のシステムでは、畳み込みネットワークを使用しないため、ここで説明した高速化を利用することができず、計算量を妥当な範囲に抑えるために、プレフィルタやリアルタイムトラッキングなどの他の技術に頼らざるを得ません。また、これらの分類器はConvolutional Networksに比べてスケール変動に対する不変性が低いため、分類器に提示する画像のスケール数を多くする必要があります。

VIII. グラフ・トランス・ネットワークとトランスデューサー

セクションIVでは、多層多モジュールネットワークの一般化として、状態情報を固定サイズのベクトルではなくグラフで表現するグラフトランスフォーマーネットワーク（GTN）を紹介した。本節では、GTNをGeneralized Transduction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.過去の作品

音声認識では、グラフベースの統計モデル（特にHMM）と音響認識モジュール（主にガウス混合モデルだが、ニューラルネットワークも含む）を統合する勾配型学習法が、数多くの著者によって用いられている[98], [78], [99], [67]。同様の考え方は手書き文字認識にも適用されている（再掲載は[38]）。しかし、多層グラフを用いた学習可能なシステムのシステム的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、重み付き有限状態トランスデューサ[86]の概念を通じて、コンピュータサイエンスで大きな関心を集めている。トランスデューサーは音声認識[100]や言語翻訳[101]に適用されており、手書き認識[102]のための提案もなされている。この研究は主に効率的な探索アルゴリズム[103]と、トランスデューサーとグラフ（ここではアクセプターと呼ぶ）を組み合わせる代数的側面に焦点を当てているが、トランスデューサーからグローバルに学習可能なシステムを構築することにはほとんど力を注いでいない。本論文では、グラフを操作するシステムの自動的な学習のための系統的なアプローチを提案する。グラフを用いた学習可能なシステムに対する別のアプローチとして、Input-Output HMMが[104], [105]で提案されている。

B.標準的なトランスダクション

有限状態トランスデューサ[86]の枠組みでは、グラフのアークに離散的な記号が付けられている。アクセプタグラフは各円弧に1つの記号を持つが、トランスジューサグラフは2つの記号（入力記号と出力記号）を持つ。特殊なヌル記号は他の記号に吸収される（記号列を作るために記号を連結するとき）。また、重み付き変換器とアクセプタは、各弧にスカラー量を持つ。この枠組みでは、合成操作はアクセプタ・グラフとトランスデューサ・グラフを入力とし、出力アクセプタ・グラフを構築する。この出力グラフの各パス（記号列S_{out}）は、入力アクセプタ・グラフの1つのパス（記号列S_{in}）に対応し、1つの

パスと、それに対応するトランスデューサーグラフの入出力シーケンスのペア (S_{out}, S_{in})。出力グラフのアーク上の重みは、入力アクセプタグラフとトランスデューサグラフのマッチングアークからの重みを加算することによって得られる。本論文の残りの部分では、トランスデューサーを用いたこのグラフ合成操作を（標準的な）トランスダクション操作と呼ぶことにする。

トランスダクションの簡単な例を図28に示す。この単純な例では、トランスデューサーアーク上の入力記号と出力記号は常に同一である。このようなトランスデューサー・グラフは文法グラフと呼ばれる。トランスダクション操作をよりよく理解するために、2つのトークンが入力アクセプタ・グラフとトランスデューサー・グラフの開始ノードにそれぞれ座っているのを想像してみよう。トークンはヌル入力記号でラベル付けされた円弧を自由にたどることができる。トークンは非ヌル入力記号でラベル付けされた円弧をたどることができるが、それはもう一方のトークンも同じ入力記号でラベル付けされた円弧をたどっている場合である。両方のトークンがグラフの終端ノードに到達したとき（すなわち、トークンが終端構成に到達したとき）、我々は許容可能な軌道を持つことになる。この軌道は、アクセプタとトランスジューサの両方に準拠した入力記号の並びを表している。次に、トランスデューサー・トークンの軌跡に沿って、対応する出力シンボル列を集めることができる。上記の手順は木を生成するが、第VIII-C章で説明する簡単な技法は、特定の出力状態がすでに見られたことを検出することにより、特定の部分グラフの複数のコピーを生成することを避けるために使用することができる。

この変換操作は非常に効率的に実行できるが[106]、ヌルと非ヌルの記号のすべての組み合わせの処理に関して、複雑な簿記の問題がある。重みが確率として解釈される場合（適切に正規化される）、アクセプタグラフは、グラフ内のすべての可能なパス（開始ノードから終了ノードまで）に関連するラベル配列の集合によって定義される言語上の確率分布を再提示することになる。

認識変換の応用例として、単語などの文字列を認識する際に、言語的制約（辞書や文法）を取り入れることが挙げられる。認識変換器では、各候補セグメントに対して、ニューラルネットワーク認識器を適用して、認識グラフ（アクセプターグラフ）を作成する。このアクセプタグラフは、文法のトランスデューサグラフと一緒に構成される。この文法変換器には、各有効な記号列のパスが含まれ、場合によっては、可能な記号列の相対的な尤度を示すペナルティで補強される。アークには同一の入力記号と出力記号が含まれる。ヒューリスティックな過分割学習GTNで使用されるパスセレクタは、コンポジットで実装可能である。トランスデューサーグラフは正しいラベル列を含む線形グラフである。解釈グラフとこの線形グラフの合成により、結合グラフが得られる。

C. 一般化されたトランスダクション

各弧に関連するデータ構造が有限個の値しかとらない場合、入力グラフを合成して

適切なトランスデューサーを使用することは、健全な解決策です。しかし、私たちのアプリケーションの場合、データ構造は
グラフの円弧は、ベクトル、画像、あるいは他の容易に列挙できない高次元のオブジェクト。
これを解決する新しい合成操作を紹介します。
の問題を解決します。

我々は、離散的な記号と円弧上のペナルティを持つグラフのみを扱うのではなく、ベクトルや画像のような連続値を持つデータ構造を含む複雑なデータ構造を持つ円弧を持つグラフを考慮することに興味がある。このようなグラフを構成するためには、さらに
情報を提供します。
- 各入力グラフから1組のアークを調べるとき、出力グラフに対応するアークとノードを作るかどうかの判断基準が必要である。
は、入力アークに付加された情報に基づく。アーク、複数のアーク、または複数のノードとアーク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円弧とノードを作成し、新たに作成された円弧に付随する情報を入力円弧に付随する情報の関数として計算する必要があります。

これらの機能は、Composition
Transformerと呼ばれるオブジェクトにカプセル化されています。Composition
Transformerのインスタンスは3つのメソッドを実装しています。
- check(arc1, arc2)
は、arc1（第1のグラフ）とarc2（第2のグラフ）が指すデータ構造を比較し、対応するarc（複数可）を出力グラフに作成すべきかどうかを示すブーリアンを再変換します。
- fprop(ngraph, upnode, downnode, arc1, arc2)
はcheck(arc1,
arc2)がtrueを返すと呼び出される。このメソッドは、出力グラフngraphのノードupnodeとdownnodeの間に新しいアークとノードを作成し、これらの新しく作成されたアークに付属する情報を、入力アークarc1とarc2の付属情報の関数として計算します。
- bprop(ngraph, upnode, downnode, arc1, arc2)
は、upnode と downnode 間の出力部分グラフから、arc1 と arc2 のデータ構造に、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、勾配形成を伝播させるために学習中に呼び出される。これは、fpropがその出力アークに付けられた値を計算するために使用する関数が微分可能であることを仮定している。

check
法は、機能依存の動的なアーキテクチャを構築すると見なすことができ、fprop
法は、そのアーキテクチャを通して前方伝搬を行い、アークに付加された数値情報を計算する。bprop
法は、同じアーキテクチャを逆伝播して、円弧に付された情報に対する損失関数の偏導関数を計算する。これは図28に示されている。

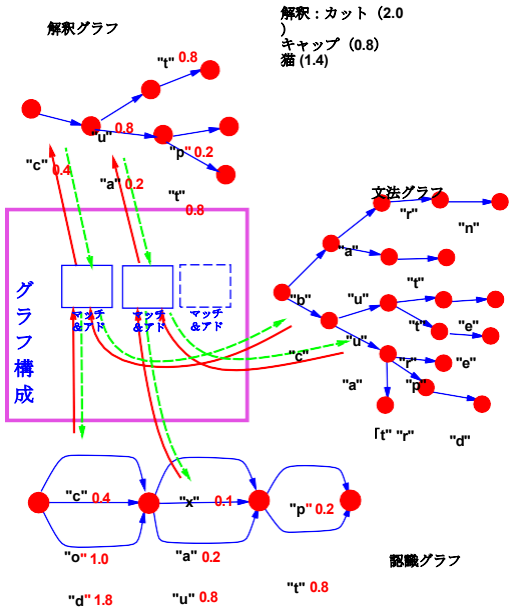


図29は一般化されたグラフ合成アルゴリズムの簡略化したものである。この簡略化されたアルゴリズムでは、Null遷移は扱われず、トークンのtra.

