VI. グラフ変換のためのグローバルトレーニング ネットワーク

前項では、認識するまでのプロセスを説明しました。
に対して低いペナルティを与えるように認識器が訓練されていると仮定すると、ヒューリスティック・オーバー・セグメンテーションを使用して文字列を認識することができる。
正しく分割された文字に対しては正しいクラスラベルを、正しく分割された文字に対しては誤ったカテゴリに対して高いペナルティを、正しく分割されなかった文字に対しては全てのカテゴリに対して高いペナルティを課す。ここでは、文字セグメントの手動ラベル付けを必要とせず、上記を実現するための文字列レベルでのシステムの学習方法について説明する。この学習は、前節で説明した認識アーキテクチャとは若干異なるアーキテクチャを持つGTNを用いて行う。

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、各モジュールに何が期待されているかについての十分な先験的知識が存在する。例えば、Heuris-tic Over-Segmentationでは、1つの文字画像を個別にラベル付けし、それに対して文字認識器を訓練することができるが、誤ってセグメントされた候補を拒否するモデルを訓練するために、適切な非文字画像群を得ることは困難であろう。個別学習は簡単であるが、しばしば不足または不完全な監視情報（正しいセグメンテーションと誤った候補セグメントのラベル）を追加で必要とする。さらに、個別の学習は最適ではないことが示されている[67]。以下のセクションでは、GTNベースの手書き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方法について説明する。ビタビ学習，判別ビタビ学習，フォワード学習，判別フォワード学習である．最後の学習は、第II-C章で紹介したMAP基準をグラフベース・システムに一般化したものである。識別的前方学習は、音声認識におけるHMMの学習に用いられる、いわゆる最大相互情報量規準にやや類似している。しかし、我々の理論的根拠は古典的なものとは異なる。我々は確率論的な解釈には頼らず、勾配に基づく学習法において、識別的な学習は広範な原理の単純な一例であることを示す。

エラー訂正学習の

HMMのようなグラフベースのシーケンス認識システムのための学習方法は、音声認識の文脈で広く研究されている[28]。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基づくことを再要求し、可能な入力シーケンスの空間上で正規化された尤度を提供するものである。Baum-Welshアルゴリズムのような一般的なHMM学習法はこの正規化に依存している。ニューラルネットのような非生成モデルをシステムに組み込むと、正規化を維持することができない。この場合、識別的学習法などの他の手法を用いる必要がある。このような手法により、ニューラルネットワーク／HMM音声認識器を単語や文レベルで学習する方法が提案されている[71], [72], [73], [74], [75], [76], [77], [78]。
[29], [67].

他のグローバルに学習可能な配列認識システムは、グラフベースの技術に頼らず、統計的モデリングの難しさを回避している。その最たるものが、リカレント

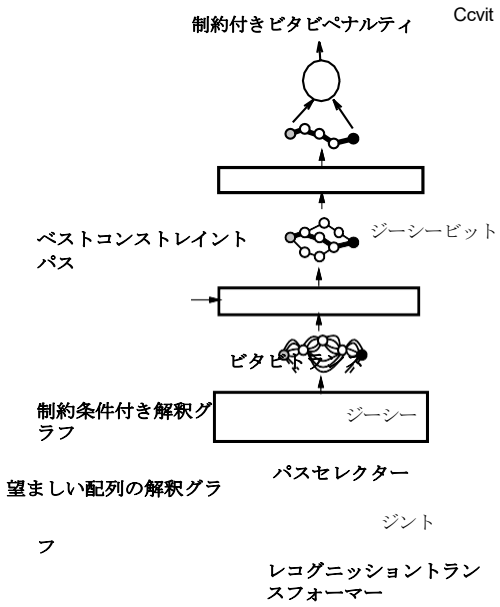


図19.ヒューリスティック・オーバーセグメンテーションに基づく文字列認識器のためのビタビ学習GTNアーキテクチャ。

ニューラルネット(RNN)。しかし残念ながら、初期の熱意にもかかわらず、勾配ベースの技術を用いたRNNの学習は、実際には非常に困難であることが判明している[79]。

以下に示すGTN技術は、音声認識のために開発されたグローバルな学習方法を簡素化し、一般化したものである。

A. ビタビトレーニング

認識時には、ビタビアルゴリズムで最もペナルティの少ないインタプリタグラフのパスを選択する。理想的には、このペナルティが最も低い経路が、できるだけ頻繁に正しいラベル列と関連することが望ましい。したがって、最小化すべき損失関数は、最も低いペナルティを持つ正しいラベル列に関連するパスのペナルティの学習集合における平均となる。学習の目標は、この「正しい：最も低いペナルティを持つパス」の平均ペナルティを最小化する認識器パラメータ（認識器がニューラルネットワークの場合、重み）のセットを見つけることであろう。この損失関数の勾配は、図19に示すGTNアーキテクチャを介したバックプロパゲーションにより計算することができる。この学習アーキテクチャは、解釈グラフとビタビ変換器の間にパスセレクタと呼ばれる余分なグラフ変換器が挿入されている以外は、前節で説明した認識アーキテクチャとほぼ同じである。この変換器は解釈グラフと目的のラベル列を入力とする。解釈グラフから、相関のある（所望の）ラベル列を含むパスを抽出する。その出力グラフG_cは制約付き解釈グラフ（HMMの文献ではフォースドアライメントとも呼ばれる）と呼ばれ、正しいラベル列に対応するすべてのパスを含む。この制約付き解釈グラフはビタビ変換器に送られ、1つのパスを持つグラフG_{cvit}が生成される。このパスは、最も低いペナルティを持つ「正しい：パス」である。最後に、パススコアラ変換器がG_{cvit}を受け取り、パスに沿ったペナルティを加算することで、その累積ペナルティC_{cvit}を単純に計算する。このGTNの出力は

現在のパターンに対する損失関数。

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

上記システムで必要とされるラベル情報は、希望する文字ラベルの並びのみである。解釈グラフのセグメンテーションの中から、最もペナルティの少ないものを選択するため、スーパーバイザ側では正しいセグメンテーションに関する知識は必要ない。

