TR-I-0089

時間遅れ神経回路網を用いた
音韻/音節スポッティング
Spotting Phonemes and Syllables Using Time-Delay Neural Networks

沢井秀文 宮武正典 A.ワイベル 鹿野清宏
H.SAWAI, M.MIYATAKE, A.WAIBEL and K.SHIKANO

1989. 7.20.

概要

音韻または音節スポッティングは、もしそれらが精度よく達成できれば、単語音声や連続音声認識に有用である。我々は、時間遅れ神経回路網(TDNN)の優れた音韻認識性能を単語/連続音声認識に拡張するべく、TDNNに基礎を置いた日本語の音韻/音節をスポッティングする技術について述べる。そこで、音韻をスポッティングする方法として2つの方法を比較検討した。その内の1つは、ある音韻グループをスポッティングした後、決定された音韻グループ内の音韻を識別する階層的な決定方法であり、他の1つは、モジュール構成したサブネットワークをすべて統合して全音韻を一括してスポッティングする方法である。また、ある一つの音節とそれ以外の音節とを識別できるTDNNを構築した。これにより、全ての音節スポッティング用のTDNNを用意しておけば、原理的に任意の音節スポッティングが可能となる。音韻と音節スポッティング実験の結果、階層的な音韻スポッティング法では91.9%、全音韻の一括スポッティング法では90.8%、音節スポッティング用TDNNは96.7%、の極めて優れたスポッティング性能を得た。これらのスポッティング技術は、連続音声認識へのステップとして有望である。

ATR 自動翻訳電話研究所
ATR Interpreting Telephony Research Laboratories

目　次

1. まえがき

近年、ニューラルネットワークを音声情報処理の研究に利用することが盛んに行われ、特に音声認識や音声合成への応用が活発である。

我々は、これまでに時間遅れ神経回路網(Time-Delay Neural Network: TDNN)を提案し、日本語の有声破裂音"BDG"の音韻認識において非常に高い性能を示すことを報告した[1、2、3]。また、各音韻グループに対応するサブネットワークをモジュール構成して、日本語の18子音にスケールアップする方法[4]や、5母音を含めた全音韻23種にもスケールアップする方法[5]を提案し、各サブネットワークの高い識別率を低下させることなく、カテゴリ数を拡張することが可能であることを示した。

本論文では、これらの成果を連続音声認識へ拡張するために、TDNNを用いた日本語の音韻スポッティングと音節スポッティング方法について報告する。入力音声中の音韻スポッティングや音節スポッティングが精度良く達成できれば、単語音声認識や連続音声認識を行う上で非常に有効である。

まず、2つの方法で音韻のスポッティングを行うことを提案する。一つは、ある音韻グループと、それ以外の音韻グループとを識別するニューラルネットにより音韻グループをスポッティングした後、音韻グループ内の音韻を識別するニューラルネットにより識別する方法である。音韻グループとしては、"B, D, G", "P,T,K", "M,N,sN (sNは撥音)", "S,Sh,H,Z", "Ch,Ts", "R,W,Y", "A,I,U,E,O" の7グループに分割した。

他の一つの方法は、モジュール構成したサブネットワークを統合したTDNN[5]を用いて、全部の音韻を一括してスポッティングする方法である。

また、音韻の代わりに音節をスポッティングすることも併わせて検討する。従来、音節のスポッティング実験としては、英語の単音節(Demi-syllable)をスポッティングしたものがある[6]が、音節数が7音節と少ない。全ての音節をスポッティングすることは、英語の場合、音節数は数千以上といわれており、現実的ではない。しかし、日本語中には、音節の種類は100程度しかない。まず、ある音節とそれ以外の音節とを識別できる音節スポッティング用のTDNNを構成する。このようなニューラルネットワークが学習でき、精度良く音節をスポッティングできれば、音節数だけのネットワークを用意することにより、原理的に全ての音節をスポッティングすることができる。しかし、ある音節と、それ以外の音節の学習データを如何に選択し、ニューラルネットワークに学習させるかは、一つの大きな問題である。

本論文では、この音節スポッティング方法を、先の二つの音韻スポッティング法とも含めて比較検討する。そして、これらの方法を単語音声に適用して連続音声認識への拡張の可能性を示す。

2. 階層的な音韻スポッティング

2.1. 階層的な音韻スポッティング用TDNNの構成

本節では、音韻ポッティングを階層的に行う方法について述べる。この方法の概略を図1に示す。日本語の23種の音韻は7つの音韻グループ "BDG", "PTK", "MNsN", "SShHZ", "ChTs", "RWY", "AIUEO" に分けられる。入力音声中をTDNNでスキャンして、まず音韻グループ(例:"BDG"グループ)のスポッティングを行う。次に、スポッティングされた音韻グループに属する音韻(例: "B"、"D"、 "G" のいずれか)の識別を、各サブネットワーク(例:BDGネット)を用いて行う。図2に、この音韻グループスポッティング用のTDNNの構造を示す。ニューラルネットの入力層と隠れ層1は、各音韻サブカテゴリー(例:BDGネットワーク[1])を識別するTDNNと同様である。但し、"AIUEO"のグループのみは、隠れ層1を12ユニットとした。これは、"BDG"のカテゴリ分けに比べ、音韻グループの方がバリエーションが大きく、大きなキャパシテイを必要とするからである。隠れ層2は2ユニットで構成され、出力層の2ユニットと対応している。図2には、入力データとして、単語中から切り出した音韻"B"を入力した場合を示し、正の値を黒い四角で、負の値を灰色の四角で示している。

