

## 単語の接続情報を利用した単純再帰結合型ネットワークの拡張

篠沢 佳久

慶応義塾大学 理工学部 管理工学科

本研究においては単純再帰結合型ネットワーク（エルマンネット）を拡張した語系列予測モデルを提案する。エルマンは言語の獲得過程のモデルとして単純再帰結合型ネットワークを提案した。これは文章を単語単位に区切り、現時点の単語のみを提示し、次単語を予測する学習（語系列予測課題学習）をエルマンネットで行なうことによって文法特徴を獲得できる可能性を示した。エルマンネットによる文法や語彙の獲得の学習は文法規則に制約を設けた人工文を対象とし、特に含まれる単語数についてはその個数も限定されている。そこで本研究においてはエルマンネットを拡張し、新たな単語を追加学習する際、局所的な構造の変更および学習で済む語系列予測モデルを提案する。これは単一のエルマンネットの機能を複数のネットワークを用いて分散処理を行ない、ネットワークの入出力を単語の接続情報（バイグラム）を用いて制限した上で、ネットワークの内部構造を局所結合化することによって蓄積される単語の履歴情報と文脈情報を区分し、新たな単語が表出するたびにネットワークを逐次増やしながら学習することによって実現を試みる。

### Extended simple recurrent networks by using bigram

Yoshihisa SHINOZAWA

Department of Administration Engineering,  
Faculty of Science and Technology, Keio University

We propose a model of word sequence prediction by improving simple recurrent networks. Elman proposed simple recurrent network which is a model of language acquisition. Elman showed that SRN learns to predict the next word of the sentences and can acquire grammatical concepts and meanings. We think that it is difficult for SRN to learn the sentences which contain a number of words, especially to learn new words. We improve SRN and propose a model of word sequence prediction which learns new words additionally. We propose how to learn to predict the next words by distributed networks. Our model adds new network as new word appears. The structure of our model is decided by using bigram and a network has local connections to learn word and grammatical feature.

#### 1. はじめに

現在、人間の言語の理解および文法や語彙の獲得過程についてのモデルを構築する研究がさまざまな分野で行なわれている。計算機科学の分

野においては、エルマンの提案するニューラルネットワーク（以下ネットワークと略す）を用いた言語情報の獲得モデルがその一つとして挙げられる。エルマンは単純再帰結合型ネットワーク（以下エルマンネットと呼ぶ）を用いて、言語の獲得過程の

モデルを提案した[1][2]。これは英語の文章を単語単位に区切り、2章で説明する文脈層への再帰結合を持ったネットワーク(エルマンネット)に文頭から一単語ごとに入力し、次単語を予測する学習(語系列予測課題学習)を文末の単語が現れるまで繰り返し行なった。その結果、学習後のネットワーク内の中間層の活性パターンにおいて、統語範疇(名詞や動詞といったカテゴリー)や意味範疇(人や物といったカテゴリー)を表す単語クラスターが形成されていた。すなわち現時点の単語のみを提示し、次単語を予測する学習を繰り返し行なうことによって、ネットワーク内部に文法的な特徴を獲得できることを示したのである。

これまでエルマンネットを基礎として文法[3][4]や語彙[5][6]についての言語獲得モデルが提案されてきた。特に関係節を含む複雑な文章の構造[2]や who を文頭にした疑問文[3]、さらには前置詞を含む副詞句[4]などの文法特徴の獲得が次単語予測学習を通して可能なことが示されている。しかしエルマンネットによる文法や語彙の獲得の学習は特定の文法規則をもとに生成した人工文を対象としている。特に人工文中に含まれる単語数についてはその個数も限定されており、また未知単語が表出した際、新たな単語の追加学習も容易ではない。

そこで本研究においては単語数の増加に対応が容易な語系列予測モデルを提案する。これはエルマンネットを下記の三点から拡張することによって実現を試みる。

- ① 単一のネットワークの機能を複数個のネットワークを用いて分散処理を行なうことによって単語数の増加に対応できるようにする。
- ② ネットワークの入出力を単語の接続情報(バイグラム)を利用し、その構造を制限することによってネットワークの大規模化を抑制する。
- ③ 蓄積される単語の履歴情報と文脈情報を区分しながら学習するため、ネットワークの内部構造の局所結合化を図る。