次に、ビタビ学習GTNを通して勾配を逆伝播するプロセスについて説明する。セクションIVで説明したように、先行するモジュールで勾配を計算し、その後そのパラメータを調整するために、勾配はGTNのすべてのモジュールを通じて逆伝播されなければならない。パススコアラを通して勾配を逆伝播するのは非常に簡単である。制約付きビタビパスG_{cvit}
上の個々のペナルティに関する損失関数の偏導関数は、損失関数が単純にそれらのペナルティの合計であるため、1に等しい。ビタビ変換器を介した逆伝播も同様に単純である。制約付きグラフG_c
のアーク上のペナルティに関するE_{vit}
の偏導関数は次のとおりである。 1
は、制約付きビタビパスG_{cvit}
に現れるアークに対して、そうでないものは0とする。なぜビタビ変換器のような本質的に不連続な関数をバックプロパゲートすることが正当なのでしょうか？その答えは、ビタビ変換器はmin関数と加算器をまとめたものにほかならないからである。第IV章で勾配をmin関数で逆伝播しても悪影響がないことを示した。パスセレクタ変換器を介した逆伝播はビタビ変換器を介した逆伝播と同様である。G_{int}
のうち G_c に現れる円弧は、G_c
の対応する円弧と同じ勾配、すなわち、G_{cvit}
に現れるかどうかに応じて 1 または 0
を持つ。他の円弧、すなわち、正しいラベルを含まないために G_c に分身を持たないものは、0
の勾配を持つ。認識変換器による順伝播中に、1
文字用の認識器のインスタンスがセグメンテーショングラフ内の各円弧に対して1つ作成された。認識器インスタンスの状態は保存される。G_{int}
の各アーク・ペナルティは、認識器インスタンスの個々の出力によって生成されるので、認識器の各インスタンスの各出力に対する勾配（1または0）を持つようになった。0でない勾配を持つ認識器出力は、正しい解析の一部であり、したがって、その値は押し下げられることになる。認識器出力に存在する勾配は、各認識器インスタンスを通して逆伝播されることができる。各認識器インスタンスについて、我々は認識器インスタンスパラメーター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・ベクトルを共有し、それらは互いのクローンに過ぎないので、認識器のパラメータ・ベクトルに対する損失関数の完全な勾配は、単に各認識器インスタンスによって生成される勾配ベクトルの和である。ビタビ学習は、定式化は異なるが、HMMベースの音声認識システムでよく使用される[28]。同様のアルゴリズムが音声認識システムに適用されている。

あるいは、ニューラルネットワークとタイムアライメントを統合したシステム [71], [72], [76]
や、ニューラルネットワークとHMMのハイブリッドシステム [29], [74] があります。
[75].

一見シンプルで満足できるように見えるが、このトレーニングアーキテクチャーには致命的となりうる欠陥がある。この問題はすでにセクションII-Cで述べた。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場合、損失関数の最小値は、認識器が常に正しい答えを与えるときではなく、入力を無視し、その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ときに達成される。これは崩壊問題として知られている。崩壊は、認識器の出力が同時にその最小値を取り得る場合にのみ発生する。一方、認識器の出力層が固定パラメータを持つRBFユニットを含む場合、そのような些細な解は存在しない。これは、固定されたパラメターベクトルを持つRBFの集合は、同時に最小値をとることができないためである。この場合、上記のような完全な崩壊は起こりません。しかし、より穏やかな崩壊の発生を完全に防ぐことはできない。なぜなら、損失関数は、認識器の出力が一定で些細な解に対して「フラットスポット」をまだ持っているからである。このフラットスポットは鞍点であるが、ほとんどすべての方向に魅力的であり、勾配ベースの最小化手順を用いてそこから抜け出すのは非常に困難である。RBFのパラメータが適応可能であれば、RBFの中心はすべて1つのベクトルに収束し、ニューラルネットワークはそのベクトルを生成することを学習し、入力を無視できるため、崩壊問題が再現される。RBFの幅も適応させると、別の種類の崩壊が起こる。この崩壊は、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。HMMベースの音声認識システムは、入力データに対して正規化された尤度を生成する生成システムなので、破綻は起こりません（これについては後で詳しく説明します）。また、入力画像から正しい解釈をする条件付き確率（クラスラベルの正しい並び）を最大化するなど、識別的な学習基準でシステム全体を学習させることで、破綻を回避することも可能です。

また、ビタビ学習では、ペナルティの低い（あるいは高得点の）競合解答が考慮されないため、解答のペナルティが信頼度の指標として信頼性高く使えないという問題があります。

B.判別型ビタビ学習

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、より信頼性の高い信頼値を生成することができます。そのアイデアは、正しい解釈で最も低いペナルティのパスの累積ペナルティを最小化するだけでなく、危険なほど低いペナルティを持つ競合する、そしておそらく間違ったパスのペナルティを何らかの方法で増加させることである。このような基準は、良い答えと悪い答えとを区別するため、識別的と呼ばれます。識別学習法は、個々のクラスを互いに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と見なすことができる。例えば