図28. を用いた認識グラフの構成例
という2つの文法グラフがある。前方伝搬（暗い矢印）では、check と fprop というメソッドが使われる。勾配（破線の矢印）は、bprop という手法を適用して逆伝播される。

jectoryが許容できる（すなわち、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遷移の管理は、トークンシミュレーションファンクションの簡単な修正である。NULLでない結合トークン遷移の可能性を列挙する前に、各トークンのNULL遷移の可能性をループし、再帰的にトークンシミュレーション関数を呼び、最後にメソッドfpropを呼び出す。許容軌道を特定する最も安全な方法は、終端構成（すなわち、終端ノード上の両方のトークン）に到達可能なトークン構成を特定するための予備的なパスを実行することである。これは、逆方向の軌道を列挙することで容易に達成できる。終端ノードから出発し、上流の円弧をたどる。メインパスでは、トークンが終端ノードに到達できるようなノードのみを構築する。

トランスデューサーを使ったグラフ合成（つまり標準的なトランスダクション）は、一般化されたトランスダクションとして簡単かつ効率的に実装されます。check
メソッドは単に2つの円弧上の入力記号の等質性をテストし、fprop
メソッドはトランスデューサー
の円弧上の出力記号をシンボルとする1つの円弧を生成する。

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。その使用例は、第IX章で述べるオンライン手書き文字認識システムと第X章
で述べる小切手読み取りシステムに示されている。
）

本論文の残りの部分では、Composition Transformer という用語は、複数のグラフの一般化された伝達に基づくグラフ変換器を示すことにする。一般化された変換という概念は、非常に一般的なものである。実際、セグメンテーションやレコグナイズなど、本論文で前述したグラフ変換器
の多くは、このような一般化された変換を行うことが可能である。

プレゼンテーションを簡略化するため、ヌル遷移の処理とデッドエンド回避の実装は行っていない。ここでは、合成の2つの主要な構成要素が明確に示されている。(a) トークンの軌跡を列挙する再帰的関数simtoken()、および (b) 合成グラフのどのノードが訪問されたかを記憶するための連想配列マップ。

```
関数 generalized_composition(PGRAPH graph1,
                             PGRAPH
                             グラフ2,
                             PTRANS
                             トランス)
PGRAPH を返します。
{
  // 新しいグラフを作成する
  PGRAPH ngraph = new_graph()

  // トークン位置間のマップを作成する
  // そして新しいグラフのノード
  PNODE map[PNODE,PNODE] =  new_empty_map()
  map[endnode(graph1),endnode(graph2)] =(endnode,
  endnode)とする。
     エンドノード(newgraph)

  //
   トークンをシミュレートする再帰的サブルーチ
  ン Function simtokens(PNODE node1, PNODE node2)
  Returns PNODE
  {
     PNODE currentnode = map[node1, node2] である。
     //
     訪問済みかどうかのチ
     ェック If (currentnode ==
     nil)
        // 新しい設定を記録する currentnode
        = ngraph->create_node() map[node1,
        node2] = currentnode
        //
        NULL でない可能性のあるものを列挙する
        // ジョイントトークンの遷移
        For ARC arc1 in down_arcs(node1) For
           ARC arc2 in down_arcs(node2)
              If (trans->check(arc1, arc2))PNODE
                 newnode =
                  simtokens(down_node(arc1),
                                down_node(arc2))
                  trans->fprop(ngraph, currentnode,
                                newnode, arc1, arc2).
     //
     構成されたグラフのノードを返す
     Return currentnode
  }

  // トークンのシミュレーションを行う
  simtokens(startnode(graph1),
  startnode(graph2)) マップの削除
  ngraphを返す
}
```

図 29.簡略化された一般化合成アルゴリズムの擬似コード。

は、一般化されたトランスダクションの観点から定式化されている。この場合、一般化された変換は、2つの入力グラフではなく、1つの入力グラフを取る。変換器のメソッドfpropは、初期グラフの各円弧に対して、いくつかの円弧、あるいは完全なサブグラフを作成することができる。実際、(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見なすことができる。

また、一つのグラフの一般化されたトランスデューシングは、このグラフと特定のトランスデューサーグラフとの標準的な合成と理論的に等価であることが示される。しかし、この方法で演算を実装すると、トランスデューサーが非常に複雑になるため、非常に効率が悪くなる可能性がある。

実際には、出力グラフ全体の構築（解釈グラフと文法グラフの合成などで膨大になる）を避けるため、汎化変換によって生成されるグラフは手続き的に表現される。認識時に探索アルゴリズムが訪れるノードのみをインスタンス化する（例：Viterbi）。この戦略により、刈り込みアルゴリズム（例えばビームサーチ）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

セクション VIでは、単純なグラフ変換器を介した勾配の逆伝播による大域的な学習のアイデアについて述べてきた。bprop法は一般的なグラフ変換器のための逆伝播アルゴリズムの基礎となるものである。一般化された構成変換器は、入力と出力の円弧上の核物理量間の関数関係を動的に確立するものと見なすことができる。チェック関数が関係を確立すべきと判断すると、fprop関数が数値の再関係を実現する。check関数は、合成変換器内部のephemeralネットワークの構造を確立する。

fpropは微分可能であると仮定しているので、勾配はその構造を通して逆伝播することができる。ほとんどのパラメータは、システムの連続したグラフの円弧に格納されたスコアに影響を与える。いくつかの閾値パラメータは、グラフに円弧が現れるか否かを決定することができる。存在しない円弧は非常に大きなペナルティを持つ円弧と等価であるため、ここではペナルティに影響を与えるパラメータの場合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（および第X章で述べる応用）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するノウハウの多くは、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。また、出力グラフの構造を学習しようとするGraph Transformerのモジュールを考えることも興味深い。これは組合せ問題であり、勾配学習には適さないと考えられるが、この問題の解決策として、グラフ候補を部分グラフとして含む大きなグラフを生成し、その中から適切な部分グラフを選択することが考えられる。

E.GTNと隠れマルコフモデル

GTNはHMMの一般化および拡張と見なすことができる。一方、確率的な解釈は、（ペナルティを対数確率とし）維持するか、（制約付き前進ペナルティと制約なし前進ペナルティの差をラベル列の負の対数確率と解釈し）最終決定段階まで進めるか、（ネットワークは単に入力空間におけるラベル列の決定表面を表す）完全に落とすかのいずれかである。一方、グラフ変換器ネットワークは、複数の処理レベル、または複数のモジュールをよく考えられたフレームワークで組み合わせることを可能にすることによってHMMを拡張する（例えば、Pereiraらは自動音声認識において異なる処理レベルを表すHMMを重ねるために変換器のフレームワークを使用してきた[86]）。

HMMを時間的に展開すると、解釈グラフと非常によく似たグラフが得られる（グラフ変換ネットワークのプロセスの最終段階、ビタビ認識の前）。これは、モデル中の各時間ステップtと状態iに関連するノードn(t, i)を持つ。n(t − 1, j) から n(t, i) へのアークのペナルティ c_i は、時間間隔 (t − 1, t) において、位置 t の観測データ o_t が放出され、状態 j から状態 i に至る否定的対数確率に相当する。このように確率論的に解釈すると、forwardペナルティは（モデルが与えられた）全観測データ列の尤度の負の対数である。