そして提案モデルを用いて新たな単語が表出する度にネットワークを逐次増やしながらか次単語予測学習を行なう方法を提案する。

## 2. エルマンネットの構造と学習方法

エルマンネットの構造は多層パーセプトロン型を基本とし、入力には外部からの入力値を処理する入力層と直前の中間層の活性状態がフィードバックされる文脈層の二つから構成される(図1)。

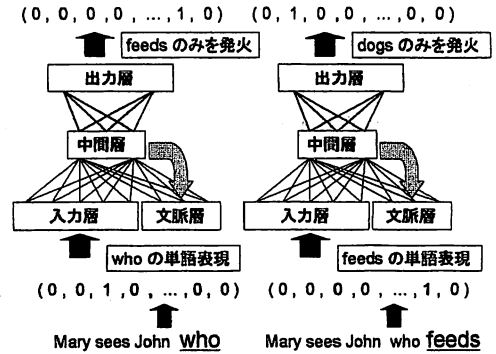


図1 エルマンネットによる次単語予測学習

文脈層は直前の中間層の状態を保持することから、その時点までのネットワークの状態(過去にされた情報)を記憶できるという特徴をエルマンネットは持つことになる。エルマンネットを用いて次単語予測学習を行なう場合、入力層および出力層には学習用の例文中に表出する単語数分のニューロンをあらかじめ用意する。そして任意の単語に対して入力層と出力層の特定のニューロンを対応させる。

学習時には、英語文を単語ごとに区切り、文頭の単語から入力層に順次入力していく。その際、現時点に入力する単語と対応する入力層のニューロンには1を、それ以外は0というように1もしくは0のベクトル(単語表現)で入力する。一方、文脈層については中間層と同じニューロン数を用意しておき、直前の中間層の値(内部表現)をそのままフィードバックして入力する。すなわち入力層には現時点の単語情報を入力し、文脈層にはこれまでの単語の履歴情報を再入力することによって上部に位置する中間層には現時点までにした文章の履歴情報が蓄積することになる。

そして出力層は次単語と対応するニューロンのみ発火(1の値を出力)し、それ以外は発火しない(0を出力)ように教師信号を与え、ネットワーク内の結合係数の修正を行なう。これを文末の単語が現れるまで繰り返し行なう。

従ってネットワークの構造上、入力層および出力層においては、総単語数分のニューロンが必要であり単語数が増加した場合、学習が困難であったり、新たな単語を追加学習することは容易ではないものと予想される。すなわちエルマンネットを語彙獲得モデルとして拡張していく上で以下の問題が生じるものと考えた。

- ① 単語数が多くなった場合、ネットワークの規模

が大きくなり単一のネットワークでは学習が困難なことが予想される。

- ② 新たな単語を学習する場合、あらかじめ入力層および出力層にその単語と対応したニューロンを用意しておくか、もしくは新規にニューロンを増やさなければならない。特に後者の場合、新規にニューロンを追加することによって全ての単語表現が変わってしまうためこれまでの学習結果に影響が生じてしまう。

そこで本研究においては単語数の増加に対応を容易とするため、単語を追加学習する際、局所的な構造の変更および学習で済むようにエルマンネットを拡張したネットワークの構造および学習方法を提案する。

### 3. 提案モデルの構造

#### 3.1 入力層制限なしモデル

単語数の増加によるネットワークの大規模化を抑制するため、単一のエルマンネットを用いて次単語予測を行なうのではなく、複数個のネットワークを用いて機能の分散処理を図ることを試みる。

まず出力層についてはニューロンを1個と制限し、特定の単語と対応づけを行なう(図2, 入力層制限なしモデルと呼ぶ)。すなわち特定のネットワークは対応づけされた一単語のみの予測を行なう。従って次単語予測学習においてN個の単語を対象とする場合、各単語を処理するためにN個のネットワークが必要となる。入力層についてはエルマンネットと同様に全単語を入力対象とする。文脈層の構造については3.3節および3.4節にて述べる。