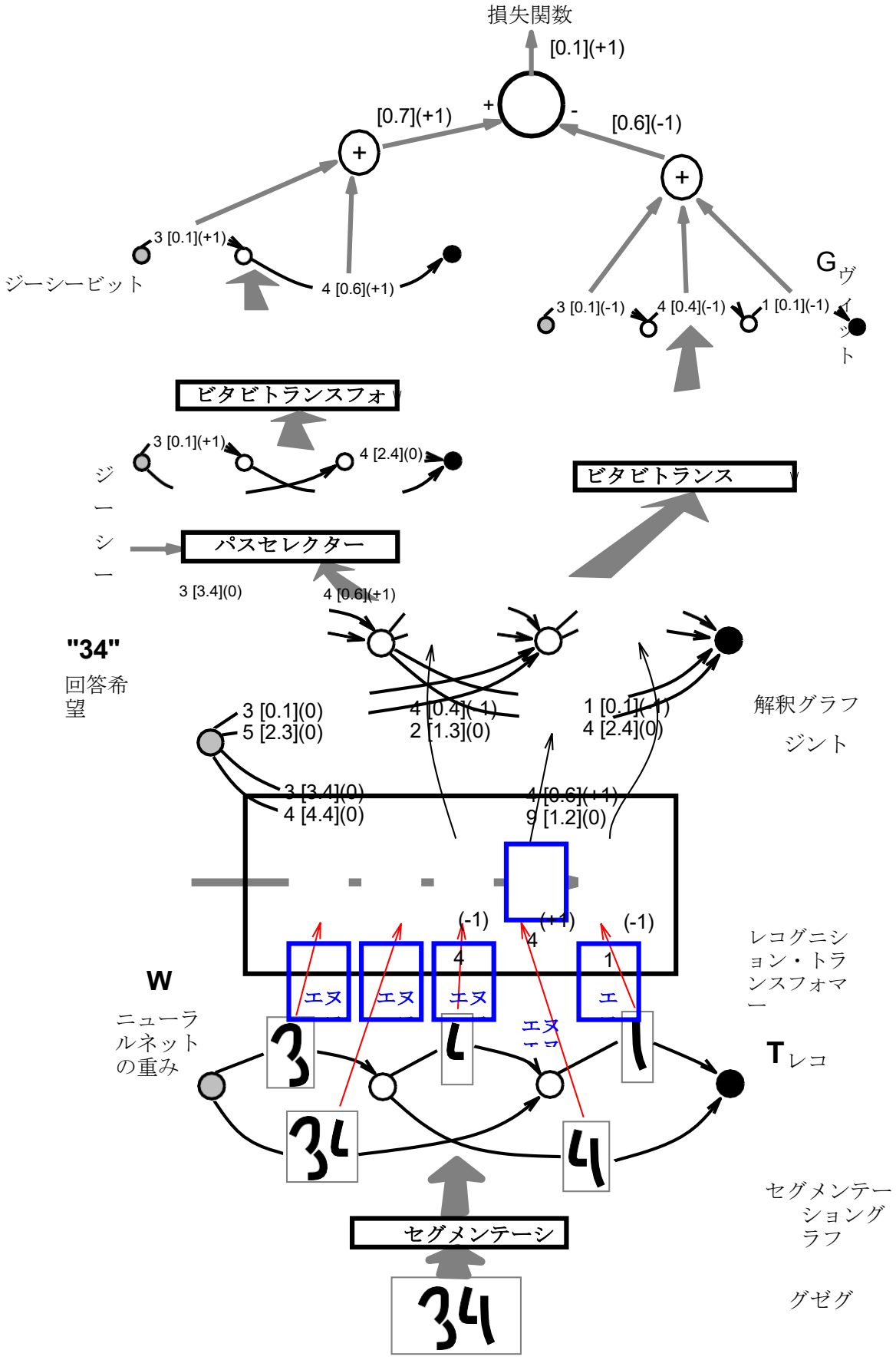

損失関数
[0.1](+1)
[0.7](+1)
[0.6](-1)
+
-
3 [0.1](+1)
4 [0.6](+1)
ジーシービット
3 [0.1](-1)
4 [0.4](-1)
1 [0.1](-1)
Gヴィット
ビタビトランスフォ
ビタビトランス
3 [0.1](+1)
4 [2.4](0)
ジーシー
パスセレクター
3 [3.4](0)
4 [0.6](+1)
"34"
回答希望
3 [0.1](0)
5 [2.3](0)
4 [0.4](-1)
2 [1.3](0)
1 [0.1](-1)
4 [2.4](0)
解釈グラフ
ジント
3 [3.4](0)
4 [4.4](0)
4 [0.6](+1)
9 [1.2](0)
(-1)
(+1)
(-1)
4
4
1
エヌ
エヌ
エヌ
エ
W
ニューラルネットの重み
レコグニション・トランスフォマー
Tレコ
セグメンテーション
セグメンテーショングラフ
34
グゼグ

図20.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文字列認識器のための識別ビタビ学習GTNアーキテクチャ。角括弧内の量は、前方伝播中に計算されたペナルティである。括弧内の量は後方伝搬で計算される偏微分である。

また，入力画像からクラスの条件分布をモデル化することは，各クラスに関連する入力データの個別の生成モデル（クラス事前分布を用いると，クラスと入力の全共同分布が得られる）よりも識別性が高い（分類面により焦点が当てられる）．これは、条件付きアプローチでは、入力データの分布に特定の形式を仮定する必要がないためである。

識別基準の一例として、制約付きグラフにおけるビタビパスのペナルティと、（制約のない）解釈グラフにおけるビタビパスのペナルティとの差、すなわち、最良の正しいパスのペナルティと、最良のパス（正しいか正しくないか）のペナルティとの差が挙げられる。対応するGTN学習アーキテクチャを図20に示す。図の左側は、非識別ビタビ学習に用いられるGTNと同一である。この損失関数は、認識器に誤って認識されたオブジェクトのペナルティを強制的に増加させるので、崩壊のリスクを減らすことができる。非識別学習はまた、誤り訂正手順の別の例として見ることができ、図20のGTN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との差を最小化する傾向がある。

判別ビタビ損失関数をE_{dvit}
とし、制約付きグラフのビタビパスのペナルティをC_{cvit}
、制約なし解釈グラフのビタビパスのペナルティをC_{vit} と呼ぶことにする。

$$E_{dvit} = C_{cvit} - C_{vit} \qquad (12)$$

制約グラフは解釈グラフのパスの部分集合であり、ビタビアルゴリズムは総ペナルティが最も小さいパスを選択するので、E_{dvit}
は常に正である。理想的なケースでは、2つのパスC_{cvit} とC_{vit} は一致し、E_{dvit} はゼロである。

判別型ビタビGTNで勾配をバックプロパゲートすると、先ほどの非判別型学習に「負」の学習が追加される。 図 20
は勾配がどのようにバックプロパゲートされるかを示している。左半分は非識別的ビタビ学習GTNと同じであり、したがってバックプロパゲーションも同じである。C_{vit}
は負の符号で損失に寄与するため、GTN
の右半分をバックプロパゲートされた勾配は -1
倍される。それ以外は左半分と同様である。G_{int}
の円弧上の勾配は、左半分から正の寄与を受け、右半分から負の寄与を受ける。G_{int}
の円弧のペナルティは、フォワードパスで "Y: "接続を介して2つのハーフに送られるので、2つの寄与は加算されなければならない。G_{int}
にある円弧は、G_{vit} にも G_{cvit}
にも現れず、勾配はゼロである。それらはコストに貢献しない。G_{vit} と G_{cvit}
の両方に現れる円弧もまた、勾配がゼロである。右半分の-1 の寄与は、左半分の+1
の寄与を打ち消す。言い換えれば、ある円弧が正当に答えの一部である場合、勾配はない。ある円弧が G_{cvit} に表示され、G_{vit}
には表示されない場合、勾配は +1
となります。$_{vitcvit}$この円弧は、G_{vit}
に入るためにもっと低いペナルティを受けるべきだった。そのアークには

のように、ペナルティは低いが、望ましい答えに含まれないため、より高いペナルティを課すべきであった。

この手法のバリエーションは、音声認識にも用いられている。Driancourt and Bottou [76]は、損失関数が固定値に飽和しているバージョンを使用した。これは、学習ベクトル量子化2（LVQ-2）損失関数の一般化として見ることができる[80]。この方法の他のバリエーションとして、ビタビ経路だけでなく、K-最適経路を用いるものがある。判別型ビタビアルゴリズムには非判別型のような欠点はないが、それでも問題がある。主な問題は、この基準ではクラス間にマージンを築かないことである。制約付きビタビ経路のペナルティがビタビ経路のペナルティと等しくなると同時に勾配がゼロになる。間違ったパスのペナルティが良いパスに危険なほど接近したときに押し上げることが望ましいと思われる。以下では、この問題に対する解決策を示す。

C. フォワードの採点、フォワードのトレーニング

ビタビパスのペナルティは認識の目的には完全に適しているが、状況の部分的な把握にしかならない。いくつかの異なるセグメンテーションに対応する最も低いペナルティのパスが、同じ答え（同じラベルシーケンス）を生成したと想像してください。その場合、同一のラベル列を持つ複数のパスは、そのラベル列が正しいという証拠になるため、1つのパスだけがその解釈を生成したときのペナルティよりも、その解釈に対する全体のペナルティは小さくなるはずだと主張することができる。複数の並列パスを含むグラフに関連するペナルティを計算するために、いくつかのルールを使用することができます。我々は、負の対数後置としてのペナルティの確率論的解釈から借用した組み合わせルールを使用する。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全てのパスの事後確率の和であるべきである。ペナルティに置き換えると、解釈のペナルティは個々のパスのペナルティの負の指数和の負の対数であるべきです。全体のペナルティは、個々のパスのすべてのペナルティよりも小さくなります。

