モジュール型コネクショニスト日本語複文解析システム
― 学習方式の検討 ―

本木 実　嶋津好生（九州産業大学）
{motoki,shimazu}@te.kyusan-u.ac.jp

1. はじめに

我々はこれまで，モジューラ統合ニューラルネットワークによる，日本語複文の格構造解析を行うシステムJSPACを提案してきた．昨年までは，JSPACの概念設計を計算機実験として実現するまでに至った[1]．しかしながら，JSPACはモジューラアプローチによるため，それらを学習させる方式に様々なヴァリエーションが考えられる．本稿では，2つの学習方式を取り上げ，各学習方式の学習過程と実行性能について調べた結果を報告する．

2. ネットワーク構成

本モデルのネットワーク構成を図1に示す．このモデルは単文パーサ，スタック，セグメンタと呼ぶ3つのモジュールで構成される．

実行モードにおいて，全モジュールでのネットワークを形成し，1つのモジュールの出力が他のモジュールの入力になる．そして，修飾－被修飾関係にある複文を入力し，節ごとの格構造表現を出力する．

本モデルの詳細は文献[1]を参考にされたい．

本モデルでは単語の表現に分散表現を採用している．つまり，単語を実数ベクトルで表している．以下このベクトルを**単語ベクトル**と呼ぶ．本稿では，単語ベクトルを，各要素が[0.0,1.0]の8次元の実数ベクトルとし，ランダムで固定とした．

3. 学習方式

本モデルの学習方式は，同時に同期をとりながら学習する方式と，各モジュール個別に学習する方式とが考えられる．さらに，詳細に分けると，主に次の4つが考えられる．

(1) 全てのモジュールを1つとして，学習する方式
学習中，各モジュール間の情報は全く復元させずに，全モジュールで出力される値をそのまま他のモジュールに入力する．セグメンタの学習が収束していない時点では，各モジュール間の情報伝達が適切でないまま学習を続けることになる．

(2) 制御信号のみ教師信号を用いて復元する方式
学習中，セグメンタの制御信号を教師信号を用いて復元する．つまり，各モジュール間のゲートの開閉は（学習途中の誤差を含んだ値ではなく）正しい値を用いる．単語ベクトルは学習時に出力される値をそのまま他のモジュールに入力する．(1)に比べて，モジュール間情報伝達のタイミングは安定するので，その間，単語ベクトルも正答信号に近い値が他のモジュールに入力されることになる．

(3) 制御信号と単語ベクトルの両方を復元する方式
学習中，セグメンタの制御信号と単語ベクトルを用いて復元し，各モジュール間のゲートの開閉とモジュールに入力する単語ベクトルに正しい値を用いる．(1),(2)に比べて，モジュール間情報伝達のタイミングと単語ベクトルは安定する．

(4) 各モジュールを別々に学習させる方式
各モジュールをそれぞれ独立して学習させ，各モジュールの誤差がある程度収束した時点で，全モジュールを結合する．(3)と類似しているが，モジュールの同期はとっていないので，収束するまでの学習回数がそれぞれのモジュールで異なる．

今回は，(2)の方式と(3)の方式について実験を行った．以下では，(2)の方式を**モジュール同期学習方式**，(3)の方式を**モジュール個別学習方式**と呼ぶことにする．