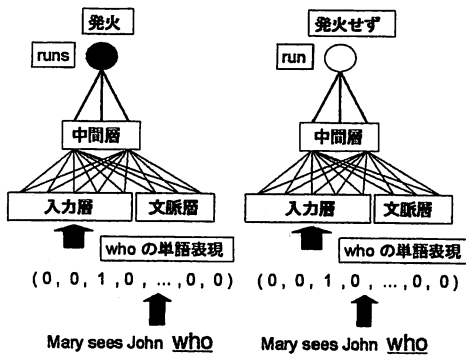


図2 入力層制限なしモデル

そして次単語予測学習を行なう場合、現時点の単語をN個のネットワークに同一の単語表現で同時に入力し、次単語と対応するネットワークの出力層のみを発火させ、それ以外は発火させないように学習する。このように出力層のニューロンを一つとすることによって単語を新たに追加学習する場合、その単語を処理するネットワークを新たに生成するだけで済む。しかし全てのネットワークの入力層において新規単語を対象としたニューロンを一個ずつ増やさなければならず、全ての単語の表現形式が変わってしまう。その結果、問題②で述べたように全てのネットワークにおいて、これまでの学習結果に影響が生じてしまう。

そこでさらに入力層についても、入力対象とする単語情報を制約することによってこの問題の解決を試みる。

#### 3.2 Bigram モデル

次に入力層については単語の接続情報(バイグラム)を利用し、ニューロン数を制限することを試みる。学習用の例文中に表出する単語の接続情報より単語iの直前に表出する単語集合 $S_i$ を求める。そして単語iに対応したネットワークの入力層は単語集合 $S_i$ のみを入力対象とする(図3, Bigramモデルと呼ぶ)。従って同じ単語を入力してもネットワークごとで入力層の対象となる単語は異なるため、その単語表現はそれぞれ異なる。

次単語予測学習を行なう場合、現時点の単語をN個のネットワークに入力する時、ネットワークごとで単語表現に変換し入力する。そして入力層制限なしモデルと同様に次単語に対応した出力層を持つネットワークのみを発火させ、その他のネットワークは発火させないように学習を行なう。

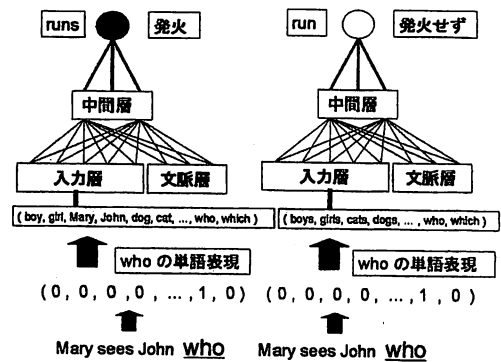


図3 Bigram モデル

このような構造をとることにより新たな単語の追加学習を行なう場合、入力層制限なしモデルと同様にその単語を対象としたネットワークを新たに生成すればよい。さらに全てのネットワークの入力層においてニューロンを新たに追加する必要はなく、新たな単語と接続する単語を処理するネットワークのみに影響が生じるだけで済むものと考えた。

### 3.3 文脈層へのフィードバック

エルマンネットは直前の中間層の状態を文脈層にフィードバックさせることによって過去に入力した情報を保持することが可能である。本研究で提案するネットワークにおいて、中間層から文脈層へのフィードバックについては次の二種類を考案した(図4)。

#### ① 自己フィードバック

通常のエルマンネットと同様に各ネットワークにおいて、それぞれ直前の中間層の活性状態をフィードバックさせる。入力層制限なしモデルの場合、各ネットワークの入力層は全ての単語を対象としているので過去に入力した単語の情報は文脈層を通して蓄積されていく。しかし Bigram モデルの場合、各ネットワークの入力層は全ての単語を対象としていないため、自己フィードバックのみでは過去に入力した情報が欠落してしまう。

#### ② 文脈フィードバック

Bigram モデルにおいては直前の単語を処理したネットワークの中間層の値を文脈層に渡していくことによって、過去の単語情報が蓄積されていく。そこで直前の単語を出力層に持つネットワークの中間層の活性状態を全てのネットワークの文脈層にフィードバックする。