ある解釈が与えられたとき、上記の係数を効率的に計算するためのフォワードアルゴリズムと呼ばれるよく知られた方法がある[28]。この方法で計算された、特定の解釈に対するペナルティをフォワードペナルティと呼びます。特定のラベル列と一致するパスだけを含む解釈グラフの部分グラフである制約グラフの概念をもう一度考えてみましょう。各ラベル列に対して1つの制約グラフが存在する（中には空のグラフもあり、これは無限のペナルティを持つ）。ある解釈が与えられたとき、対応する制約グラフ上で前進アルゴリズムを実行すると、その解釈に対する前進ペナルティが得られる。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方法で進行するが、各ノードで入力される累積ペナルティを結合するために使用される演算が、min関数ではなく、いわゆるlogadd演算である点が異なる。

関数を使用します。 $f_n = {}_{logaddi\in U_n}(c_i + f_{s_i})$ とする。 (13)

ここで $f_{立ち上がり} = 0$, U_n は、ノードnの上流アークの集合である。

c_i はアークiのペナルティ、そして

$$logadd(x_1 - x_2 - \ldots - x_n) = -\log(\sum_{i=1}^{n} e^{-x_i}) \quad (14)$$

なお，数値が不正確なため，最大の e^{-x_i} (最小のペナルティ)を対数で出す。

グラフに順方向アルゴリズムを適用することは、ニューラルネットワークに順方向プロパティを適用することと等しいと考えると、興味深いアナロジーが描ける。
ただし、乗算は加算に、加算は対数加算に置き換えられ、シグモイドは存在しない。

フォワードアルゴリズムを理解する一つの方法として、乗法的なスコア（例えば確率）を考えるのではなく

アークに加算されるペナルティ： score = exp(–penalty). である。
この場合、ビタビアルゴリズムは累積スコアが最大の経路を選択する（スコアは、その経路に沿って乗算される）。
一方、前方スコアは、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。順方向ペナルティは常に低い

しかし、ある経路が「支配的」な場合（ペナルティがかなり低い）、そのペナルティは前方ペナルティとほぼ等しくなる。前方アルゴリズム
このアルゴリズムは、隠れマルコフモデルを学習するための有名な Baum-Welsh アルゴリズム [28] の前進パスからその名前を取った。セクションVIII-Eでは、本研究とHMMの関係についてより詳しく述べる。

ビタビ・ペナルティに対する前進ペナルティの利点は、最も低いペナルティを持つものだけでなく、答えを生成するすべての異なる方法を考慮することである。これは、セグメンテーションに曖昧さがある場合に重要である。同じラベルシーケンスに関連する2つのパス C_1 と C_2 の組み合わせの前方ペナルティは、別のラベルシーケンスに関連するパス C_3 のペナルティよりも小さいかもしれないが、C_3 のペナルティは C_1 または C_2 のいずれかより小さいかもしれないからである。

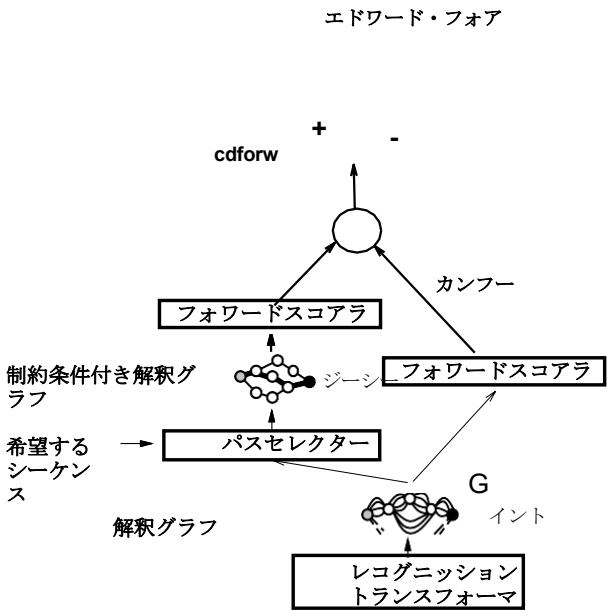


図21.
ヒューリスティック・オーバー・セグメンテーションに基づく文字列認識器のための識別的前方学習GTNアーキテクチャ。

G_c

$$\frac{\partial E}{\partial f_n} = e^{-f_n} \sum_{アイイーディーエヌ} \frac{\partial E}{\partial f_{d_i}} e^{f_{d_i} - c_i} \quad (15)$$

ここで、D_n = farc i with source s_i = ng は、ダウン・ミュージックの集合である。
ストリームアークがノードnから発生する。以上の導関数から、アークペナルティに関する導関数が求められる。

$$\frac{\partial E}{\partial c_i} = \frac{\partial E}{\partial f_{d_i}} e^{-c_i - f_{si} + f_{di}} \quad (16)$$

フォワード学習GTNは、先に紹介したビタビ学習GTNを少し変更しただけのものである。図19のビタビ変換器を、解釈グラフを入力とし、そのグラフの前方ペナルティを出力とする前方スコアラーに変えるだけで十分である。そして、正解を含むすべての経路のペナルティを、最良の経路のペナルティだけでなく、低くするのである。

前方ペナルティ計算（前方変換器）の逆伝播は、ビタビ変換器の逆伝播とは全く異なる。入力グラフのすべてのペナルティは前方ペナルティに影響を与えるが、
低ペナルティパスに属するペナルティはより強い影響を与える。グラフの各nノードで計算された前方ペナルティ f_n に対する微分の計算は、グラフを介した逆伝播によ

って行われる。

これは、ビタビスコアラーと変換器を介したバックプロパゲーションの「ソフト」バージョンと見ることができる。G_c
のすべてのアークが損失関数に影響を与える。低ペナルティ・パスに属するアークほど大きな影響力を持つ。経路選択器による逆伝播は前と同じである。G_{int}
G_c に分身を持つアークに関する導関数は、G_c の対応するアークから単純にコピーされる。
他のアークに関する導関数は0である。

また、Bridle の o:-net モデル [73] や Haffner の o:/-TDNN モデル [81] など、音声認識システムの学習に前方スコアラによる勾配の逆伝播の考え方を適用した著者もいるが、次節で述べるように、これらの著者は識別的な学習を推奨している。

D.判別型フォワード学習

前方ペナルティに含まれる情報は、識別的前方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る。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。この事後確率は、制約付き前方ペナルティのマイナス指数と、制約なし前方ペナルティのマイナス指数で正規化されたものとして定義される。制約付きグラフの前方ペナルティは常に制約なし解釈グラフの前方ペナルティより大きいか等しいことに注意してください。理想的には、制約付きグラフの前方ペナルティは、以下のように等しいことが望ましい。