セクション VI では、非識別的損失関数を用いてニューラルネットとHMMのハイブリッドシステムを学習する場合、コラプシング現象が発生する可能性があることを述べた。前処理を固定した古典的な HMM では、確率変数の確率の値の和または積分が 1 になるような確率的制約を放出確率モデル、遷移確率モデルのパラメータが強制されるので、この現象は発生しない。つまり、ある事象の確率を上げると、他の事象の確率は必然的に下がります。一方、HMM（または他の確率モデル）の確率的仮定が現実的でない場合は、セクション VI 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) [105], [109]はグラフ変換器と強く関連している。IOHMMは確率モデルとして、（同じか異なる長さの）入力列が与えられたときの出力列の条件付き分布を表現する。IOHMMは放出確率モジュールと遷移確率モジュールから構成される。放出確率モジュールは出力変数の条件付き放出確率を計算する（入力値と離散的な「状態：変数」の値が与えられた場合）。遷移確率モジュールは、入力値が与えられたとき、「状態：変数」の値が変化する条件付き遷移確率を計算する。グラフ変換器として見ると、入力グラフの各パスに出力グラフ（出力変数の配列に対する確率分布を表す）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、そのペナルティは

のアークを単純に加算して、完全な出力グラフを得る。エミッションモジュールとトランジションモジュールの入力値は、IOHMMグラフトランスミッタの入力アーク上のデータ構造から読み取られる。実際には、出力グラフは非常に大きくなる可能性があり、完全にインスタンス化する必要はありません（つまり、低ペナルティ経路のみを作成する刈り込みが行われます）。

IX. オンライン手書き文字認識システム

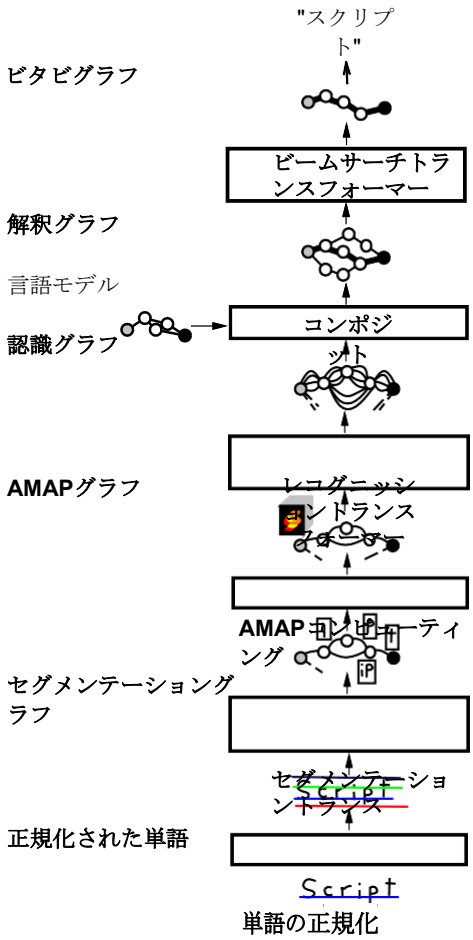
自然な手書き文字には，小文字の印刷物，大文字の文字，草書体など，さまざまなスタイルが混在していることが多い．このような手書き文字を確実に認識できれば、ペン型デバイスとのインタラクションが大幅に向上しますが、その実現には新たな技術的課題があります。また、文字だけを見ると非常に曖昧であるが、単語全体の文脈を考慮すれば、十分な情報が得られる。我々は、単語構造に幾何学的モデルを当てはめることで単語や単語群を正規化するプリプロセッサ、正規化されたペンの軌跡から「注釈付き画像」を生成するモジュール、文字を発見し認識する複製畳み込みニューラルネットワーク、単語レベルの制約を考慮してネットワークの出力を解釈するGTNという4つの主要モジュールに基づいて、ペンベースのデバイス用の単語認識システムを構築している。ニューラルネットワークとGTNは、単語レベルで定義された誤差を最小化するように共同で学習される。

この研究では、SDNNに基づくシステム（セクションVIIで説明）と、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（セクションVで説明）を比較している。ペンの軌跡の情報は連続的であるため（画像からの純粋な光学的入力よりも多くの情報を明らかにする）、ヒューリスティック・オーバーセグメンテーションは、特に非循環的な文字に対して、適切な文字のカットを提案するのに非常に効率的である。

A. 前処理

入力の正規化により、文字内のばらつきを抑え、文字認識を単純化することができる。我々は、単語構造の幾何学的モデルのフィッティングに基づく単語正規化スキーム[92]を使用した。このモデルは4本の「柔軟な」線からなり、それぞれアセンダー線、コア線、ベース線、ディセンダー線を表している。これらの線はペンの軌跡の局所的な極小値または極大値に適合させる。線のパラメータはEMアルゴリズムの改良版で推定され、観測点とパラメータ値の結合確率を最大化し、線が互いに潰れないようなパラメータに関する事前分布を用いる。

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 [110], [44], [111]．一般に、軌跡はノルマライズされ、局所的な幾何学的または動的な特徴が抽出される。そして、カーブマッチング [110]や、TDNN [44], [111]などの分類技法を用いて認識することができる。これらの表現にはいくつかの利点があるが、筆順や個人の書風に依存するため、認識が困難である。



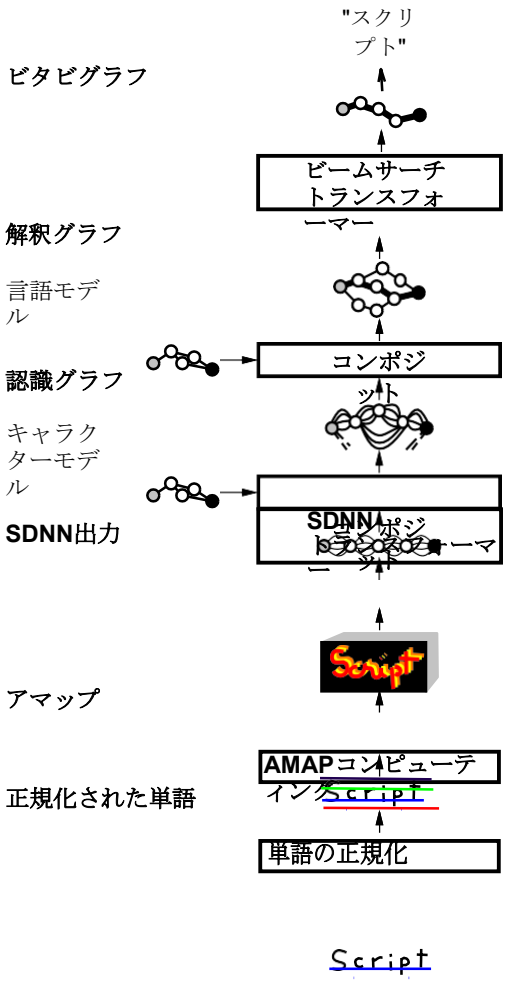


図30. ヒューリスティックなオーバーセグメンテーションに基づくオンライン手書き文字認識GTN

セグメンテーションと認識を統合した高精度なライター独立型システムで使用することができます。

書き手の意図は読みやすい画像を作成することなので、信号の絵画的性質をできるだけ維持しながら、同時に軌跡の連続的な情報を利用することが自然であると思われる。この目的のために、我々はAMAP[38]と呼ばれる表現方式を考案した。この方式では、ペンの軌跡は、各画像要素が軌跡の局所特性に関する情報を含む低解像度画像によって表現される。AMAPは、各画素が5要素の特徴ベクトルを持つ「注釈付き画像」と見なすことができる。4つの特徴は、その画素の周囲にあるペンの軌跡の4つの方向と関連付けられ、5つ目の特徴は、その画素の周囲にある局所的な曲率に関連付けられる。AMAP表現の特に有用な特徴は、入力軌跡の性質についてほとんど仮定をしないことである。これはストロークの順序や書く速度に依存せず、あらゆるタイプの手書き文字（大文字、小文字、草書、句読点、記号）に使用することができる。他の多くの表現（全体特徴など）と異なり、AMAPはセグメンテーションを必要とせず、完全な単語に対して計算することができる。