図1のネットワーク全体の出力ユニット数は37個(23音韻と、各7音韻グループ以外の7ユニットに対応)、総ユニット数は2,119個、コネクション数は88,275である。

2.2. 実験条件とデータベース

音韻スポッティング用TDNNの学習用と評価用の音声データベースとしては、日本語の重要語5,240単語と音韻バランス216単語[10]を用いた。これらは、256ポイントのハミング窓を掛けてFFTを行った後、特徴パラメータとして10ms毎に16次元のメルスケールのFFT出力に変換した。入力層の値は、15フレーム分で平均値0.0、最大値1.0、最小値-1.0に正規化されている。この内、学習用としては偶数番目の2,620単語を用いた。学習用の音韻データは、学習用単語の全ての音韻環境から、23音韻を音韻ラベルに従ってセグメンテーションして抽出した。そして、各音韻グループ毎に音韻を混合し、学習データとした。ただし学習効率の点から、各音韻グループと特にコンフュージョンを起こし易い音韻グ

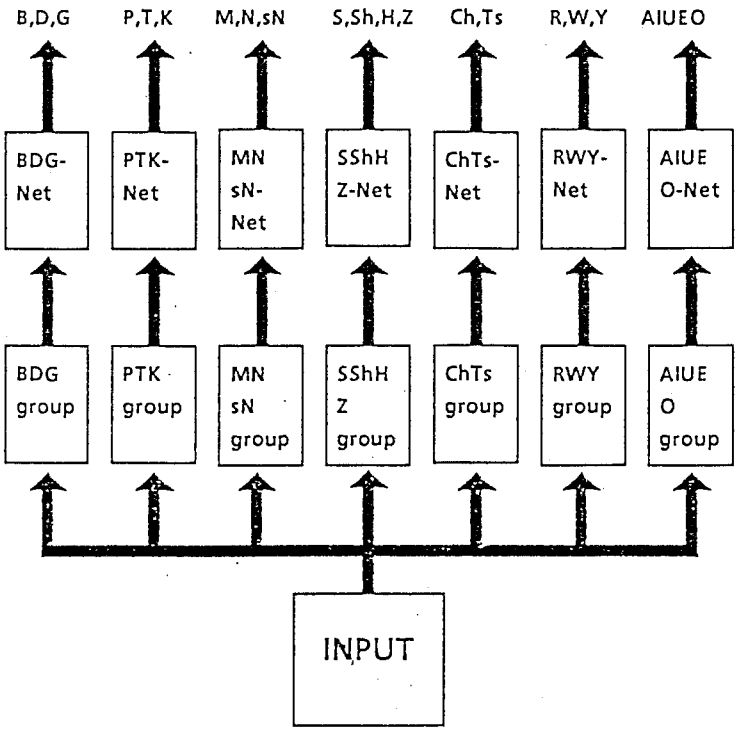

図1　音韻スポッティング用TDNNの階層的な構成

Fig. 1. A Hierarchecal structure of TDNNs for spotting phonemes

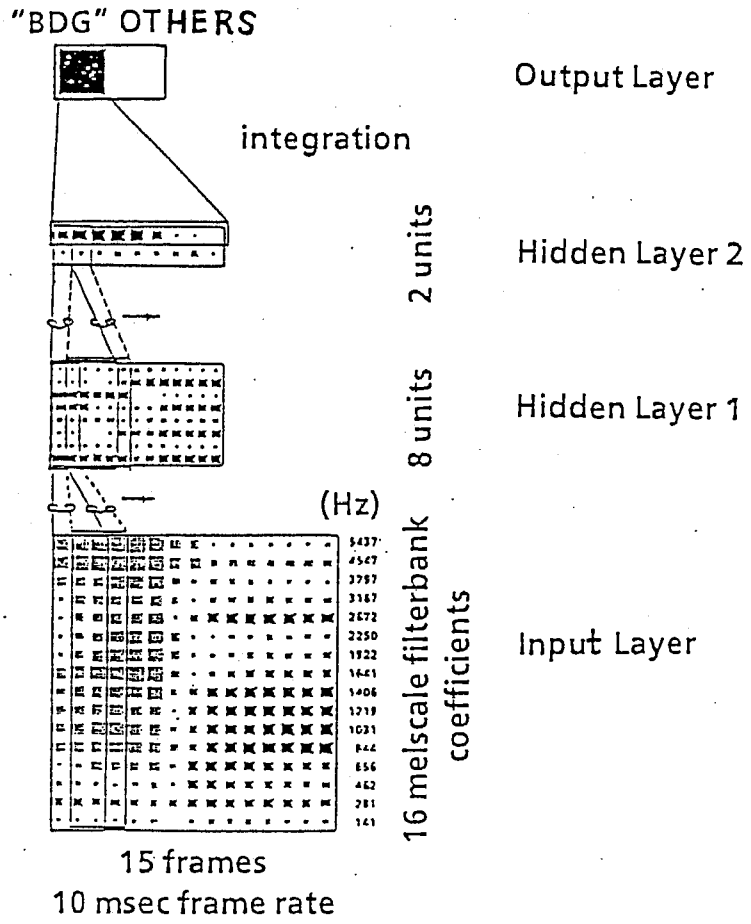

図2　音韻グループスポッティング用TDNN
Fig.2. TDNN for spotting a phoneme group

ループのみを各音韻グループ以外の学習データとして選択した。

評価用の単語としては、音韻バランス216単語の内の50単語を用いた。

2.3. 学習方法