完全な解釈グラフの前方ペナルティこの2つの量の等価性は、正しいラベルシーケンスを持つパスの合計ペナルティが、他のすべてのパスのペナルティに比べて無視できるほど小さい場合、あるいは正しい解釈を持つパスに関連する事後確率がほぼ1である場合に達成され、これがまさに我々が望むことである。対応するGTN学習アーキテクチャを図21に示す。

差分をE_{dforw}とし、制約付きグラフの前方ペナルティをC_{cforw}、完全解釈グラフの前方ペナルティをC_{forw}と呼ぶことにする。

$$E_{dforw} = C_{cforw} - C_{forw} \quad (17)$$

E_{dforw}は常に正である。なぜなら、制約グラフは解釈グラフのパスの部分集合であり、グラフの前方ペナルティは常にこのグラフの部分グラフの前方ペナルティより大きいからである。理想的なケースでは、不正確なパスのペナルティは無限に大きいので、2つのペナルティは一致し、E_{dforw}はゼロになります。ボルツマンマシンのコネクショニストモデルに詳しい読者は、制約付きグラフと制約なしグラフが、ボルツマンマシンアルゴリズムの「クランプ：（出力変数のオブサーブ値による制約）」と「フリー：（制約なし）」のフェーズに類似していると認識するかもしれない[13]（Boltzmann machine algorithm）。判別可能なForward GTNで導関数をバックプロパゲートすると、Viterbiの場合よりも勾配が均等に分散される。導関数は、図21のGTNの左半分から解釈グラフまでバックプロパゲートされる。導関数は否定され、右半分に逆伝播され、各弧の結果は左半分からの寄与に加えられる。G_{int}の各アークは導関数を持つようになる。正しい経路の一部である円弧は、正の導関数を持つ。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ペナルティを持つ場合、非常に大きくなる。同様に、低ペナルティの不正確なパスの一部であるアークに関する導関数は、大きな負の導関数を持つ。一方、正しい解釈に関連するパスのペナルティが他のすべてのパスよりはるかに小さい場合、損失関数は非常に近い0であり、勾配はほとんど逆伝播されない。したがって、学習は、分類誤りをもたらす画像の例に集中し、さらに、その誤りを引き起こす画像の断片に集中する。識別的前方学習は、グラフのような「動的」データ構造を扱う学習マシーンの悪名高い単位割り当て問題を解決する、エレガントで効率的な方法である。より一般的には、学習機械が離散的な代替解釈を選択しなければならないすべての状況で、同じ考えを用いることができる。

前述したように、解釈グラフのペナルティに関する導関数は、文字認識器のインスタンスにバックプロパゲートすることができる。文字認識器へのバックプロパゲーションは、そのパラメータに導関数を与える。異なる候補セグメントに対する全ての勾配寄与は、1つのペア（入力画像、正しいラベル列）、つまり、トレーニングセット内の1つの元例に関連する総勾配を得るために、合計される。そして、確率的勾配降下のステップを適用して、パラメータを更新することができる。

E.識別トレーニングに関する備考

以上の議論において、大域的な学習基準には確率的な解釈を与えたが、グラフの円弧の個々のペナルティには確率的な解釈を与えなかった。これには十分な理由がある。例えば、あるペナルティが異なるクラスラベルに関連している場合、それらは（1）和が1にならなければならない（クラス後置）、あるいは（2）入力領域上で積分して1にならなければならない（尤度）、などである。

まず、最初のケース（クラスの後置正規化）について説明します。例えば，画像の一部が有効な文字クラスに対応しない場合，分割候補の一部が誤っている可能性があるため，局所的にすべてのクラスを拒否するために重要な情報を排除してしまう可能性がある[82]．しかし、このようなクラスを確率的に特徴付け、システムを訓練することは困難であるため、いくつかの問題が残っている（未見またはラベル付けされていないサンプルの密度モデルが必要である）。

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な役割を果たす。しかし、残念ながら、これらの方法は識別的な学習基準に適用することができず、勾配に基づく方法を用いるしかない。勾配に基づく学習を行う際に、確率的な量の正規化を強制することは、複雑で非効率的で時間がかかり、損失関数の非条件付けを引き起こす。

したがって、[82]に従い、正規化は可能な限り（実際、システムの最終決定段階まで）延期することが望ましい。正規化しなければ、システムで操作される量子タイは直接的な確率論的解釈 を持たない。

ここで、2番目のケース（入力の生成的モデルを用いる）について説明しよう。生成モデルは、まず各クラスに対して独立した密度モデルを構築し、そのモデルに基づいて分類決定を行うことで、間接的に境界を構築する。これは、学習の最終目標（この場合は分類判定面を学習すること）に着目していない点で、識別的なアプローチとは言えない。理論的な議論[6], [7]は、分類のための判別関数を得ることが真の目標であるときに入力密度を推定することは、最適でない戦略であることを示唆している。理論的には，高次元空間の密度を推定する問題は，決定境界を求める問題よりもはるかに非論理的である．

システムの内部変数が直接的な確率論的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。実際、図 21 の GTN において、あるラベル列が「望ましい列」として与えられると仮定すると、マイナス E_{dforw} の指数は、入力が与えられたときのそのラベル列の事後確率の推定値と相互予測することが可能である。もう一つのアプローチは、誤分類の数の近似値を直接最小化することである [83] [76]。我々は，判別可能な前進損失関数を用いることを好む.

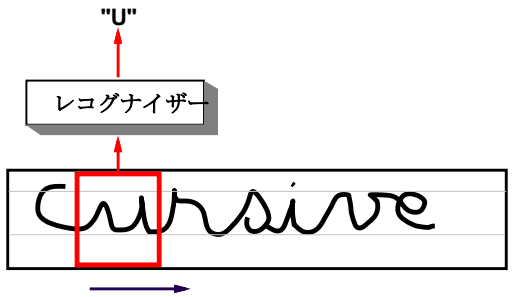


図
22.明示的なセグメンテーションは、入力フィールドの可能な限りの位置でレコグナイザーを掃引することで回避することができる。

最適化の際に数値的な問題が少なくなる。我々は、これが棄却戦略の基礎となるスコアを得るための良い方法であることをセクションX-
Cで見ることになる。ここで重要なことは、分類モデルに適切と思われるパラメタリゼーションを自由に選択できることです。ある特定のパラメタリゼーションが明確な確率的解釈を持たない内部変数を使用するという事実は、そのモデルを正規化された量を操作するモデルよりも正当性が劣るものにすることはない。

グローバルトレーニングや識別トレーニングの重要な利点は、学習が最も重要なものに集中することです。

を学習し、セグメンテーションアルゴリズムからの曖昧性と文字認識器からの曖昧性を統合するように学習する。セクションIXでは、オンライン手書き文字認識システムの実験結果を示し、個別学習に対するグローバル学習の優位性を確認した。また、ニューラルネットワークとHMMのハイブリッドによる音声認識の実験でも、大域的学習によって顕著な改善が見られた[77], [29], [67], [84]。

VII. 複数物体認識空間変位ニューラルネットワーク

.