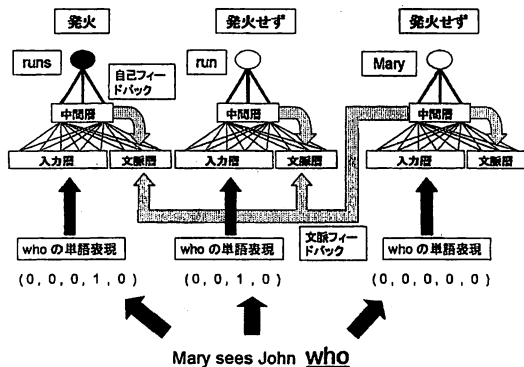


図4 文脈層へのフィードバック

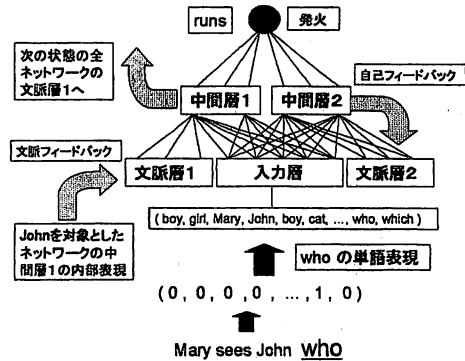


図5 局所結合型モデル

### 3.4 局所結合型モデル

Bigram モデルにおいては、自己フィードバックによって過去の単語の履歴情報が、文脈フィードバックによって文脈情報が保持されていく。しかし図4に示すように同一の中間層と二種類の文脈層を結合させることによって、中間層には二種類の情報が混在することになってしまう。そこで図5に示すように中間層も二つに分割し、自己フィードバックと文脈フィードバックによって再入力する中間層と文脈層をそれぞれ対応づける。

すなわち入力層は全ての中間層と結合しているが、二つの文脈層はそれぞれ対応した中間層とのみ結合した局所結合構造とする(局所結合型モデルと呼ぶ)。このような構造とすることによって中間層1による文脈情報と中間層2による過去の単語の履歴情報とが別々に蓄積され、二種類の情報が混在せず学習できるのではないかと考えた。

### 4. 学習アルゴリズム

局所結合型モデルにおけるネットワークの学習アルゴリズムは下記の通りである。

#### ● 各ネットワークの構造

単語数を  $N$  個含む文章を学習対象とした  $N$  個のネットワークの状態を想定する(初期状態は  $N=0$  から開始する)。単語  $i(i=1,2,\dots,N)$  を出力層の対象としたネットワーク  $i$  において、入力層の対象を単語  $i$  の直前に表出する単語集合  $S_i$  とする。従って入力層の個数は単語集合  $S_i$  の要素数である。また二種類の文脈層と中間層の個数をそれぞれ  $M$  個とする(合計で  $2M$  個)。

- 各ネットワークの学習

- ① 学習に用いるための例文数を  $J$  個とする。入力文章  $L_j = (w_j(0), w_j(1), \dots, w_j(t), \dots, w_j(T))$  (例文数  $j = 1, 2, \dots, J$ ) を単語単位  $w_j(t)$  に区切る。学習の高速化のために文章  $L_j$  を学習する場合、学習対象のネットワークは  $L_j$  に含まれる単語を処理するネットワークのみとする。
- ② 初期状態を  $t=0$  とする。
- ③ 全ネットワークの中間層を初期化する。
- ④ 現時点の単語  $w_j(t)$  と次単語  $w_j(t+1)$  が新たな単語かどうかを調べる。次単語  $w_j(t+1)$  を対象としたネットワークが存在しない場合、単語  $w_j(t+1)$  を対象とするネットワークを一個生成する ( $N=N+1$  とする)。そのネットワークの入力層には単語  $w_j(t)$  を対象としたニューロンを一個生成する。また次単語  $w_j(t+1)$  を対象としたネットワークにおいて入力層に単語  $w_j(t)$  を対象としたニューロンが存在しない場合は、単語  $w_j(t)$  を対象としたニューロンを入力層に追加する。
- ⑤ 単語  $w_j(t)$  を対象とするネットワーク  $k$  ごとに、単語表現  $V_{jk}(t)$  に変換し入力層に入力する。
- ⑥ 文脈フィードバックによって直前の単語  $w_j(t-1)$  に対応したネットワーク  $m$  の中間層 1 の値  $H_{jm}(t-1)$  を全てのネットワークの文脈層 1 にコピーする。
- ⑦ 自己フィードバックによってネットワーク  $k$  ごとに直前の中間層 2 の値  $P_{jk}(t-1)$  を文脈層 2 にコピーする。
- ⑧ 出力層は次単語  $w_j(t+1)$  に対応したネットワークは発火し、それ以外は発火しないように教師信号を与え結合係数の修正を行なう。学習には誤差逆伝播則 (バックプロパゲーションアルゴリズム) を用いる。
- ⑨ 単語  $w_j(t+1)$  が文末でなければ、 $t=t+1$  とし④へ戻る。文末であれば  $j=j+1$  とし②へ戻り、別の例文で学習する。