図31.空間変位ニューラルネットワークに基づくオンライン手書き文字認識GTN

B.ネットワーク・アーキテクチャ

オンラインとオフラインの両方の文字認識で最も優れたネットワークの1つは，LeNet-5にやや似た5層畳み込みネットワークである（図2）。
サイズ 5x5 の 25 カーネル、サイズ 4x4 の 84 カーネルによる層 4 の畳み込み、層 5: 2x1 サブサンプリング、分類層。
95個のRBFユニット（印刷可能なASCIIフルセットで1クラスにつき1個）。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な過分割システムで使用する場合、上記のネットワークへの入力は、5面、20行、18列のAMAPで構成される。この解像度は手書き文字を表現するのに十分であると判断された。SDNNバージョンでは、入力された単語の幅に応じて列の数を変化させた。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、入力を含むすべての層のサイズは一義的に決定される。あとは、各層の特徴マップの数と、どの特徴マップがどの特徴マップに接続されているかという情報だけが、アーキテクチャ上のパラメータとして選択される。我々の場合は、サブサンプリング率をできるだけ小さく（2x2）し、カーネルをできるだけ小さく（pos-1）した。

第1層(3x3)では、コネクションの総数を制限するために可能である。上位層のカーネルサイズは、上記のサイズ制約を満たしつつ、可能な限り小さくなるように選択されている。大きなアーキテクチャは必ずしも性能が良いとは言えず、学習にかなりの時間を要した。また、入力フィールドを半分にした非常に小さなアーキテクチャでは、入力の分解能が不十分であるため、性能が低下した。しかし、入力の解像度は光学的文字認識の場合よりもはるかに低い。なぜなら、各ピクセルに単一のグレーレベルを与えるよりも、アングルと曲率がより多くの情報を与えるからである。

C.ネットワーク・トレーニング

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。これは、前の層と正しいクラスのRBFの中心との間の平均二乗誤差を最小にすることと等価である。このブートストラップ段階は、孤立した文字に対して行われた。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、ネットワーク重み、RBF中心をグローバルに学習させた。

ヒューリスティック・オーバーセグメンテーションアプローチにより、GTNは主に4つのグラフ変換器から構成されていた。

1.セグメンテーショントランスファは、ヒューリスティックなオーバーセグメンテーションを行い、セグメンテーショングラフを出力する。そして、このグラフの円弧に接続された各画像に対して、AMAPが計算される。

2.文字認識トランスフォーマーは、各候補セグメントに畳み込みネットワーク文字認識器を適用し、各円弧にペナルティとクラスを持つ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言語モデルを表す文法グラフを合成する。

4.ビームサーチ変換器は、解釈グラフから良好なインタープリテーションを抽出します。このタスクは、通常のビタビ変換器でも達成可能であった。しかし、Beam Searchアルゴリズムは、大きな解釈グラフに適した刈り込み戦略を実装しています。

SDNNのアプローチでは、主に以下のようなグラフ変換が行われる。

1.SDNN Transformerは、単語画像全体に対して畳み込みネットワークを複製し、入力画像上の一定間隔を中心とした窓ごとにクラスペナルティを持つ線形グラフである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右へHMMで合成する（図27のように）。

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ランガージュモデルを組み合わせ、解釈グラフを出力する。

4.ビームサーチトランスフォーマーは、解釈グラフから良い解釈を抽出する。

このアプリケーションでは、言語モデルは最終的な出力グラフが与えられた辞書の文字ラベルのシーケンスを表現するように制約するだけである。さらに、解釈グラフは完全にインスタンス化されるわけではなく、ビームサーチモジュールに必要なノードのみが作成されます。したがって、解釈グラフは明示的ではなく、手続き的に表現されます。

本研究の重要な貢献は、第 VI 章および第 VIII 章で説明するように、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関して共同学習を行ったことである。最終的な出力グラフに対して識別的前進損失関数を用いた：制約された解釈の前進ペナルティを最小化し（すなわち、すべての「正しい：パス」に沿って）、同時に解釈グラフ全体の前進ペナルティを最大化する（すなわち、すべてのパスに沿って）。

グローバルトレーニングでは，付録 C に記載した確率的対角ルベンベルグ・マルカールト法を用いて損失関数を最適化した．この最適化は、システムのすべてのパラメータ、特にネットワークの重みとRBFの中心に対して行われる。

D.実験結果

最初の実験では，ニューラルネットワーク分類器と単語正規化前処理およびAMAP入力表現を組み合わせて，その汎化能力を評価しました．全ての結果は書き手非依存モードである（トレーニングとテストは異なる書き手）。手書き文字約10万字（大文字、小文字、数字、句読点の95クラス）のデータベースに対して、孤立文字の初期学習を実施した。テストは、大文字(9122パターン、誤差2.99%)、小文字(8201パターン、誤差4.15%)、数字(1.5%)の4種類の文字について個別に実施した。

大文字（9122パターン、2.99%）、小文字（8201パターン、4.15%）、数字（2938パターン、1.4%）、洒落文字（881パターン、4.3%）の4種類の文字について、別々に実験を行った。実験は、上記のネットワークアーキテクチャで行った。また、位置、大きさ、向き、その他の歪みに対する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学習データを追加作成した。

2番目と3番目の実験セットは、小文字の単語の認識に関するものである（ライター独立）。実験は881語のデータベースに対して行われた。まず、単語の正規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、ネットワークが一度に1つの単語全体を見るため、単語レベルの正規化を使用しなければならない。Heuris-tic Over-Segmentationでは、単語レベルの学習を行う前に、文字レベルの正規化を行った場合、25461語の辞書に制約された検索において、単語と文字の誤り（挿入、削除、置換の追加）は7.3%と3.5%であった。文字レベルの正規化ではなく、単語の正規化を行った場合、エラー率は単語と文字のエラーでそれぞれ4.6%と2.0%に低下し、すなわち単語と文字のエラーでそれぞれ37%と43%の相対的減少が見られた。

このことから、単語と文字の正規化により、誤認識率が大幅に減少することが示唆された。

は、最初にセグメント化し、各セグメントをノルマライズして処理するよりも、全体として処理することが望ましい。

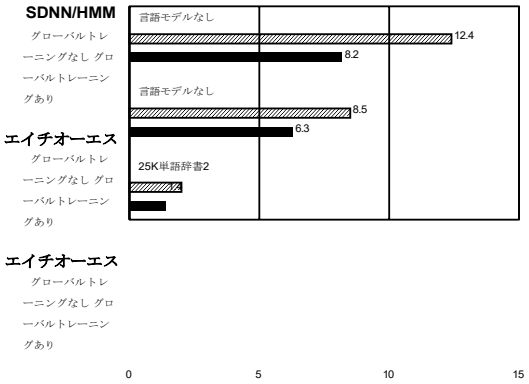


図32.SDNN/HMMハイブリッドとヒューリスティック過分割システム（HOS）のグローバル学習による改善を、25461語の辞書なしと辞書ありで比較した結果（文字誤り率）です。

第3の実験では，ニューラルネットワークとポストプロセッサの合同学習において，文字レベルの誤差のみによる学習と比較して，単語レベルの基準による学習でどの程度の効果が得られるかを測定した．上記のように文字単位での初期学習を行った後、3500語の小文字単語データベースを用いて、単語レベルのグローバルな識別学習を行った。

3500語の小文字単語データベースを用いてグローバルな単語レベルの識別学習を行った。SDNN/HMMでは、辞書制約がない場合、単語エラー38%、文字エラー12.4%から、単語レベル学習後に26%、8.2%となり、32%、34%の相対的な低下が見られた。ヒューリスティック・オーバーセグメンテーションシステムとそのアーキテクチャを若干改良し、辞書制約をなくした場合、単語エラー22.5%、文字エラー8.5%から17%、6.3%に低下し、相対的に24.4%、25.6%の低下となった。25461語の辞書では、単語レベル学習により、単語エラー4.6%、文字エラー2.0%から3.2%、1.4%に低下し、相対的に30.4%、30.0%の誤差が生じた。さらに、辞書のサイズを350語まで小さくすると、単語誤差1.6%、文字誤差0.94%となり、さらに低い誤差を得ることができる。