ヒューリスティックを使って文字列の画像を明示的にセグメント化する代わりに、単純な方法がある。このアイデアは、図22に示すように、単語または文字列全体のノーマライズされた画像のすべての可能な位置で認識器をスイープすることである。この手法では、システムが基本的に入力の可能なすべてのセグメンテーションを調べるため、セグメンテーションのヒューリスティックは必要ない。しかし、この方法には問題がある。第一に、この方法は一般的に非常に高価である。認識器は、入力上のすべての可能な位置、あるいは、少なくとも十分大きな位置のサブセットを適用しなければならない。第二に、認識器が認識すべき文字の中心にあるとき、中心文字の近傍は認識器の視野内に存在し、中心文字に触れる可能性がある。したがって、認識器は入力フィールドの中心にある文字を正し

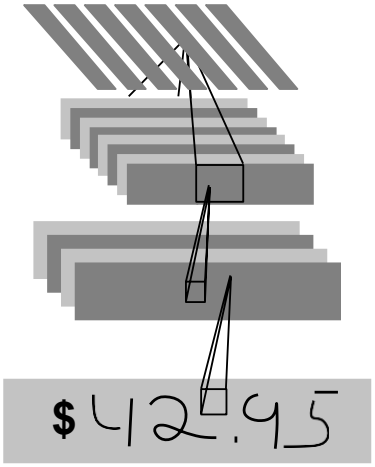


図23 空間変位ニューラルネットワークSpace Displacement Neural Networkは、広い入力フィールドに複製された畳み込みネットワークである。

文字列内の文字のサイズやベースラインの位置は、大きく変化することがある。そのため、認識装置はシフトやサイズの変動に対して非常に堅牢でなければならない。

これら3つの問題は、入力フィールド上で畳み込みネットワークを複製すれば、エレガントに回避することができる。
く認識する必要がある。第三に、単語や文字列は、完全にサイズ正規化することができません。個々の

まず、III 章で示したように、畳み込みニューラルネットワークは、入力画像のシフトやスケール変化、ノイズや余計なマークに対して非常に頑健である。これらの特性により、前項で述べた後2つの問題を解決することができる。第二に、畳み込みネットワークは、大きな入力フィールドに複製された場合、計算量を大幅に削減することができる。図23は、Space Displacement Neural Network または SDNN
[27]とも呼ばれる、複製された畳み込み回路網である。一般に、認識器のスキャンは法外なコストがかかるが、畳み込みネットワークは、大きな可変サイズの入力フィールド上で非常に効率的にスキャンまたは複製することができる。ある畳み込みネットのインスタンスと、近くの場所にあるその分身を考えてみよう。畳み込み網の性質上、入力の同じ場所を見る2つのインスタンスのユニットは同じ出力を持つので、その状態を2回計算する必要はない。2つのネットワークインスタンスで共有されていない新しい状態の薄い「スライス」だけが再計算される必要がある。すべてのスライスを合わせると、その結果は、特徴マップが水平方向に大きくなっていることを除けば、元のネットワークと同じ構造を持つ、より大きな畳み込みネットワークとなる。つまり、畳み込みネットワークの複製は、畳み込みが行われるフィールドのサイズを大きくし、それに応じて出力層を複製することで可能となる。出力層は事実上畳み込み層となる。受容野が初等オブジェクトを中心とする出力は、そのオブジェクトのクラスを生成し、その中間の出力は、特徴がないことを示すか、ゴミを含むかもしれない。出力は、入力フィールドのすべての可能な位置にオブジェクトが存在することの証拠と解釈することができる。

SDNNアーキテクチャは特に魅力的だと思います。

信頼できる分割ヒューリスティックが存在しない筆記体の認識。SDNN
のアイデアは非常に古く、そのシンプルさが非常に魅力的で
あるが、上記のように認識器への要求が非常に高いため、最近まで 広く関心を集めることはなかった [26], [27]。音声認識では、認識器が少なくとも1桁小さいので、Haffnerの多状態TDNNモデル [78], [85]のように、複製された畳み込みネットの方が実装が簡単である。

A.GTNによるSDNNの出力の解釈

SDNNの出力は、入力の対応する位置で特定のクラスラベルを持つキャラクターを見つける尤度、ペナルティ、またはスコアをコード化したベクトルのシーケンスである。このベクトル列から最適なラベル列を抽出するために、ポストプロセッサが必要となる。
クエンスを生成する。図25にSDNNの出力例を示す。非常に頻繁に、個々の文字が複数の隣接する認識器のインスタンスに よって発見される。

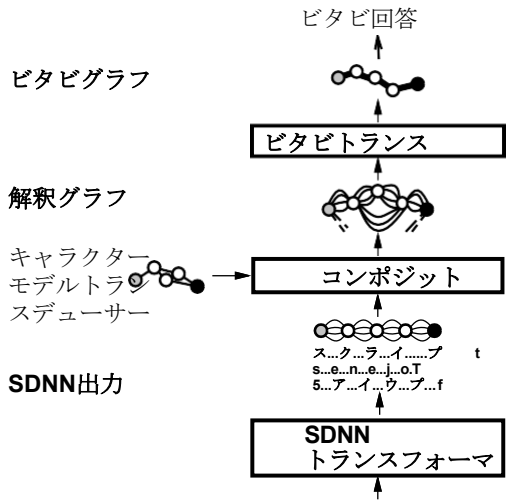
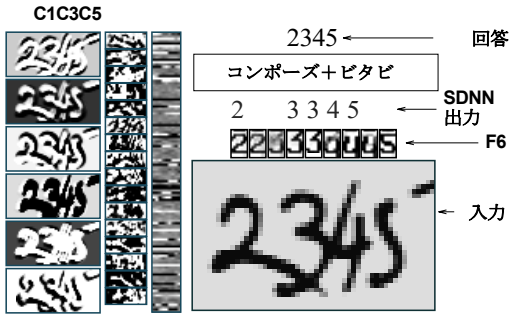