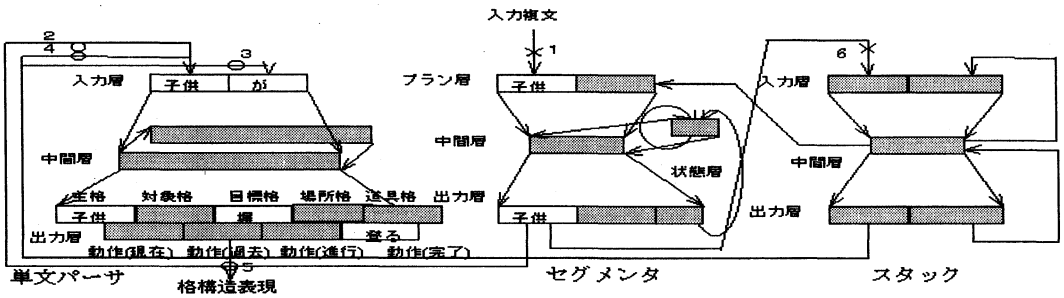

図 1: JSPACのネットワーク構成

4. 実験

4.1 データ

計算機実験に用いた単語は表2に示す計30単語である．これらの単語を使用し，学習，テストのための複文データを計算機を用いて自動生成した．生成した文に対し，日本人である筆者の言語感覚で日本語として正しいものだけ選別し，さらにそれらに助動詞を付加してより自然になるように作成した．作成した文は，単文156文，修飾節の深さが1の複文735文，修飾節の深さが2の複文1041文の計1932文であった．今回の実験では，主節が現在時制の文のほか，過去時制，現在進行時制の文についても作成した．

作成した文をパターン別に整理すると表1のようになる．

4.2 実験条件

学習に用いたデータは修飾節の深さ1の複文735文のうちの約70%である514文である．単文と修飾節の深さ2の複文は学習には用いず，解析テストに用いた．

各モジュールのネットワーク条件は以下のとおりである．全モジュールとも中間層数は1層とした．単文パーサの中間層ユニット数，文脈層ユニット数は80とした．セグメンタの中間層ユニット数は100とした．スタックの中間層ユニット数は60とした．すなわち，セグメンタの入力層は単語ベクトルの8ユニットとあわせて68ユニットで構成される．また，スタックの入力層，出力層も単語ベクトルの8ユニットとあわせて68ユニットで構成されることになる．全モジュールとも，シグモイドの傾きは1.0とした．

表 1: 作成文の文パターン

No.	文数	文パターン（N：名詞 V：動詞 P：助詞 AUX：助動詞）
1	40	N P N P V 。
2	80	N P N P V AUX 。
3	12	N P N P N P V 。
4	24	N P N P N P V AUX 。
5	79	(N P V AUX) N P N P V 。
6	158	(N P V AUX) N P N P V AUX 。
7	2	(N P N P V AUX) N P N P V 。
8	4	(N P N P V AUX) N P N P V AUX 。
9	75	N P (N P V AUX) N P V 。
10	150	N P (N P V AUX) N P V AUX 。
11	2	N P (N P N P V AUX) N P V 。
12	4	N P (N P N P V AUX) N P V AUX 。
13	81	(N P V AUX) N P (N P V AUX) N P V 。
14	162	(N P V AUX) N P (N P V AUX) N P V AUX 。
15	3	(N P V AUX) N P (N P N P V AUX) N P V 。
16	6	(N P V AUX) N P (N P N P V AUX) N P V AUX 。
17	3	(N P N P V AUX) N P (N P V AUX) N P V 。
18	6	(N P N P V AUX) N P (N P V AUX) N P V AUX 。
19	80	((N P V AUX) N P V AUX) N P N P V 。
20	160	((N P V AUX) N P V AUX) N P N P V AUX 。
21	64	N P ((N P V AUX) N P V AUX) N P V 。
22	128	N P ((N P V AUX) N P V AUX) N P V AUX 。
23	64	((N P V AUX) N P V AUX) N P (N P V AUX) N P V 。
24	128	((N P V AUX) N P V AUX) N P (N P V AUX) N P V AUX 。
25	63	(N P V AUX) N P ((N P V AUX) N P V AUX) N P V 。
26	126	(N P V AUX) N P ((N P V AUX) N P V AUX) N P V AUX 。
27	76	((N P V AUX) N P V AUX) N P ((N P V AUX) N P V AUX) N P V 。
28	152	((N P V AUX) N P V AUX) N P ((N P V AUX) N P V AUX) N P V AUX 。

表 2: 使用単語

名詞 (13)	妻 子供 犬 猫 雪 雨 雪割草 木 塀 庭 魚 花壇 犬小屋
動詞 (9)	登る 入る 走る 遊ぶ 降る 植える 吠える 叱る くわえる
助詞 (4)	が を で に
助動詞 (2)	た ている
その他 (2)	。 NULL

表 3: 文構造別解析結果

文構造	文数	正解率 (%)	
		同期	個別
深さ 1 の複文 (学習文)	514	99.0	78.4
単文	156	73.7	64.1
深さ 1 の複文 (未学習文)	221	97.3	74.2
深さ 2 の複文	1041	92.2	56.3