これらの結果は、グローバルに学習されたNeural-Net/HMMハイブリッドが手書き文字認識に有用であることを明確に示している。これは、以前に音声認識で得られた同様の結果を確認するものである[77]。

X.チェック・リーディング・システム

このセクションでは、産業界への即時展開を意図した GTN ベースのチェックリーディングシステムについて説明します。また、勾配学習とGTNを使用することで、正確で信頼性の高いソリューションを実現しながら、この展開をいかに迅速かつコスト効率の高いものにするかを説明します。

小切手の金額確認は、銀行にとって非常に時間とコストのかかる作業です。そのため、このプロセスを可能な限り自動化することに高い関心が寄せられています（例えば、[112]、[113]、[114]を参照）。部分的な自動化でさえ、相当なコスト削減につながるだろう。銀行が設定した自動小切手読取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。残りの50％の小切手は拒否され、人間のオペレーターに送られる。そのような場合の性能について説明する。

このシステムは、50%の正解率/49%の拒否率/1%のエラー率で構成されています。今回発表したシステムは、ビジネスと個人のチェックが混在する代表的なもので、この閾値を超えた最初の1つである。

小切手には、少なくとも2種類の金額が記載されています。Courtesyの金額は数字で書かれ、Legalの金額は文字で書かれます。一般に機械で印刷されるビジネス・チェックでは、これらの金額は比較的読みやすいが、ビジネス・チェックのレイアウトに標準がないため、見つけるのはかなり困難である。一方、個人小切手では、これらの金額は見つけやすいが、読むのはかなり難しい。

単純化するため(および速度要件)、最初のタスクはCourtesy量のみを読み取ることである。このタスクは主に2つのステップで構成されています。

- システムは、すべてのフィールド（テキスト行）の中から、表意金額を含む可能性が最も高い候補を見つけなければならない。これは、金額の位置が標準化されている多くの個人用小切手では当然のことである。しかし、すでに述べたように、ビジネス小切手では、金額を見つけることは、人間の目でもかなり難しい。小切手番号や日付、あるいは "not to exceed "の金額など、実際の金額と混同するような数字の羅列がたくさんあるのです。多くの場合、完全な認識を行う前に、どの候補が表意金額であるかを判断することは非常に困難です。
- ある金額の候補を読む（選ぶ）ために、システムはフィールドを文字に分割し、候補の文字を読んで点数をつけ、最後に小切手金額の確率的文法で表される文脈の知識を使って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、個人用小切手と事業用小切手の両方に対応する小切手金額読み取りシステムを構築しました。

A.小切手金額認識のためのGTN

次に、このネットワークがチェック量を読み取ることを可能にする連続的なグラフ変換を説明する(cf. Fig-33)。各グラフ変換器は、そのパスがシステムのこの段階で考慮された現在の仮説を符号化し、得点化したグラフを生成する。

システムへの入力は、小切手全体の画像を運ぶ単一の円弧を持つ些細なグラフである(図33を参照)。フィールド位置変換器 T_{field} は、まず古典的な画像解析（連結成分解析、インク濃度ヒストグラム、レイアウト解析などを含む）を行い、小切手金額を含む可能性のある矩形ゾーンを発見的に抽出する。T_{field} は、フィールドグラフと呼ばれる出力グラフを生成する(図33参照)。このグラフでは、各チェックゾーンが、開始ノードと終了ノードを結ぶ1つの弧と関連付けられている。各円弧は、ゾーンの画像と、ゾーンから抽出された単純な特徴（絶対位置、サイズ、アスペクト比など）から計算されたペナルティ項を含む。ペナルティ項は、特徴量がそのフィールドが候補であることを示唆する場合には0に近くなり、そのフィールドが量である可能性が低いと判断される場合には大きくなる。ペナルティ項

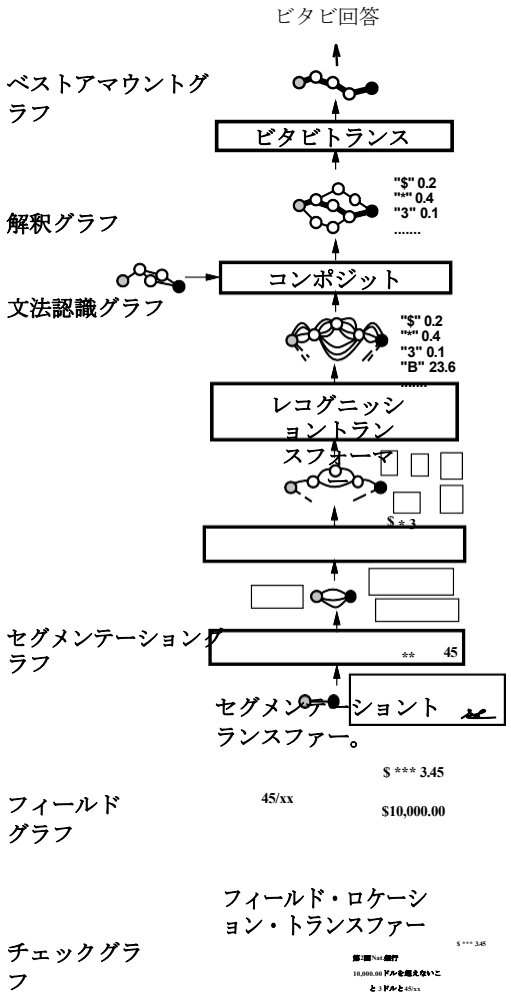


図33. グラフ変換モジュールの単一のカスケードとして実装された完全なチェック量リーダ。連続的なグラフ変換により、より高いレベルの情報が徐々に抽出される。

は微分可能であるため、そのパラメータはグローバルに調整可能である。

円弧は、一連のフィールドとして、ドルとセントの金額を別々に表すことができる。実際、手書き小切手では、セント金額は端数バーの上に書かれ、ドル金額と全く並んでいないことがある。最悪の場合、同じドル金額に対して、セント金額の候補が複数（端数バーの上下に）あることがある。

セクションVIIIで説明したものと同様のセグメンテーション変換器T_{seg}、フィールドグラフに含まれる各ゾーンを調べ、ヒューリスティック画像処理技法を用いて各画像をインクの断片に切り分ける。各インク片は、文字全体であっても、文字の一部であってもよい。フィールドグラフの各円弧は、インクの断片のすべての可能なグループを表す、対応するセグメンテーショングラフに置き換えられる。各フィールド分割グラフは、フィールドグラフのフィールドのペナルティを含む円弧に付加される。各円弧は、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最初の評価となるペナルティを運ぶ。このペナルティは、インクの断片間のスペースや、セグメント画像のグローバルなベースラインとの適合性など、いくつかの単純な特徴と、いくつかの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。セグメンテーショングラフは、すべてのフィールド画像に対して可能なすべてのセグメンテーションを表す。対応するパスに沿ったアークペナルティを追加することで、1つのセグメント化されたフィールドに対するペナルティを計算することができる。前述と同様に、ペナルティを計算するために差分化可能な関数を使用することで、パラメータをグローバルに最適化することができる。

セグメンテーション機能では、様々なヒューリスティックを用いてキャンディデートカットを見つけることができる。最も重要なものの1つは、"hit and deflect "と呼ばれるものである。[115].このアイデアは、フィールド画像の上部から下に向かって線を投げることである。線が黒い画素にぶつかると、線はオブジェクトの輪郭に沿うように偏向される。線が上部プロファイルの局所的な最小値に当たったとき、つまり、黒画素を横切らずに下方に進むことができないとき、線はインクを通して垂直下方に伝搬されるだけである。このような2つの線が互いに出会うとき、それらは1つの切り口に統合されます。この手順を下から上へ繰り返すことができる。この方法によって、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_{rec} は、セグメンテーショングラフのすべてのセグメントアークを反復処理して、認識変換を行う。を、対応するセグメント画像上で認識する。私たちのの場合、認識器はLeNet-5、Convolutional Neural セクションIIで説明したネットワークで、その重みは調整可能なパラメーターの最大かつ最も重要なサブセットを構成している。ターがある。認識器は、セグメント画像を以下のいずれかに分類する。95クラス（印刷可能なASCIIフルセット）＋ゴミ箱クラス