図24
グラフ変換器グラフ変換器により、SDNNの出力から最適な解釈を引き出せる。



を翻訳する必要があります。また、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、文字が誤って検出されることがよくあります。例えば、"4:
"の右3分の1しか見ていない認識インスタンスは、"1
"というラベルを出力するかもしれません。このような余計な文字を出力列から排除し、最適な解釈を引き出すにはどうすればよいのだろうか。これは、図24に示すように、2つの入力グラフを持つ新しいタイプのグラフ変換器を用いて行うことができる。SDNNが生成するベクトル列は、まず連続するノードのペア間に複数のアークを持つ線形グラフにコード化される。特定のノードのペア間の各円弧には、考えられるカテゴリのラベルと、その位置でそのクラスラベルに対してSDNNが生成するペナルティが含まれる。このグラフをSDNN出力グラフと呼ぶ。変換器への2番目の入力グラフは、文法変換器、より具体的には有限状態変換器[86]であり、クラスラベルの入力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エンコードする。変換器は、各弧にラベルのペアと場合によってはペナルティを含む重み付き有限状態機械（グラフ）である。有限状態機械と同様に、トランスデューサーはある状態にあり、観測された入力シンボルが円弧に付けられたシンボルペアの最初のシンボルと一致すると、新しい状態へと円弧をたどる。このときトランスデューサーは、入力記号のペナルティと円弧のペナルティを合わせたペナルティとともに、ペアの2番目の記号を出力する。従って、変換器は、重み付き記号列を別の重み付き記号列に変換する。図 24
のグラフ変換器は、認識グラフと文法変換器との間の合成を行う。この演算は、認識グラフのすべてのパスに対応する可能な配列を取り、文法トランスデューサのパスとマッチングさせる。この合成により、解釈グラフが生成される。解釈グラフには、対応する各出力ラベル列のパスが含まれる。この合成操作は組合せ的に難解に見えるかもしれないが、効率的なアルゴ
リズムが存在することが判明した（詳細は第 VIII 節で述べる）。

図25 SDNNによる複数文字認識の例
SDNNによる複数文字認識の一例。
SDNNでは、明示的なセグメンテーションは行われない。

B.SDNNを用いた実験

一連の実験では、LeNet-5は、セグメンテーションなしに複数のキャラクターを認識するように複製されることを目標に学習された。データは、先に述べた修正NISTセットから以下のように作成した。学習画像は、中央の文字と、学習セットからランダムに選ばれた2つの脇文字で構成されている。文字の境界ボックスの間隔は-1〜4ピクセルの間でランダムに選択された。また、中央の文字が存在しない場合は、空白クラスが出力される。さらに、学習画像は10%の塩コショウノイズで劣化させた（ランダムな画素反転）。

図25と図26はLeNet-5SDNNが複数の文字の認識に成功した例である。ヒューリスティック・オーバー・セグメンテーションに基づく標準的な手法では、これらの例の多くで惨憺たる結果になる。これらの例からわかるように、このネットワークは顕著な不変性と耐ノイズ性を示している。不変性にはフィードフォワードニューラルネットより高度なモデルが必要であると主張する著者もいるが[87]、LeNet-5はこれらの特性を大きく発揮している。

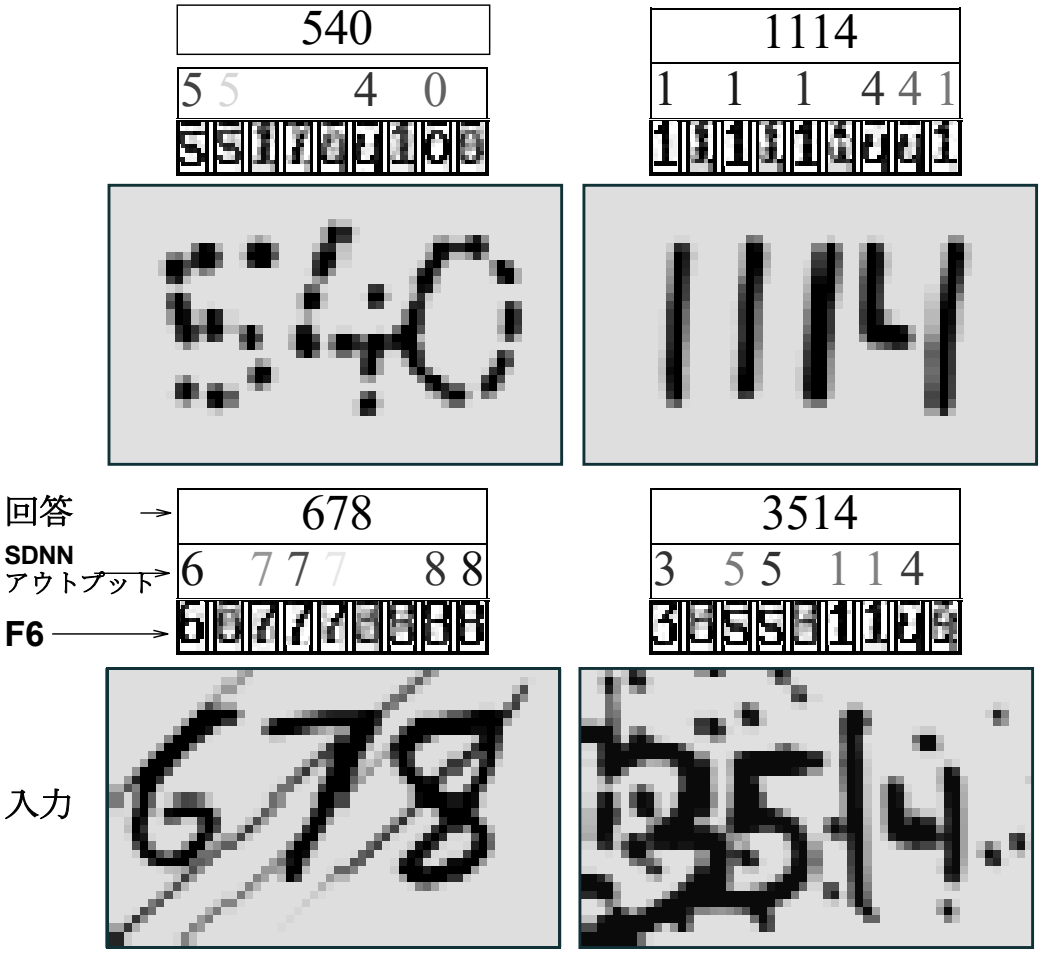


図26.SDNNを数字列のノイズの多い画像に適用したもの。SDNNの出力に示される数字は、勝利したクラスラベルを表し、高ペナルティの回答はより明るいグレーレベルである。

同様に、重なり合う複数の物体を正確に認識するためには、いわゆる特徴束縛問題を解決する明示的なメカニズムが必要であることが示唆されている[87]。図25と26に見られるように、SDNNは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る。SDNNはまた、文字を形成する切断されたインクの断片を正しくグループ化することができる。その好例が図26の上半分に示されている。左上の例では、4と0はそれ自身よりも互いにつながっているが、システムは4と0を別々のオブジェクトとして正しく識別している。右上の例は、いくつかの理由で興味深い。まず、システムは3つの個別のものを正しく識別しています。次に、切断された4の左半分と右半分は、幾何学的な情報がなくても、左半分をその左または右の縦棒に関連付けることを決定することができ、正しくグループ化されています。右半分の4はSDNN出力に誤った1を出現させるが、これは文字モデル変換器によって取り除かれ、連続した出力に文字が出現するのを防ぐことができる。

SDNNのもう一つの重要な利点は、「簡単」であることです。そのため、並列ハードウェアに実装することができます。文字認識や画像前処理のアプリケーションでは、特別なアナログ／デジタルチップが設計され使用されている[88]。しかし、IntelのMMXのような精度の低いベクトル演算命令を持つ従来のプロセッサ技術の急速な進歩は、専用ハードウェアの成功をせいぜい仮定に過ぎない。