以上、例文の個数分だけ②から⑨を誤差の自乗和が最小になるまで繰り返す。

## 5. 評価実験

### 5.1 提案モデルによる次単語予測学習

エルマンのシミュレーションで使用された人工文の生成規則[2](図6)を基に50,000文の例文を作成し、次単語予測学習を行なった。

S	→ NP VP .
NP	→ PronN   N   N RC
VP	→ V2   V1 NP
RC	→ whom NP VP   who VP
N	→ boy   girl   cat   dog   boys   girls   cats   dogs
PronN	→ John   Mary
V1	→ chase   feed   see   hear   chases   feeds   sees   hears
V2	→ walk   live   walks   lives

図6 人工文の生成規則

生成規則には図6の他にも feed の後は cat や dog といった動物を対象とする名詞しか表出しないといった規則もある。単語数は文頭を示す'<H>'と文末を示す'.'を含め26個である。文章中に含まれる単語数は平均5.96語であった。

局所結合型モデルにおいてはネットワークは単語数分必要となる。各ネットワークは4層型とし出力層は1個、圧縮層は20個、二種類の中間層と文脈層はそれぞれ64個と固定した(合計で128個)。そして学習後のネットワークを用いて学習用の例文に対して次単語予測を行ない、各単語についてその正解となる順位を調べた。そして例文中の全単語の順位を調べ、その平均値を求めた(予測順位)。しかし学習ごとに生成されるネットワークの挙動は異なるため、予測順位は毎回変動する。そこで学習と予測を5回別々に行ない、5セット分の予測順位の平均値を求めた。そして同様な条件で学習を行なった下記のモデルの結果[7]と比較する。

- 単一のエルマンネット(入力層と出力層は26個、圧縮層は20個、中間層と文脈層は128個)
  - 入力層制限なしモデル(各ネットワークの構造は出力層は1個、圧縮層は20個、中間層は64個、文脈層は128個、入力層は26個)
  - Bigram モデル(入力層制限なしモデルと同じ構造だが、入力層は単語の接続情報より決定)
- これらのモデルにおいても同様に5回学習と予測を別々に行ない、5セット分の予測順位の平均値を求めた。その結果を図7に示す。

図7より、いずれのモデルにおいても平均予測順位は3程度の値となっている。これは次単語予測を行なった際、正解の単語がほぼ第3位に表出していることを意味する。単語数が26個と少ない場合、入力層制限なしモデルとBigramモデルの精度が他と比較して低い。これは入力層制限なしモデルの場合、自己フィードバックのみで学習が可能なのに対し、文脈フィードバックも用いてい

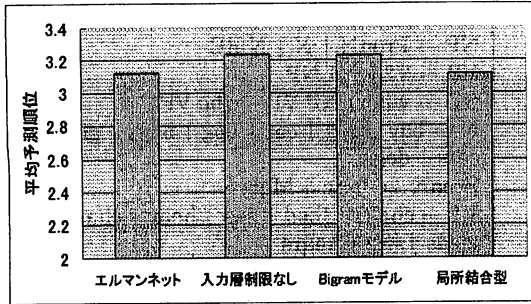


図7 各モデルによる平均予測順位

るため、学習に利用した情報量が多かったためであると考えられる。また入力層制限なしモデルと Bigram モデルを比較してほぼ精度が同じことから、ネットワークの入力情報を単語の接続情報を用いて制限しても問題ないことが分かる。さらに局所結合型モデルと Bigram モデルを比較して、局所結合型モデルの精度が良いことから、ネットワークの内部構造の局所結合化の効果が表れていることが分かる。

