれ 7.3% と 3.5% であった. 文字レベルの正規化の代わりに単語レベルの正規化を施した場合, それぞれ 4.6% と 2.0% とそれぞれ相対的に 37% と 43% 減少した. このことから最初に分割して各分割結果を正規化するよりも全体を正規化することで誤識別率が大幅に減少するといえる.

3 番目の実験ではニューラルネットワークと前処理器を組み合わせた学習において, 文字レベルの学習と比較した. 上記のように文字単位での初期学習誤, 3500 後の小文字単語データセットを用いて単語レベルの大域的な識別学習をした. SDNN/HMM システムでは辞書の制約がない場合, 単語, 誤差それぞれについて誤識別率は 38%, 12.4% から, 単語レベルの学習後には 26%, 8.2% となり, 32%, 34% の相対的な低下がみられた. ヒューリスティックオーバーセグメンテーションとそのアーキテクチャを若干改良し, 辞書制約を無くした場合, 単語と文字の誤識別率が 22.5%, 8.5% から 17%, 6.3% に減少し相対的に 24.4%, 25.6% 低下した. 25461 誤の辞書では, 単語の文字の誤識別率は単語レベル学習によりそれぞれ, 4.6%, 2.0% から 3.4%, 1.4% に低下し, 相対的に 30.4%, 30.0% 低下した. 辞書のサイズを小さくするとさらに誤差がさらに低くなった. これらの結果から大域的に学習された NN/SMM ハイブリッドが手書き文字認識に有用であることが明確に示された.

8 小切手読み取りシステム

本節では, 産業界への展開を意図した GTN ベースの小切手認識システムについて説明する. 小切手の金額確認は, 銀行にとって非常に時間とコストがかかる作業であるため自動化に関心が集まっている. 銀行が設定した自動小切手読み取り装置の経済的な実行可能性の閾値は, 小切手の 50% が 1% 未満のエラーで読み取られるときである. このようなケースでは, システムは 50% の正解率 / 49% の拒否率 / 1% のエラー率で構成されている. 今回提案するシステムはこの閾値を超えた最初の 1 つである.

小切手は Coutesy の金額は数字で書かれ, Legal の金額は文字で書かれる. 単純化のために最初のタスクは Coutesy 金額のみを読み取ることとする.

- システムはすべてのフィールドの中から Coutesy 量が最も高い候補を見つからなければならない. 多くの混同する数字の羅列がたくさんあるため, 多くの場合どの候補が Coutesy 金額であるかを判断することは非常に困難である
- システムは入力領域を文字に分割し, 候補の文字を呼んでスコアを付け最後に確率的な解釈から金額の最適な解釈を見つける必要がある

GTN の手法を用いて個人用, 商業用両方に対応する小切手読み取りシステムを構築した.

A. 小切手金額認識のための GTN

ネットワークが小切手の数値を読み取ることを可能にするグラフ変換を説明する. 各グラフ変換器ではパスがその段階で考慮された確率的な仮説を符号化しスコア付けしたグラフを生成した. システムの入力は小切手全体の画像を伝搬する 1 つの枝を持つ単純なグラフである.

領域位置変換器 T_{field} は古典的な画像解析をして小切手の金額を含む可能性のある長方形領域を抽出し, 領域グラフを生成する. 領域グラフは各候補領域が開始ノードと終了ノードを結ぶ 1 つの枝と関連付けられている. 各枝はその領域の画像とその特徴量から計算された制約値を持つ. 制約値はその領域が候補であることを示唆する場合には 0 に近くなり, そうでない場合には大きくなる. 制約値を計算する関数は微分可能であったためパラメータは大域的に調整可能である. 枝は連続した領域としてドルとセントの金額を別々に表せることができる.

分割変換器 T_{seg} は領域グラフに含まれる各領域を調べ, ヒューリスティックな画像分割を用いて各画像をインクの断片に切断する. 領域グラフの各枝はインクの断片のすべての可能なグループを表す分割グラフに置き換える. 各枝はセグメント画像とそのセグメントが実際に文字を含む可能性の初期評価となる制約値を含んでいる. このペナルティはいくつかの単純な特徴と調整可能なパラメータを組み合わせた微分可能な関数で得られる. 分割グラフは領域画像に対して可能なすべての分割を示す.