従来、ニューラルネットワークの学習法としては、種々の手法が提案されてきた[7、8]。図2のTDNNの学習には、バックプロパゲーションの高速化アルゴリズム[9]を用いて行った。これは、ネットワークの出力値と教師信号との誤差を重み係数を更新することにより、逐次的に減少させていく方法であるが、高速化の手法として、急峻な誤差空間を用い、重み係数を頻繁に更新したり、ステップサイズやモーメンタムをオーバーシュートしないようにできる限りスケールアップして高速化を図っている。これにより、従来Alliantスーパーミニコンピュータの CPU時間で4日を要した(図2とほぼ同じ規模の)"BDG"ネットワークの学習[1]が1分以内で終了でき、飛躍的に学習時間を短縮できた[9]。図2のTDNNの学習も数分以内で終了した。音韻グループネットワークの学習データは、コンフュージョンを起こし易い各音韻グループカテゴリ毎に最大600個まで用いた。同様に、各音韻識別ネットワークの学習データも、各音韻カテゴリ毎に最大600個まで用いた。図1のネットワーク全体の学習時間は、合計で1.5時間程度である。

2.4. 階層的な音韻スポッティング実験

図1のネットワークを用い、階層的な音韻スポッティング実験を行った。図3に、スポッティング結果の一例を示す。入力単語は男性話者の発声した「TORIATSUKAU」という単語である。下段がネットワークの入力層のスペクトログラム、中段が音韻グループネットワークの出力層、上段が音韻識別ネットワークの出力層である。音韻識別ネットワークの出力値は、各音韻カテゴリの出力値に、各音韻が対応する音韻グループネットワークの出力値を掛けた値である。

中段の音韻グループスポッティングの結果では、"BDG","SShHZ","RWY" グループの挿入誤りが生じているが、大部分の音韻グループは正しくスポッティングされている。また、上段の音韻識別の結果では、階層的な構成のため、音韻グループネットワークの挿入誤りによるスポッティング誤りが生じているが、"Ts"の後の無声化母音"U"を除いて、全ての音韻が正しくスポッティングされている。