未知の記号や整形不良の文字がある。入力グラフT_{rec}の各円弧は、出力グラフの96個の円弧に置き換えられる。これらの96個の円弧のそれぞれには、以下のうちの1つのラベルが含まれる。

と、入力（セグメンテーション）グラフの対応するアークのペナルティと、認識器によって計算された、画像を対応するクラスに分類することに関連するペナルティとの和であるペナルティと、がある。言い換えれば、認識グラフは、スコア化された文字クラスの重み付きトレリスを表している。このグラフの各パスは、対応するフィールドの可能な文字列を表している。パスに沿ってペナルティーを加えることで、この解釈に対するペナルティーを計算することができる。この文字列は有効なチェック量であるかもしれないし、そうでないかもしれない。

構成変換器T_{gram}、チェック量に有効な文字列を表す認識グラフのパスが選択される。この変換器は、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、整形された金額を構成する記号のすべての可能なシーケンスが含まれる。合成変換器の出力は解釈グラフと呼ばれ、認識グラフの中で文法に適合するすべてのパスが含まれる。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 (セクション VIII 参照)。微分可能な関数が、入力アークに付属するデータから出力アークに付属するデータを計算するために使用される。この場合、出力アークは 2 つのアークのクラス・ラベルと、2つの入力アークのペナルティ（認識器のペナルティと文法グラフのアーク・ペナルティ）を単純に合計して計算されたペナルティを受け取る。解釈グラフの各パスは、チェック上の1つのフィールドの1つのセグメンテーションの1つの解釈を表す。パスに沿ったペナルティの合計は、対応する解釈の「悪さ」を表し、文法だけでなく、プロセスに沿った各モジュールからの証拠を組み合わせています。

ビタビ変換は、最終的に累積ペナルティが最も小さい経路を選択し、最適な

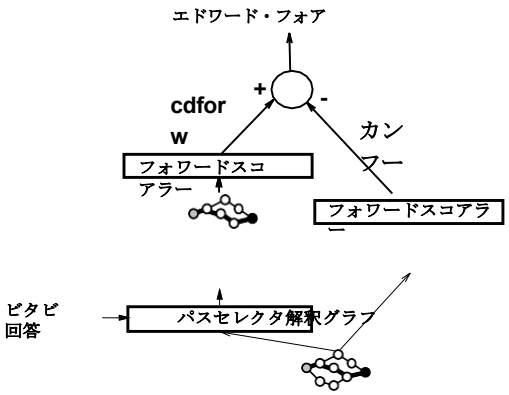


図34
信頼度算出に必要な追加処理信頼度を計算するために必要な追加処理。

文法的に正しい解釈をする。

B.勾配に基づく学習

このチェックリーディングシステムの各ステージには、調整可能なパラメータが含まれている。これらのパラメータの中には、フィールドロケータやセグメンテーションのパラメータなど、手動で調整可能なものもあるが、大部分は学習する必要がある。

システムを全体最適化する前に、各モジュールのパラメータは妥当な値で初期化される。フィールドロケータとセグメンテーションのパラメータは手作業で初期化し、ニューラルネットの文字認識器のパラメータは、あらかじめセグメント化されラベル付けされた文字のデータベースで学習することで初期化する。次に、正しい金額のラベルが付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。このとき、金額の明示的な分割は不要であり、小切手レベルで学習される。

大域的な学習手順で最小化される損失関数Eは、セクションVIで説明した判別可能な前進基準である。これは、（a）制約付き解釈グラフの前進ペナルティ（正しいラベル列によって制約される）と、（b）制約なし解釈グラフの前進ペナルティとの間の差である。派生物は構造全体を通じて逆伝播することができるが、実用的なのはセグメントまでである。

C.低信頼性チェックの拒否

を拒否できるようにするためです。

誤ったビタビ回答を伝える可能性がある場合、それらを信頼度で評価し、この信頼度が与えられた閾値より低い場合、そのチェックを拒否しなければなりません。2つの異なるチェックの正規化されていないビタビ・ペナルティを比較することは、どちらの答えを最もを、図21で説明したように、目的のシーケンスとしてビタビ回答を使用します。これは、図34に、まとめられている。

信頼性 = exp(E　ダーウィン）

D.結果

上記のシステムのバージョンを完全に実装し、機械印刷のビジネス・チェックでテストした。このシステムは基本的に汎用のGTNエンジンに、タスク固有のヒューリスティックをチェックとfpropメソッドにカプセル化したものである。その結果、記述するコード量はごくわずかであり、主に以前のセグメンテーション・トランスファをセグメンテーション・トランスファに適合させただけであった。手書きの小切手を扱うシステムは、GTNコンセプトを限定的に使用した初期の実装がベースになっています。

ニューラルネットワークの分類器は、まず印刷可能なASCIIセット全体にまたがる様々な出自の文字画像50万枚で学習されました。この画像には、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械で印刷された文字の両方が含まれています。追加画像は、画像の単純なアフィン変換を使用して元の画像をランダムに歪ませることによって生成された。さらに、チェック画像から自動的にセグメント化され、手作業で真偽判定された文字画像でネットワークを学習させた。このネットワークは、セグメンテーションエラーに起因する非文字を拒否するための初期学習も行った。次に、この認識器を小切手読み取りシステムに挿入し、小切手画像全体に対して、パラメーターの小さなサブセットをグローバルに（フィールドレベルで）学習させた。

機械印刷と自動的に分類された646枚のビジネスチェックについて、性能は82%が正しく認識されたチェック、1%がエラー、17%がリジェクトでした。これは、同じテストセットに対する従来のシステムの性能と比較することができる。正答率68%、エラー率1%、リジェクト率31%であった。機械印刷と判定されたのは、標準位置のドル記号に近い文字が機械印刷と判定された場合、または、標準位置に何もない場合、他の場所に少なくとも1つの礼儀量の候補が検出された場合である。この改善には、主に3つの原因があると考えられます。まず、ニューラルネットワークの認識器が大きくなり、より多くのデータで学習されるようになったこと。第二に、GTNアーキテクチャのため、新システムは、より多くのデータで訓練されました。
信頼するかを決定する際には無意味です。

信頼度の最適な尺度は、入力画像が与えられたときのビタビ回答の確率である。セクションVI-Eで見たように、ターゲットシーケンス（この場合、ビタビ回答である）が与えられると、識別的前方損失

関数はこの確率の対数の推定値である。したがって、信頼度の良い推定値を得るための簡単な解決策は、解釈グラフ（図33参照）を再利用して判別前方損失関数を計算することである。

は、以前のシステムよりもはるかに効率的な方法で文法的制約を利用することができました。第三に、GTNアーキテクチャは、ヒューリスティックのテスト、パラメータの調整、システムのチューニングに非常に柔軟性があります。この最後のポイントは、意外と重要です。GTNのフレームワークは、システムの「アルゴリズム：」の部分と「知識ベース：」の部分を分離し、後者を簡単に調整することができます。今回の課題では、グローバルトレーニングはパラメータのごく一部にしか関係しないため、グローバルトレーニングの重要性はごくわずかでした。

1995年にシステムインテグレーターが行った独立したテストでは、このシステムが他の市販の礼儀正しい金額読み取りシステムよりも優れていることが示された。このシステムは、NCRの小切手読み取りシステムのラインアップに統合されました。それは

は、1996年6月から全米のいくつかの銀行で実地され、以来、1日あたり数百万枚の小切手を読み取ってきました。

XI. 結論

パターン自動認識の短い歴史の中で、学習の役割を増やすことは、認識システムの全体的な性能を必ず向上させてきたと思われる。本論文で紹介するシステムは、この事実をさらに証明するものである。畳み込みニューラルネットワークは、手作業による特徴抽出の必要性をなくすことが示されている。グラフ変換ネットワークは、文書認識システムにおいて、手作業によるヒューリスティック、手動ラベリング、手動パラメータチューニングの必要性を低減することが示されている。学習データが豊富になり、コンピュータが高速化し、学習アルゴリズムに対する理解が深まれば、認識システムはますます学習に依存するようになり、その性能は向上するだろう。