分割器は様々なヒューリスティックな知識を用いて分割候補を発見する. "hit and deflect" というアイデアが重要であり, 画像に線を投げて黒い画素にぶつかるか判定することでダブルゼロのような接触文字を分離することができる.

認識変換器 T_{tec} は分割グラフ内のすべての枝を反復的に処理して一致する分割画像上で文字認識器を適用する. 本研究ではこの認識器は LeNet-5 である. 認識器は画像を ASCII フルセットの 95 クラス, その他のクラスの計 96 このクラスに分類する. 入力グラフ T_{tec} の各枝は出力グラフの 96 この枝に置き換えられ, 制約値は入力の分割グラフん一致する枝の制約値の総和であり, さらに制約値を認識器によって計算された画像を対応することに関連付ける. 各パスは対応する領域の可能な文字列を示している.

構成変換器 T_{gram} は認識グラフと文法グラフの 2 つのグラフを入力とする. 文法グラフには金額を構成する紀伊豪のすべての系列が含まれている. 2 つの入力グラフを結合して出力を生成する操作は一般化された変換である. 出力グラフの枝に付加されるデータは微分可能な関数によって計算される. 出力グラフの枝の制約値は 2 つの入力グラフの枝の制約値を単純に加算したものである. 得られた解釈グラフの各パスにそった制約値の合計はパスに対応する解釈の悪さを表しており, 文法グラフだけでなく各モジュールからの結果も組み合わせている.

微旅変換器 は最終的に累積制約値が最も小さいパスを選択することで文法的に正しい解釈と一致する.

B. 勾配に基づく学習

この小切手認識システムの各段階において調整可能なパラメータが含まれている. これらの大部分は学習する必要がある. 各モジュールのパラメータは妥当な値で初期化されセグメンテーションや Lenet-5 も適切な初期化, 事前学習を施す必要がある. 次に, 正しい金額のラベルがつけられた小切手画像全体からシステム全体をグローバルに学習させる. 最小化される損失関数 E は第 6 節で説明した判別可能なフォワード基準である.

C. 低信頼小切手の拒否

ビタビ認識器で誤った結果が得られる可能性がある場合, それらを信頼度で評価して閾値よりも低い場合その小切手を拒否できるようにする必要がある. 2 つの異なる小切手の正規化されていないビタビ制約値を比較することはどちらの答えを最も信じるべきか決定する際に意味が無い. この信頼度の適切な尺度は入力画像に対するビタビ解答の確率である. ビタビ会頭のようなターゲット系列が与えられると識別的フォワードロスは目的の系列としてビタビ解答を使用する.

$$\text{confidence} = \exp(E_{dforw})$$

D. 結果

上記のシステムを実装して機械印刷された小切手画像を用いてテストした. ニューラルネットワークの分類器はまず, 様々な文字画像 50 万枚で学習された. 画像にはあらかじめ文字列レベルでサイズ正規

化された手書き文字と機会で印刷された文字の両方が含まれている．また, 単純なアフィン変換によりデータ拡張した．また, 小切手画像から自動的に分割され手作業で判定された文字画像でネットワークを学習させた．また, セグメンテーションで ASCII クラス以外の火文字を拒否するための初期学習も施した. つぎに小切手画像全体に対してパラメータの小さなサブセットを大域的に学習させた.

646 個の商業用小切手について, 82% が正しく認識, 17% が拒否, 1% がエラーとなった. 従来のシステムでは, 68% が正しく認識, 31% が拒否, 1% であり性能が向上したことが確認された．この原因として, 認識器が大規模になりより多くのネットワークで学習されるようになったこと, GTN アーキテクチャにより既存手法と比較してかなり効率的に文法的な制約の利点が得られたこと, GTN アーキテクチャがテストやパラメータの調整やチューニングにおいて非常に柔軟であることの 3 つが考えられる. 最後の点は重要である．GTN の枠組みはシステムのアルゴリズム部分と知識ベースの部分を分離して後者を簡単に調整ウすることができる. 今回の課題は大域的な学習により調整したパラメータはごく 1 部しかないため大域的な学習の重要性はごくわずかであった.