表1に、50単語中の音韻のスポッティング結果を示す。評価基準として

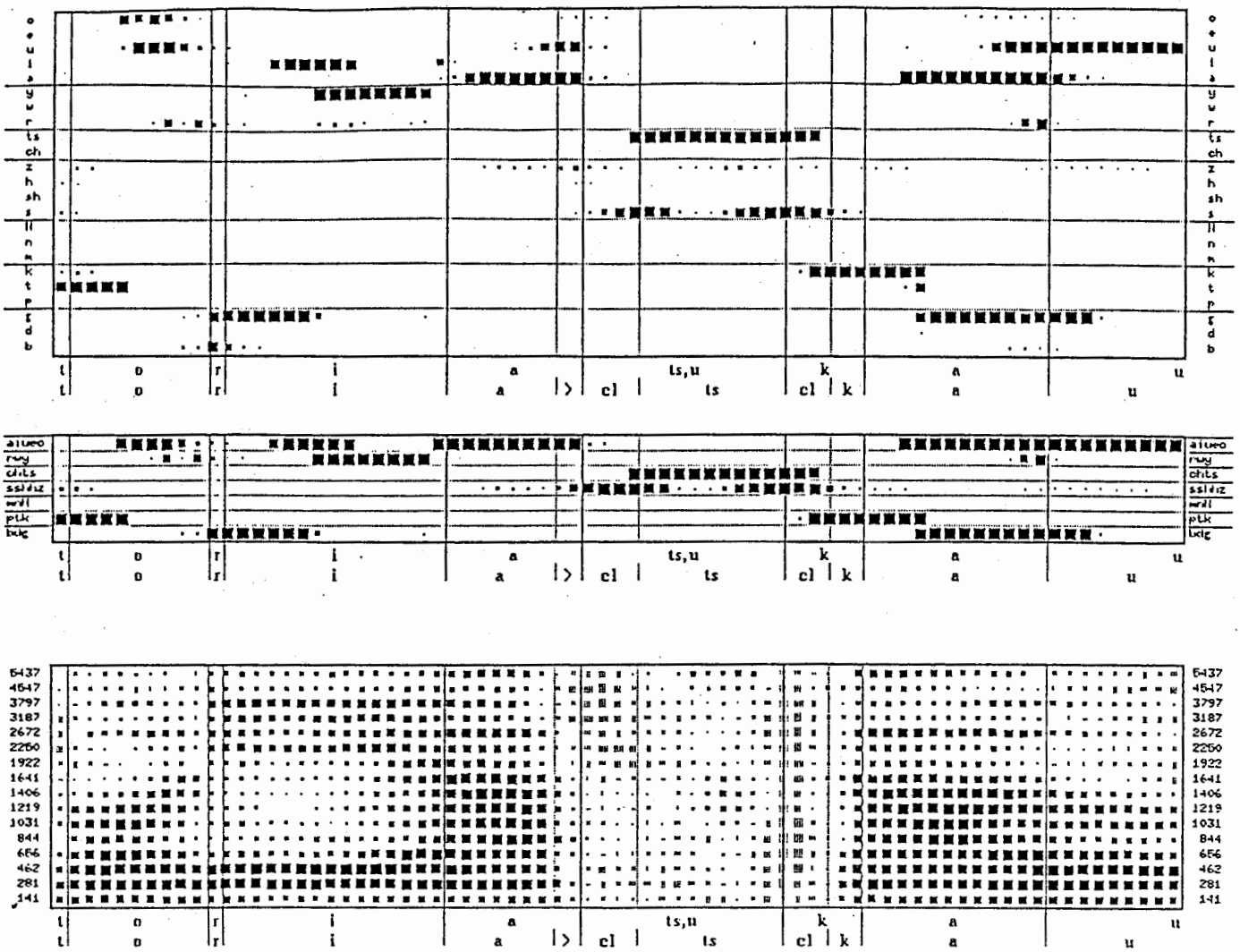

図3　階層的なTDNNによる音韻スポッティング
例：入力音声は“TORIATSUKAU”

Fig.3. An example of phoneme spotting by a hierarchical TDNN : input utterance is “TORIATSUKAU”.

は、音韻ラベルの付与されている区間の内、約半分以上の区間で正しく出力されていれば正解とした。各ネットワークは、音韻境界での挿入誤りが多いが、脱落誤りは少ない。挿入誤りの傾向としては、破裂前の閉鎖部における"H"や"Z"の挿入、語頭や過渡部分の"G"や"K"などの挿入が多かった。正解の音韻グループのスポッティング率は95.1%、さらに(音韻識別による)音韻スポッティング率は91.9%という、良好な結果を得た。

3. 全音韻を統合した音韻スポッティング

本節では、全音韻を統合したTDNNにより、一括して音韻をスポッティングする方法について述べる。
まず、音韻のサブネットワークの構成と学習方法、スケーリングアップの方法[4、5]について簡単に述べ、次に統合ネットワークを用いた音韻スポッティング実験について述べる。

3.1.全音韻一括スポッティング用TDNNの構成

全音韻スポッティング用のTDNNを図4に示す。6つの子音グループ内識別用のサブネットワーク"BDG", "PTK", "MNsN", "SShHZ", "ChTs", "RWY"と子音グループ間を識別するネットワーク"C-class"に、母音グループ内識別用サブネットワーク"AIUEO"をモジュールで構成し、これら全てのサブネットワークを23ユニットを持つ隠れ層3で統合した構成になっている[5]。各サブネットワークを別々に学習した後に、一つのネットワークに統合した。統合ネットワークの総ユニット数は1,628、コネクション数は79,281である。

3.2.実験条件とデータベース

実験条件は、2.2節と同様である。TDNNの学習用音韻データは、重要語の偶数番目の単語の全ての音韻環境から抽出して作成した。そして、(各サブ)ネットワークの音韻カテゴリ毎に混合し、学習に用いた。評価用の単語として、2.2節と同じ音韻バランス単語からの50語を用いた。