バックプロパゲーション・アルゴリズムが多層ニューラルネットにおける単位割り当て問題をエレガントに解決したように、本論文で紹介するグラフ変換ネットワークのための勾配に基づく学習法は、新しい入力ごとに動的に機能アーキテクチャが変化するシステムにおいて単位割り当て問題を解決する。ここで紹介する学習アルゴリズムは、ある意味で、複雑で動的なアーキテクチャにおける勾配降下法の珍しい形態であり、勾配を計算するための効率的な逆伝播アルゴリズムにほかならない。本論文の結果は、大規模システムにおける学習のための一般的な組織原理として、勾配に基づく最小化法の有用性と関連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、勾配を逆伝播することができるグラフ変換として定式化できることが示された。システムの非学習的な部分においても、グラフ変換の設計思想は、ドメイン固有のヒューリスティック（例：セグメンテーションヒューリスティック）と一般的で手続き的な知識（生成された変換アルゴリズム）の間に明確な分離を提供する。

データ生成モデル（HMMなど）と最尤法は、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。大域的な識別損失関数に適用される勾配に基づく学習は、しばしば性能を犠牲にしてシステムアーキテクチャに強い制約を与える「正当化困難な」原理を用いることなく、最適な分類と棄却を保証するものである。

具体的には、本論文で紹介する手法とアーキテクチャは、パターン認識システムで遭遇する多くの問題に対する汎用的な解決策を提供するものである。

1.特徴抽出は伝統的に固定された変換であり、一般にそのタスクに関する専門家の事前知識から導かれる。これは、人間の設計者が入力に含まれるすべての重要な情報を捕らえることができるという、おそらくは誤った仮定に依存している。我々は、勾配に基づく学習を畳み込みニューラルネットワークに適用することで、入力に含まれるすべての関連情報を捉えることができることを示した。

ニューラルネットワークは、例から適切な特徴を学習することができる。このアプローチの成功は、NIST データベースを用いた広範な数字認識比較実験により実証された。

2.画像中のオブジェクトのセグメンテーションと認識は、完全に切り離すことはできない。我々は、セグメンテーションを早期に決定する代わりに、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、多数の仮説を並行して生成・評価し、全体の基準が最小化されるまで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、画像から分割された文字を得るためのハンドトゥルーシングはコストがかかり、文書全体や文字列の認識方法を考慮していない（特に、真の文字のように見えても、いくつかの分割候補が間違っている可能性があるという事実）。その代わりに、マルチモジュールシステムを訓練して、グローバルな性能評価を最適化する。この方法では、時間を消費する詳細な手作業による真偽判定を必要とせず、共通の目標に向かってモジュールを協調して訓練できるため、著しく優れた認識性能が得られる。

4.セグメンテーション、文字認識、言語モデルに内在する曖昧性は、オペレーショナルに統合される必要がある。我々は、これらの情報源を結合するためにタスクに依存した一連のヒューリスティックを用いる代わりに、入力に関する仮説の重み付けされたセットを表すグラフに一般化された変換手法を適用する統一フレームワークを提唱した。このアプローチの成功は、一日に何百万枚もの企業や個人の小切手を読み取る、商業的に利用されている小切手読み取りシステムで実証された：一般化されたトランスダクションエンジンは、わずか数百行のコードに存在する。

5.従来の認識システムは、個別に認識可能な対象を分離するために、多くの手作業によるヒューリスティックに頼っていた。空間変位ニューラルネットワークのアプローチは、Con-volutional Neural Networksの頑健性と効率性を利用し、明示的なセグメンテーションを完全に回避することができる。また、勾配に基づく学習により、セグメンテーションと認識を同時に自動学習することが可能である。

この論文では、グラフの少数の例を紹介します。しかし、このコンセプトは、ドメイン知識や状態情報がグラフで表現できる多くの状況に適用できることは明らかである。これは、多くの音声信号認識タスクや、視覚的な情景分析アプリケーションに当てはまります。将来的には、グラフ変換ネットワークをこのような問題に適用することで、自動学習への依存度を高め、詳細なエンジニアリングを少なくすることができると期待している。

付録

A.高速収束のための前提条件

先に見たように、Con-volutional Networksで使用するスカッシング関数は $f(a) = A \tanh(Sa)$ である。対称関数は収束が早いとされているが、重みが小さすぎると学習が極端に遅くなることがある。この問題の原因は、重み空間では原点が学習ダイナミクスの固定点であること、である。

は鞍点であるが、ほぼ全方向に魅力的である[116]。我々のシミュレーションでは、A = 1.7159

および$S = \frac{2}{3}$ (20]、[34]参照)。このようにパラメータを選択することで

の場合、$f(1) = 1$ と $f(-1) = -1$ の等式が満たされる。その理由は、通常の動作状態ではスカッシュ変換の総合利得は1程度であり、ネットワークの状態の解釈が簡略化されるからである。さらに、fの2次導関数の絶対値は+1および−1で最大となり、学習セッションの終盤での収束が改善される。このような特殊なパラメータの選択は単なる便宜的なものであり、結果には影響を与えない。

学習前に、重みは$-2.4/F_i$ ～の間の一様分布を用いてランダムな値で初期化される。

$2.4/F_i$ ここで、F_i は接続が属するユニットの入力(ファンイン)の数である。複数の接続が同じ重みを共有しているので、この規則を適用するのは難しいかもしれないが、我々の場合、同じ重みを共有する接続はすべて同一のファンインを持つユニットに属している。ファンインで割る理由は、重み付き和の初期標準偏差が各ユニットで同じ範囲にあり、シグモイドの正常動作領域内に入るようにしたいからである。初期重みが小さすぎると、勾配が非常に小さくなり、学習が遅くなる。また、大きすぎるとシグモイドが飽和し、勾配が非常に小さくなる。加重和の標準偏差は、入力が独立であれば入力数の平方根のようにスケールし、入力に高い相関があれば入力数に対して直線的にスケールする。いくつかのユニットは相関の高い信号を受信しているので、我々は第二の仮説を仮定することにした。

B.確率的勾配とバッチ勾配の比較

勾配に基づく学習アルゴリズムでは，パラメータの更新に2種類の方法のうちどちらかを用いることができる．最初の方法は「バッチ勾配」と呼ばれ，古典的な方法である．勾配は訓練セット全体にわたって蓄積され，正確な勾配が計算された後にパラメータが更新される．もう1つは「ストキャスチック勾配」と呼ばれる方法で，1つの学習サンプル（あるいは少数のサンプル）に基づいて部分的な，あるいはノイズの多い勾配を評価し，この近似勾配を用いてパラメータを更新する方法である．学習サンプルはランダムに選択することも，適切にランダム化されたシーケンスに従って選択することも可能です．確率版では勾配推定値にノイズが含まれるが，バッチ版に比べてパラメータの更新頻度が高くなる．実用上重要な経験則として、大規模で冗長なデータセットのタスクでは、バッチ版より確率版の方がかなり速く、時には桁違いの速さになることがあります[117]。この理由は理論的に完全には，この小さな部分集合に対して2回の完全な学習反復を行う．この考え方は、以下のような訓練セットに一般化することができる。

相似形は存在しないが

は解明されていないが、次のような極端な例で直感的に説明することが可能である。例えば、学習データベースが同じ部分集合の2つのコピーから構成されている場合を考えてみましょう。その場合、集合全体に対する勾配を累積すると、冗長な計算が行われることになる。一方、この訓練集合に対してStochas- tic Gradientを1回実行すると、以下のようになる。