LeNet-5

SDNNの短いビデオクリップは、http://www.research.att.com/~yann/ocrで見ることができます。

C.SDNNのグローバルトレーニング

上記の実験では、文字列画像は個々の文字から人工的に生成されたものであった。その利点は、重要な文字の位置とラベルがあらかじめ分かっていることである。実際の学習データでは、文字列に対するラベルの直列は一般に入手可能であるが、入力画像中の対応する各文字の正確な位置は不明である。

前節で述べた実験では、非常に単純なグラフ変換器を用いてSDNNの出力から最適な解釈を抽出した。SDNNの大域的な学習は、VI節で述べたようなアーキテクチャに配置されたこのようなグラフ変換器を介して勾配を逆伝播することによって行うことが可能である

。

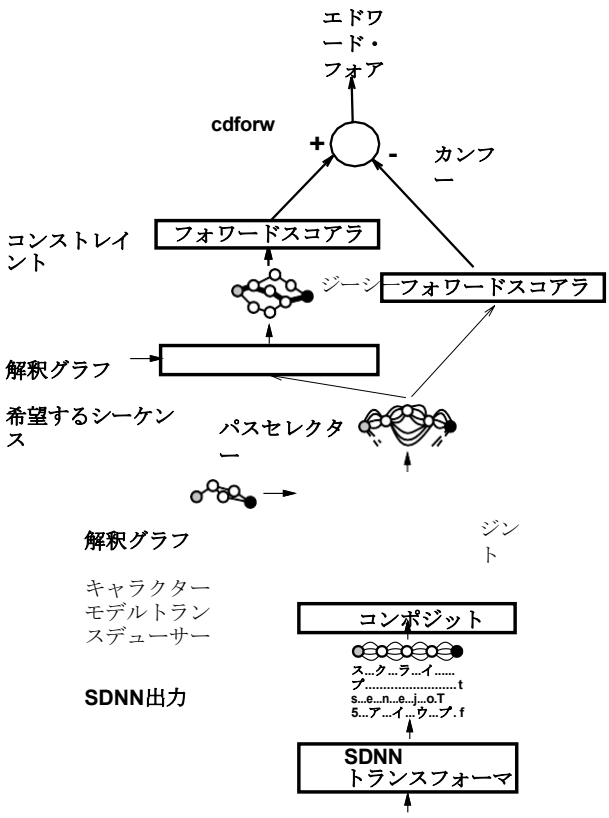


図27.グローバルに学習可能なSDNN/HMMハイブリッドシステムをGTNとして表現したもの。

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同じようなものである。グローバルに学習された可変サイズのTDNN/HMMハイブリッドは、音声認識やオンライン手書き文字認識に利用されている[77], [89], [90], [67]．また、空間変位ニューラルネットワークは、HMMや他の弾性マッチング手法と組み合わせて、手書き単語認識に用いられている[91]、[92]。

図27は、SDNN/HMMハイブリッドをDiscriminative Forward Criterionで学習するためのグラフ変換器アーキテクチャを示す。上側は図21の上側と同等である。右側では、認識グラフと文法との合成により、すべての可能な法的解釈を持つ解釈グラフが得られる。左側は、所望のラベルの並びを持つパスのみを含む文法との合成を行う。これは前節で用いたパスセレクタとやや類似の機能を持つ。セクションVI-Dと同様に、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。コムポジション変換器をバックプロパゲートするためには、認識グラフのどのアークがインタープリテーショングラフのどのアークを発生させたかを記録しておく必要がある。認識グラフ

郵便番号認識[91]や、より最近のオンライン手書き認識[38]の実験では、グローバルに学習されたSDNN/HMMハイブリッドのアイデアが示されています。SDNNはOCRにとって非常に有望で魅力的な技術であるが、これまでのところ、ヒューリスティック・オーバー・セグメンテーションよりも良い結果は得られていない。我々は、これらのモデルでより多くの経験を積むことで、これらの結果が改善されることを期待している。

D.SDNNによる物体検出とスポッティング

SDNNの興味深い応用例として、物体検出とスポッティングがある。コンボリューショナルネットワークの不変性ネットワークは、大規模なフィールドで複製できる効率性と相まって、大規模画像における「総当り的なオブジェクトの発見と検出」に利用できることが示唆される。主なアイデアは、1つのConvolutional Networkを訓練して、背景の画像から目的のオブジェクトの画像を区別することである。利用モードでは、ネットワークは分析される画像全体をカバーするように複製され、それによって2次元空間変位ニューラルネットワークが形成される。SDNNの出力は2次元の平面であり、その中で活性化されたユニットがのある円弧に関する導関数は、その円弧を始点とする解釈グラフの全円弧に関する導関数の和に等しい。また、文法グラフのペナルティについても導関数を計算することができ、それらを学習することができる。前の例と同様に、非識別的基準を用いると、ネットワークの出力RBFが適応的である場合、破綻する可能性があるので、識別的基準を用いなければならない。上記の学習方法は、HMMの用語で等価に定式化することができる。の初期の実験では

は、対応する受容野に注目する物体が存在することを示す。画像内の検出対象物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提示し、複数の解像度の結果を結合することが可能である。このアイデアは、顔の位置検出[93]、封筒の宛名ブロックの位置検出[94]、ビデオにおける手の追跡[95]などに応用されている。

本手法を説明するために、[93]に記載されているような画像中の顔検出の場合を考えてみることにする。まず、様々なスケールで顔を含む画像を収集する。これらの画像はゼロ平均ラプラシアンフィルタによってフィルタリングされ、全体的な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、これらの画像から顔と非顔の学習サンプルが手動で抽出される。次に、顔の部分画像は、かなり大きな変動（2倍以内）を維持しながら、顔全体の高さが約20ピクセルとなるようにサイズ正規化される。背景サブ画像のスケールはランダムに選択される。これらのサンプルに対して1つの畳み込みネットワークが学習され，顔部分画像と非顔部分画像の分類が行われる。

シーン画像を解析する場合、まずラプラシアンフィルタを通してフィルタリングし、2のべき乗の解像度でサブサンプリングする。ネットワークは複数の解像度の画像のそれぞれに対して複製される。複数の解像度の結果を結合するために、単純な投票技術が用いられる。

前節で説明したグローバルトレーニング手法の2次元版を使用することで，トレーニング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軽減することができる[93]．各可能な位置は、代替解釈、すなわち、開始ノードと終了ノードのみを含む単純なグラフのいくつかの平行な弧の1つとみなされる。

他の著者らは、顔検出にニューラルネットワークやサポートベクトルマシンのようなクラスシフィエーションを使用し、大きな成功を収めている[96]、[97]。彼らのシステムは、複数のスケールでネットワークに画像を提示するというアイデアを含め、上記のものと非常によく似ています。しかし、それらの