3.3.学習方法

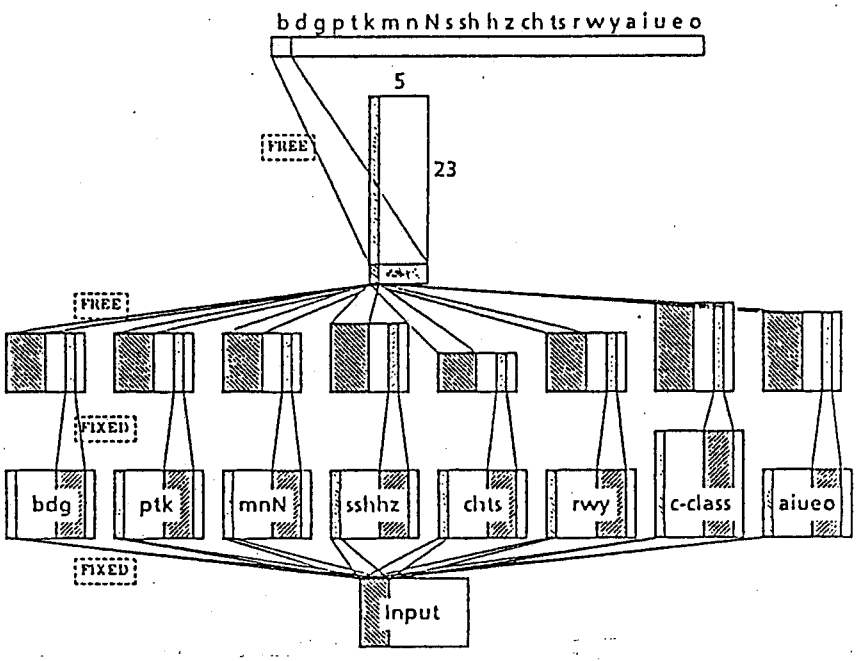

図4　全音韻一括スポッティング用 TDNN
Fig.4. Integrated Modular TDNN
for spotting all phonemes.

図4の統合ネットワークの学習は、まず8つの各サブネットワークの学習を行い、学習された入力層と隠れ層1、及び、隠れ層1と2の間の重み係数を図4のネットワークにコピーした。

次に、これらの重み係数を固定し、隠れ層2と3、隠れ層3と出力層の重み係数をランダムにして、高速なバックプロパゲーション法[9]により学習を行った。さらに、全ての重みをフリーにして重みの微調整を行った。各サブネットワークに用いた学習サンプル数は、カテゴリ当り最大600個、全音韻統合ネットワークの学習には、カテゴリ当り最大200個(全体で4,600個)を用いた。学習は階層的なTDNNと同じく、CPU時間で約1.5時間で終了した。音韻認識実験の結果、重要語の奇数番目の単語中から切り出した評価用の音韻データについて、94.7%の識別率を得た[5]。

3.4.全音韻一括スポッティング実験

図4の全音韻統合ネットワークを入力音声中を1フレームずつシフトしながらスキャンして、音韻のスポッティング実験を行った。図5に、スポッティング結果の一例を示す。入力単語は、図3の階層的なスポッティング法と同じ「TORIATSUKAU」である。下段がネットワークの入力層、上段が出力層を表わす。図5では、"O"の置換誤りと無声化母音"U"の脱落が生じているが、その他の音韻は全て活性化している。挿入誤りは、"T"と"K"の前の閉鎖部で若干生じている。"T"から"A"への渡りの"Y"は自然な発火パターンであり、"R"の発火位置がずれているのも、TDNNの入力層の幅が150msと"R"の継続時間に比べてずっと長いためである。図3の階層的な結果と比べると、挿入誤りは少ない。音韻の発火パターンの継続時間は、図3に比べ、図5の方が音韻間の側抑制(lateral inhibition)がよく効いているため相対的に短くなっている。

表1に、評価用50単語のスポッティング結果を付け加えておく。正解音韻のスポッティング率は90.8%、脱落誤りは9.2%、挿入誤りは102.0%となり、挿入誤りが目立つ。今後、無音区間や境界データの追加学習等により挿入誤りを減少させることが必要となる。