1995 年にシステムインテグレーターにより独立したテストが実行され, 他の小切手認識システムより優れたシステムであると示された. NCR の小切手読み取りシステムのラインナップに統合され, 1996 年 6 月から全米のいくつかの銀行に導入されて以来 1 日当たり数百万枚の小切手を読み取ってきた.

9 結論

CNN は手作業における特徴抽出の必要性を無くすことが示されている．GTN は文書認識システムにおいて手作業のヒューリスティック, ラベリング, パラメータチューニング といった処理の必要性を軽減する方法が示されている. 学習データがより豊富になり計算機の性能が上がれば認識システムはより学習部分に依存することになり性能が向上すると考えられる.

逆伝搬アルゴリズムが多層ニューラルネットワークにおけるユニット割り当て問題をエレガントに解決したように, 本論文で提案した学習法は入力ごとにアーキテクチャが動的ン変化することでユニット割り当て問題を解決する. 本論文の結果は大規模なシステムにおける学習のための原則として勾配に基づく最小化法の有用性を確立するのに役立つ.

文書解析システムのすべてのステップは勾配を逆伝搬できるグラフ変換として定式化できることが示された. グラフ変換の設計思想は, ヒューリスティックな部分と一般的な手続き的な知識部分を分離する.

HMM のようなデータ生成モデルはこの論文で説明したアーキテクチャと学習基準のほとんどを正当化するために要求されなかったことは指摘に値する.

具体的には本論文で紹介されている手法とアーキテクチャはパターン認識システムで直面する多くの問題に対する汎用的な問題に対する解決策を提供する.

1. 一般的に特徴抽出はそのタスクに関連する専門家の事前知識から導かれる．本研究では CNN に勾配に基づく学習を適用したことで, 例から適切な特徴を学習することに成功した.
2. 画像中の物体の分割と認識は切り離すことができない. 早めに分割する代わりに本研究ではヒューリスティックオーバーセグメンテーションを用いて多数の仮説を平行して生成し評価することで全体の基準が最小化されるまで分割の決定を先送りしている.
3. 文字認識器を学習させるためにマルチモジュールシステムを訓練して大域的な性能評価を最適化する. 手作業による事前作業が必要とならずコストがかからないだけでなく, モジュール間で協調して学習できるため著しく優れた認識性能が得られる.
4. 分割, 文字認識, 言語モデルといった情報源を結合するためにタスクに依存したヒューリスティックな仕組みを用いるのではなく, 入力に関する仮説を重みづけされた集合を表すグラフに対して一般化された変換手法を適用する統一的な枠組みを提案した.
5. 従来は多くの手作業によるヒューリスティックに頼っていた．CNN の頑健性を利用して明示的な分割を完全に回避することが可能となった. また, 勾配に基づく学習により分割と認識を同時に学習することが可能となった.

将来的には GTN を音声信号認識タスクや, 景色分析アプリケーションに適用することで, より学習への依存度を高めることで自動化を進めることができると期待している.

参考文献

[1] Richard O. Duda and Peter E. Hart. Pattern classification and scene analysis. In *A Wiley-Interscience publication*, 1973.

[2] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, Vol. 1, No. 4, pp. 541–551, 1989.

[3] H. S. Seung, H. Sompolinsky, and N. Tishby. Statistical mechanics of learning from examples. *Phys. Rev. A*, Vol. 45, pp. 6056–6091, Apr 1992.

[4] Vladimir Vapnik, Esther Levin, and Yann Lecun. Measuring the vc-dimension of a learning machine. *Neural Computation*, Vol. 6, pp. 851–876, 09 1994.

[5] Corinna Cortes, L. D. Jackel, Sara Solla, Vladimir Vapnik, and John Denker. Learning curves: Asymptotic values and rate of convergence. In J. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems*, Vol. 6. Morgan-Kaufmann, 1993.