## 5.2 単語数を増加した場合

次に局所結合型モデルにおいて、学習対象の人工文中に含まれる単語数を増やし学習を行なった。人工文の生成規則は図 6 のままとし、単語数のみを増やしていった。単語数は 26 個から 202 個まで段階ごとに 22 個ずつ増やしていった(生成規則中の N, PronN, V1, V2 を段階的に 8 個, 2 個, 8 個, 4 個と増やしていった)。5.1 節の実験と同様に生成規則より 50,000 文の人工文を生成し学習対象とした。そして学習後、次単語予測を行ない各単語の予測順位を求めた。5.1 節の実験と同様に学習と予測を 5 回別々に行ない、5 セット分の予測順位の平均値を求めた。

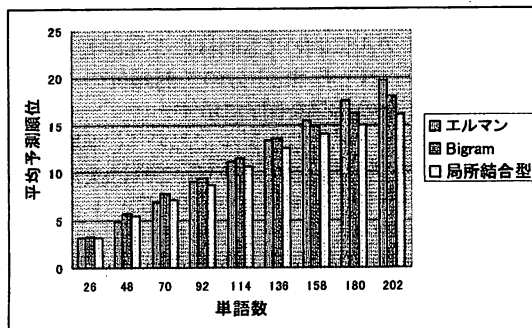


図8 単語数を増加した場合の平均予測順位

通常のエルマンネットおよび Bigram モデルにおいて同様な実験条件で行なった結果[7]と比較する。その結果を図 8 に示す。

図 8 より単語数が少ない場合は単一のエルマンネットの精度が高いものの、単語数が増加していくにつれ Bigram モデルと局所結合型モデルの精度が向上する傾向にあることが分かる。すなわち単一のエルマンネットで処理するよりも、複数個のネットワークを用いて分散処理した方が精度の向上が図れること、文脈フィードバックによって過去の文脈情報を伝達していくことで入力情報を単語の接続情報を用いて制限できること、ネットワークの内部構造の局所結合化を図り、過去の単語情報と文脈情報を区分し、学習することによってさらに精度の向上を図れることが確認できた。

## 6. まとめ

本研究においてはエルマンネットを拡張し、単語数の増加に対応が容易な語系列予測モデルを提案した。そして提案モデルにおいて次単語予測学習を通じた実験の結果、単語数の増加にともない予測精度に向上がみられることが確認できた。今後は提案モデルにおいて中間層の活性状態を分析し、文法特徴の獲得についての解析をすすめていく予定である。

### 参考文献

- [1] Elman, J. L., "Finding Structure in Time", *Cognitive Science*, 14, pp.179-211(1990)
- [2] Elman, J. L., "Distributed representations, simple recurrent networks, and grammatical structure", *Machine Learning*, 7, pp.195-224(1991)
- [3] 玉森彩弥香, 乾敏郎, "Elman ネットによる統語範疇の配列と格関係の学習", *認知科学*, Vol.6, No.3, pp.359-368(1999)
- [4] 上原謙吾, 原田賢一, "後続要素の予測を利用する系列予測モデル", 第4回情報科学技術フォーラム(FIT2005)論文集, H-014(2005)
- [5] 森藤大地, 乾敏郎, "自己組織化マップを含む Elman 型ネットワークを用いた文法知識の獲得モデル", *認知心理学会第1回大会*(2003)
- [6] 下斗米貴之, 遠山修治, 大森隆司, "文法メタ知識による語彙学習加速のコネクショニストモデル", *認知科学*, Vol.10, No.1, pp.104-111(2003)
- [7] 篠沢佳久, 上原謙吾, "単語の接続情報を利用した語系列予測モデル", *信学技報*, NC2006-8(2006)