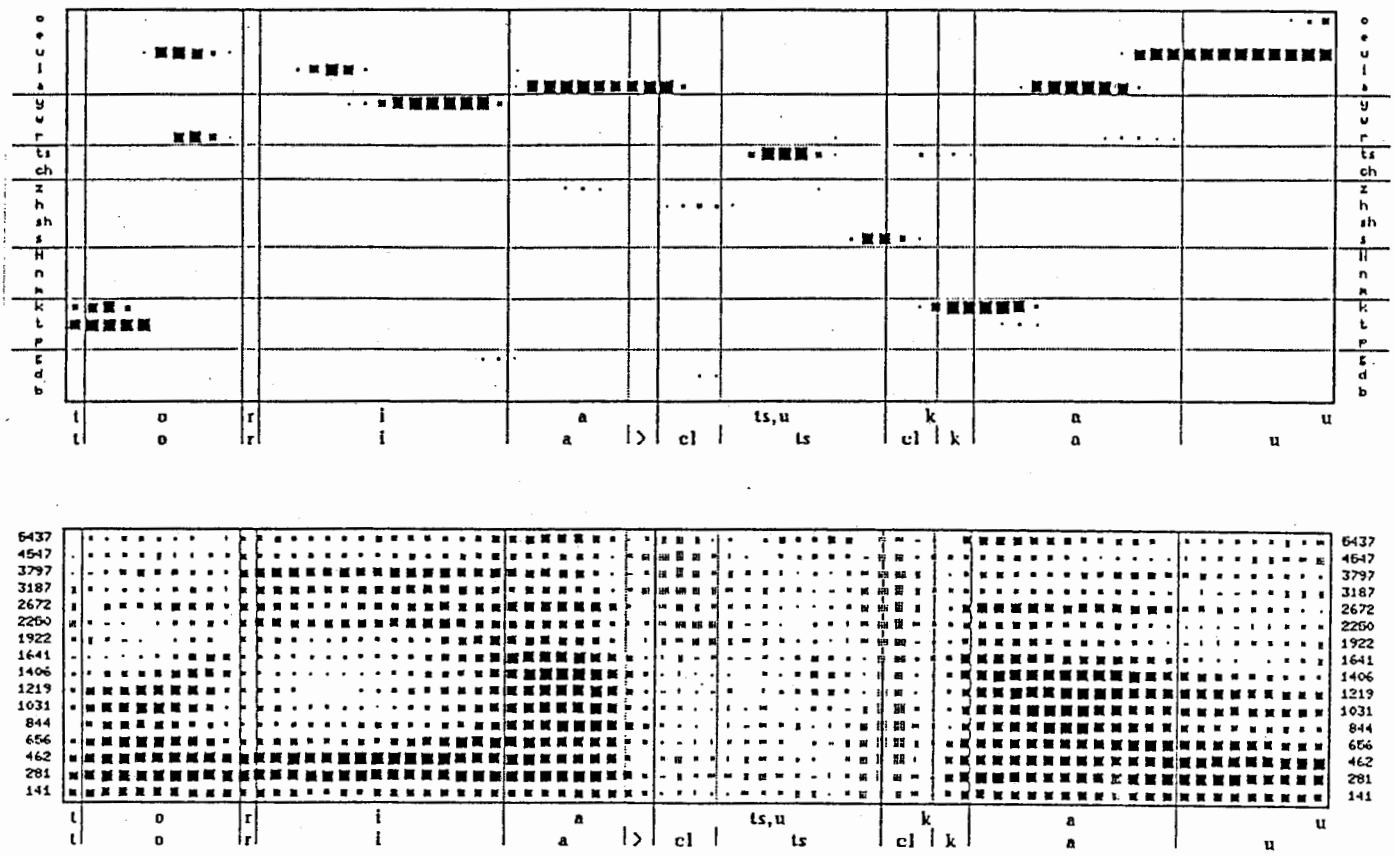

図5　全音韻一括法による音韻スポッティング例：
入力音声は“TORIATSUKAU”

Fig.5. An example of phoneme spotting by a large TDNN : input utterance is “TORIATSUKAU”.

表 1. 音韻スポッティング結果

	階層的スポッティング法		一括スポッティング法
	音韻グループ	音韻	音韻
正解音韻数	330/347(95.1%)	319/347(91.9%)	315/347(90.8%)
脱落誤り数	17/347(4.9%)	28/347(8.1%)	32/347(9.2%)
挿入誤り数	279/347(80.4%)	438/347(126.2%)	354/347(102.0%)

4.TDNNによる音節スポッティングの試み

4.1.音節スポッティング用TDNNの構成

2、3節で述べた音韻スポッティング法との比較のために、音節スポッティング法についても検討した。音節スポッティング用のTDNNの構成は、図2の音韻グループスポッティング用TDNNと同様である。隠れ層1のユニット数は、4ユニットで構成されている。これは、識別すべきカテゴリが、ある音節とそれ以外と少ないためである。ユニット数は313個、コネクション数は2,946である。100音節に対しては、ユニット数7,441個、コネクション数は、約30万必要となる。最初の音節スポッティングの試みとして、"BA"を取り上げた。

4·2·実験条件とデータベース

実験条件は、音韻スポッテイング用TDNNの場合と同様である。学習用の音節としては、学習用の単語中から"BA"を含む53語を抽出し、子音"B"と母音"A"の境界を中心にして、"BA"の部分15フレームを切り出した。"BA"以外の学習用の音節データとしては、約100音節が考えられるが、これらの音節全てを用いることは学習の効率上好ましくないので、"BA"とコンヒュージョンを起こし易いと考えられる5つの音節、"DA","GA","PA","TA","KA"を用いた。これは、バックプロパゲーション学習[7]の誤差空間における識別超平面が、図6に示すように"BA"を囲む形で、"DA","GA","PA",TA","KA"によって形成されることが期待できるからである。

4.3.学習方法

学習は、2.3節の音韻スポッテイング用のTDNNと同様に、バックプロパゲーションの高速化アルゴリズム[9]を用いて行った。学習音節数は全部で1,014音節である。学習時間は、CPU時間で数分以内で終了した。