冗長性がある場合実際、確率的アップデートは、冗長性がある場合、つまり、あるレベルの汎化が期待される場合には、より良いものになるはずです。

多くの著者はニューラルネットの学習に勾配降下の代わりに2次手法を用いるべきであると主張している。文献にはGauss-NewtonやLevenberg-Marquardtアルゴリズムなどの古典的2次法、Broyden-Fletcher-Goldfarb-Shanno法(BFGS)やLimited-storage BFGSなどの擬似Newton法、Conjugate Gradients (CG)法の様々なバージョンの推奨が豊富にあります[118]。残念ながら、上記の方法はすべて、大規模なデータセットで大規模なニューラルネットワークを学習するには不向きです。Gauss-Newton法とLevenberg-Marquardt法は、更新ごとに$O(N^3)$の演算が必要で、Nはパラメータ数であるため、中程度のサイズのネットワークでも非現実的なものになってしまいます。準ニュートン法では、1回の更新に$O(N)$回の演算が必要です。準ニュートン法は更新ごとに$O(N^2)$演算を必要とするが、それでも大規模なネットワークでは実用的でない。Limited-Storage BFGSとConjugate Gradientは、更新ごとにO(N)演算しか必要としないので、適切であると思われます。しかし、その収束速度は、連続する「共役降下法」の正確な評価に依存しており、これは「バッチ」モードでのみ意味を持ちます。大規模なデータセットでは、通常のバッチ勾配降下法よりもこれらの方法がもたらすスピードアップは、確率勾配の使用によってもたらされる膨大なスピードアップにはかないません。いくつかの研究者は、小さなバッチ、またはサイズが大きくなるバッチで共役勾配を使用することを試みましたが[119], [120]、これらの試みは、注意深く完全に調整された確率勾配を超えることはまだ証明されていません。我々の実験は、誤差面の偏心を最小化するようにパラメータ軸をスケーリングするストキャスティック法を用いて行われた。

C.確率的対角レーベンベルグ・マルカート

付録Bに示した理由により、我々は確率的更新法に従い、1つのパターンが提示されるたびに重みを更新することが望ましいと考える。パターンは一定のランダムな順序で提示され、学習セットは通常20回繰り返される。

我々の更新アルゴリズムは，確率的対角ルヴェンベルグ・マルカール法と呼ばれ，学習セットを通過するたびに，各パラメータ（重み）に対して個別の学習率（ステップサイズ）が計算される[20], [121], [34]．これらの学習率は，ヘシアン（2次微分）行列に対するガウス・ニュートン近似の推定値の対角項を使用して計算される．このアルゴリズムは学習速度の驚異的な向上をもたらすとは考えられないが、学習パラメータの大規模な修正を必要とせず、確実に収束する。また、ネットワークアーキテクチャと学習データの特異性に起因する損失関数の主要な悪条件を修正することができる。標準的な確率的勾配降下法に比べて、この方法を用いることによる追加コストはごくわずかである。

各学習反復において、特定のパラメータw_kは

は、以下の確率的更新ルールに従って更新されます。

$$W_k +- W_k - \frac{8E^p}{8w_k}. \qquad (18)$$

C_k

ここで、E^p、パターンpの瞬時損失関数である。

Convolutional Neural Networkでは、重みのため
を共有することで、偏微分$\frac{8E^p}{ブィ_k}$は偏微分の和となる。

を共有する接続に関して、デリバティブを使用することができます。
パラメータw_k：

$$\frac{8E^p}{8w_k} = \sum_{(イーヴィー_k} \frac{8E^p}{8u_{ij}} \qquad (19)$$

ここで、u_{ij}、ユニットjからユニットiへの接続重み、V_kは、i と j の間の接続がパラメータw_kを共有するような単位インデックスペア(i, j)の集合である、すなわち。

$$u_{ij} = w_k \quad V(i, j) E V_k \qquad (20)$$

前述のように、ステップサイズc_kは一定ではなく、損失関数の2階微分の関数である軸wに沿って$_k$。

$$c_k = \frac{}{\mu + h_{kk}} \qquad (21)$$

ここで、μは手で選んだ定数であり、h_{kk}は損失関数Eの二次導関数の推定値で、w_kに対するリスペクトである。h_{kk}が大きければ大きいほど、重みの更新は小さくなる。パラメータμは、2次導関数が小さいときにステップサイズが大きくなりすぎるのを防ぐもので、非線形最適化における「モデル信頼法」や「Levenberg-Marquardt法」[8]に非常によく似ています。接続重みに関する二次導関数からh_{kk}を計算する正確な式は以下の通りである。

$$h_{kk} = \sum_{(イーヴィー_k} \sum_{(イーヴィー_k} \frac{8^2 E}{8u_{ij} 8u_{kl}} \qquad (22)$$

しかし、我々は3つの近似を行う。最初の近似はは、上式の接続重みに関するヘシアンの非対角項を削除することである。

$$h_{kk} = \sum_{(i,j)EV_k} \frac{8^2 E}{8u_{ij}^2} \qquad (23)$$

は、ユニットiへの総入力（a_i）とする。興味深いことに、これらの2次導関数を計算する効率的なアルゴリズムが存在します。これは、バックプロパゲーションの手順と非常によく似ています。
は一次導関数の計算に使用されます[20], [121]。

$$\frac{8^2 Ep}{8a_i^2} = f'(a)^2 \sum_{キー_k} \frac{8^2 Ep}{8a_k^2} + f''(a_i) \frac{8E^p}{8x_i} \qquad (26)$$

残念ながら、これらの導関数を使用すると、よく知られた
全てのニュートンライクアルゴリズムに関連する問題これらの項は負になることがあり、その場合、グラディエント
アルゴリズムで、下り坂ではなく上り坂を移動するようにしました。そのため
という、よく知られたトリックがあります。
ガウス・ニュートン近似は、2次導関数の推定値が非負であることを保証するものである。ガウス・ニュートン近似は、推定関数（我々の場合はニューラルネットワーク）の非線形性を本質的に無視しますが、損失関数の非線形性は無視しません。2次導関数のガウス・ニュートン近似のバックプロパゲーション方程式は以下の通りです。

$$\frac{8^2 Ep}{8a_i^2} = f'(a)^2 \sum_{キー_k} \frac{8^2 Ep}{8a_k^2} \qquad (27)$$

これはシグモイドの微分値と重み値が2乗されることを除けば、1次微分のバックプロパゲーションの式と非常によく似ています。右辺は非負の項の積和なので、左辺の項は非負になります。

3つ目の近似は，式24の平均をトレーニングセット全体に対して行うのではなく，トレーニングセットの小さなサブセットに対して行うことである．さらに、誤差面の2次特性はかなりゆっくり変化するので、再推定は10回行う必要はない。本論文の実験では

の前に500パターンでh_{kk}を再推定した。
トレーニングセットを通過させる．トレーニングセットのサイズは60,000であるため，トレーニングセットの再推定にかかる追加コストが発生する．
h_{kk}は無視できる程度である。推定値は、平均化に使用したトレーニングセットの特定のサブセットに対して特に敏感ではない。このことは、誤差面の2次的な性質が主に
の詳細よりも、むしろネットワークの構造によるものです。

の統計量を算出する。このアルゴリズムは、重み当然ながら、$\frac{B^2E}{\text{プ}_i}$の項は、トレーニングの平均値である局所2次導関数のセットです。

$$\frac{8^2E}{8u^2_{ij}} = \frac{1}{P}\sum_{p=1}^{p} X \frac{8^2E^p}{8u^2_{ij}} \qquad (24)$$

接続重みに関する局所2次導関数は、下流ユニットの全入力に関する局所2次導関数から計算することができる。

$$\frac{82Ep}{8u^2_{\text{アイジェイ}}} = \frac{82Ep}{8a^2_i} x_j^2 \qquad (25)$$

ここでx_jはユニットjの状態、$\frac{B^2Ep}{\text{パ}_i^2}$は、2番目の$j$に関する瞬時損失関数の微分を行う。

共有ネットワークに特に有効である。
は、誤差面の条件不一致を引き起こす。共有のため、最初の数層における1つのパラメータが出力に大きな影響を与える可能性があります。その結果、このパラメータに対する誤差の2次導関数が、出力に大きな影響を与えることになる。
の場合は非常に小さくなる可能性があります。

のパラメータが変化する。上記のアルゴリズムはそのような現象を補償する。

バックプロパゲーションの他の多くの2次加速法と異なり、上記の方法は、確率的なモードを使用します。
ヘシアンの対角近似を使用します。
古典的なLevenberg-Marquardtアルゴリズムと同様に，2次導関数推定値が小さい場合にステップサイズが大きくなりすぎないように「安全係数：μ」を使用します．したがってこの方法は、「確率的対角レーベンベル法」と呼ばれます。
マーカート法