4.4.音節スポッティング実験

音節スポッティング用TDNNを、入力音声中を1フレームずつシフトしながら実験を行った。評価用の単語としては、先の重要語中の奇数番目の単語か

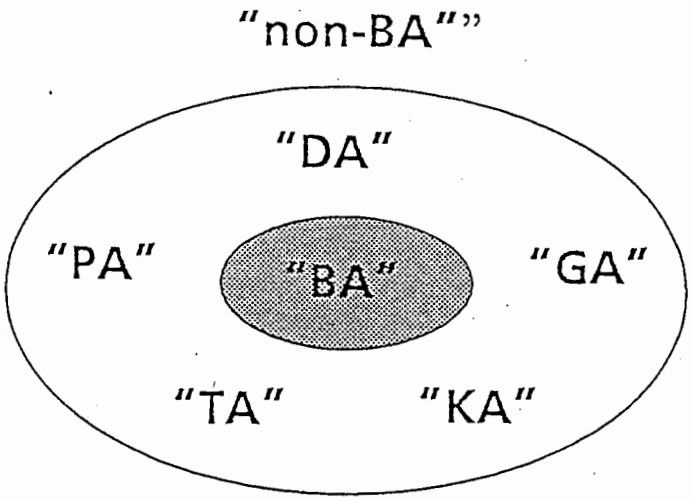

図 6 "BA"とコンフュージョンし易い学習用音節

Fig.6. Training syllable tokens confusable with "BA" in Japanese

ら"BA"を含む61単語を抽出した。これらの単語中には、"BA"が各単語中に1個ずつ、"BA"以外の音節として、学習用に用いた"DA","GA","PA","TA",KA"以外の音節も含めて全部で138音節を含む。未知入力音声の部分が"BA"か"BA"でないかは、次の判定条件に従った。

(判定条件)

O("BA")＞O("non-BA")なら"BA"であると判定。
O("non-BA")＞O("BA")なら"BA"でないと判定。

ここで、O("BA")とO("non-BA")は、各々"BA"と"BA"以外の出力ユニットの値 0≦O("BA"),O("non-BA")≦1である。

表2に、音節スポッティングの実験結果を示す。ここで、"BA"の音節スポッティング率を、音節"BA"の存在する位置で正しくスポッテイングされた割合と定義した。また、"non-BA"の音節スポッティング率は、"BA"以外の全ての位置で正しく抑圧された割合である。音節スポッティング用TDNNは、"BA"を96.7%の割合で正しくスポッティングできた。また、"BA"以外の音節("DA","GA","PA","TA","KA"以外の音節を含む)を、音節境界も含めて99.7%の確率で正しく抑圧することができた。

図7に、スポッティングの実験結果の一例として、出力ユニットの値O("BA")とO("non-BA")を示す。入力単語は、男性話者が"SUBARASHII"と発声したものである。それぞれの出力値から"BA"が音節の中心位置で強く発火しており、"BA"以外の音節が音節境界も含めて良く抑圧されていることが判る。これは、TDNNの時間に対するシフトインバリアントな構造[1]が、音節位置に対する側抑制機能を発揮しているためであると考えられる。

スポッテイング誤りは、"ASHIBA"の語頭の"A"を"BA"と判定した場合と、"KEIBATSU"と"SHIBARAKU"の語中の"BA"が、強く発火しなかった場合の3例のみであった。

5. 検討および考察

表3に、音韻と音節スポッティング用TDNNの比較を示す。2つの音韻スポッティング用TDNNの構成では、階層的方法に比べ全音韻一括法のユニット数は約23%、コネクション数は約10%程度少ない。学習に要する時間は全体でいずれも1.5時間と大差はないが、ネットワークの構成は一括法の方がより複雑である。階層的なネットワークでは、選択的にあるカテゴリを追加学習することが容易である。音節スポッティング用TDNNは、スポッティング精度は良い

表2　音節スポッティング結果(例：“BA”)

音節	“BA”	“non-BA”	Total
音節数	61	138	199
正解数/合計	59/61	137/138	196/199
識別率(%)	96.7	99.3	98.5

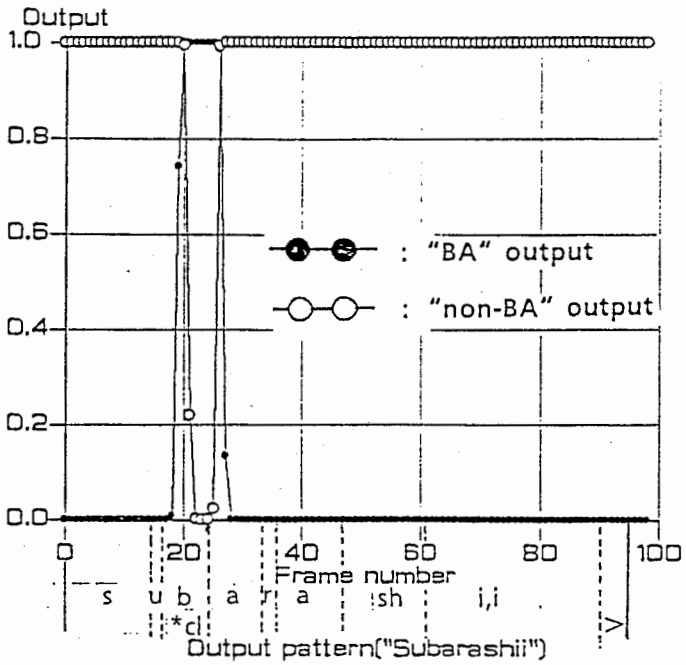

図 7　音節スポッティング結果の例：
入力音声は "SUBARASHII"

Fig.7. An example of CV-syllable spotting results : input utterance is "SUBARASHII".

表3. 音韻スポッティング用TDNNと音節スポッティング用TDNNとの比較

	音韻スポッティング用		音節スポッティング用	
	階層的ネット	全音韻一括ネット	1音節用	全音節用
入力ユニット数	241	241	241	241
出力ユニット数	37	23	2	200
全ユニット数	2,119	1,628	313	7,441
コネクション数	88,275	79,281	2,946	294,600
学習時間	1.5時間	1.5時間	数分以内	約5時間

が、ネットワーク全体の規模が階層的ネットに比べ約3.5倍とかなり大きくなる。2つの音韻スポッティング法において、スポッティング正解率に大差はないが、全音韻一括法の約91%に比べ、約92%とやや階層的な方法が勝っている。脱落誤りも、階層的音韻スポッティング法の方が約1%少ない。挿入誤りは、階層的な方法の方が多い。これは、ある音韻グループを取り囲む識別(超)平面が、音韻の中心位置でのコンヒュージョンを起こし易い音韻グループだけでは充分に形成されなかったためと考えられる。また階層的な構成上、各音韻グループ間の側抑制(lateral inhibition)機能が働き難い。

脱落や挿入誤りの傾向としては、破裂前の閉鎖部における"H"や"Z"の挿入、語頭や過渡部分の"G"や"K"などの挿入、"S"と"Ts"、"Sh"と"Ch"の置換などが共通して見られた。特に、全音韻一括スポッティング法では、母音や半母音相互の置換や挿入誤りが多かった。これは、音韻間の側抑制は機能するが、誤って正解音韻が抑圧される場合もあるためである。

音節のスポッティング正解率は96.7%であり、中心位置で切り出されたデータのみでも、よく側抑制機能が働いていることが判る。これは、"BA"以外の5種類の音節だけでも識別平面が適切に形成されている傍証になる。

スポッティング誤りは、無音部や境界データを含む誤って出力した学習データをニューラルネットに追加学習させることにより、かなりの部分を改善できると考えられる。したがって、これらの音韻/音節スポッティング結果は、今後文節発声や連続音声の認識を目指す上で大いに有望な結果であるといえる。

6. むすび

本論文では、時間遅れ神経回路網(TDNN)を用いて連続音声認識を目指すために、日本語の音韻スポッティング法と音節スポッティング法を提案し、第一段階として、これらの方法を単語音声に適用して連続音声認識への適用の可能性を示した。

音韻スポッティング法では、階層的な方法と、全音韻を一括してスポッティングする方法とを提案し、これらの性能を比較検討した。その結果、91%以上の確率で音韻のスポッティングが可能であることが判った。また、音韻の挿入や脱落誤りについては、今後追加学習により、かなり改善される見通しを得た。

また音節スポッティングの試みでは、ある音節とそれ以外の音節とを識別してスポッティングすることが可能なTDNNを構成し、ある音節(例:"BA")を96.7%の確率でスポッティング可能なことを示した。

これらの良好なスポッティング結果は、TDNNの時間方向に対するシフト・インバリアントな性質を顕著に示すものであり、TDNNを用いた連続音声

認識を目指す上で非常に有効な結果である。今後は、連続音声認識を目指し、言語処理との統合についても検討していく。

謝辞

日頃御指導頂く樽松社長と、御討論頂いた音声情報処理研究室の皆様に感謝します。

参考文献

(1)アレックス・ワイベル:"時間遅れ神経回路網(TDNN)による音韻認識",電子情報通信学会技術研究報告SP87-100,pp19-24,(1987-12).
(2)A.Waibel, T.Hanazawa, G.Hinton, K.Shikano and K.Lang: "Phoneme Recognition Using Time-Delay Neural Networks", ATR Technical Report TR-I-0006 (Oct.1987).
(3)A.Waibel, T.Hanazawa, G.Hinton, K.Shikano and K.Lang: "Phoneme Recognition: Neural Networks vs. Hidden Markov Models", IEEE, International Conference on Acoustics, Speech and Signal Processing, pp107-110 (Apr.1988).
(4)A.Waibel, H.Sawai and K.Shikano: "Modularity and Scaling in Large Phonemic Neural Networks", ATR Technical Report TR-I-0034 (Aug.1988).
(5)沢井秀文、アレックス・ワイベル、宮武正典、鹿野清宏:"モジュール構成ニューラルネットワークのスケールアップによる音韻認識"、電子情報通信学会技術研究報告SP88-105,pp73-80,(1988-12).
(6)C.Kamm, T.Landauer and S.Singhal: "Using an Adaptive Network to Recognize Demisyllables in Continuous Speech", J. Acoust. Soc. Amer., vol.83, suppl. 1. X7, May 1988.
(7)D.E.Rumelhart and J.L.McClelland:"Parallel Dictributed Processing; Explorations in the Microstructure of Cognition", vol I and II, MIT Press, Cambridge, MA, 1986.
(8)R.P.Lipmann: "An Introduction to Computing with Neural Nets", IEEE ASSP Magazine, 4-22, Apr. 1987.
(9)P.Haffner, A.Waibel, H.Sawai and K.Shikano : "Fast Back-Propagation Learning Methods for Neural Networks in Speech", ATR Technical Report TR-I-0058 (Nov.1988).
(10)武田一哉、勾坂芳典、片桐滋、桑原尚夫: "研究用日本語音声データベースの構築"音響学会誌、pp747-754、(1988.10).