学習アルゴリズムは純粋な誤差逆伝搬法 [3] を用いた．学習方法は，逐次学習法を用い，モジュールの実行動作が前方向に 1 回行われるたびごとに，誤差を逆方向に 1 回伝搬した．学習は 1000 回で停止させた．学習条件は以下のとおりである．全モジュールとも学習係数 η は 0.1 で慣性項はなしとした．セグメンタの残存率 γ は 0.5 とした．結合荷重と各ユニットのバイアスの初期値は $[-1.0, +1.0]$ でランダムに設定した．

ネットワークの動作条件は以下のようにした．単文パーサの出力層での単語の同定はユークリッド距離を用い，入力単語の中で最も近い距離の単語を出力単語とした．セグメンタの出力層におけるコントロールユニットが 0.5 より大きければ 1 の制御信号，0.5 以下であれば 0 の制御信号とみなした．また，解析テストでは，単文パーサからの格構造表現が，全て正しいタイミングで，全ての格スロットに対して正しい単語が同定された場合を正解とした．このような条件で，学習 1000 回のうち，5 回ごとに解析テストを行なった．

4.3 結果

学習終了後，単文，修飾節の深さが 1 の複文，修飾節の深さが 2 の複文に対して，解析テストを行った．モジュール同期学習方式，モジュール個別学習方式の結果をそれぞれ，図 2〜図 5 に示す．これらより，モジュール同期学習方式の方がモジュール個別学習方式よりも高い正解率を示していることがわかる．また，単文に関しては，修飾節の深さが 1 の複文と，深さが 2 の複文よりも正解率が低くなっている．

モジュール同期学習方式（同期）とモジュール個別学習方式（個別）とについて，格解析正解率が最高時の結果を表 3 にまとめた．学習に用いた修飾節の深さ 1 の複文（約 70%）を学習させ，未学習の修飾節の深さ 1 の複文（残り約 30%）を高い正解率で格解析できていることがわかる．また，修飾節の深さ 1 の複文のみ学習させ，修飾節の深さ 2 の複文に対して格解析を行った場合も最高 92.2%の正解率を示した．これらのことから，JSPAC は通常の一般化のみならず，文の構造上の一般化を高い正解率で実現していることがわかる．

5. 考察

モジュール同期学習方式とモジュール個別学習方式を比べた場合，モジュール同期学習方式の正解率が高かった．当初の予想は逆であった．モジュール個別学習方式の方が，入力される単語ベクトルが安定するので，学習がスムーズに行われ，その結果，正解率も高いと考えたのだ．しかし実際には，両学習方式における学習速度はそれほど変わらなかった．また，モジュール同期学習方式は，実際の実行過程に即した学習方式なので，格解析の正解率も高かったと考えられる．

このモジュール同期学習方式の学習がうまくいった理由は，よくわからない．1 つの可能性として，モジュール間の情報伝達が，学習に有利な方向に働いたのではないか，と考えられる．例えば，セグメンタの自己相関学習では，プラン層の右側のスロットパターンに近くなるよう，出力層の右側のスロットパターンが，結合荷重を更新することで調整される．しかし逆に，出力層の右側のスロットパターンに，プラン層の右側のスロットパターンが近くなるとすれば，どうであろうか．その場合も学習が進むことになる．プラン層の右側のスロットパターンはすなわち，スタックの中間層パターンのコピーである．このスタックの中間層パターンが偶然にも，出力層の右側のスロットパターンに近くなったというのであろうか．このことは，セグメンタだけでなく，同様にスタックについてもいえる．以上のことは，学習中に各スロット間のパターンの距離を求めることで確かめられる．

しかしながら，モジュール個別学習方式の方が，単語ベクトルが学習中に変化しないことを考えると，学習が安定するはずである．実際は，両学習方式の学習の速度はそれほど変わらず，モジュール個別学習方式の方が，学習速度が格段に速いともいえない．ひとつの理由としては，モジュール個別学習方式の中間層ユニットが最適でない可能性があるということである．中間層ユニット数を，色々に変えてみて両学習方式で実験してみる必要がある．

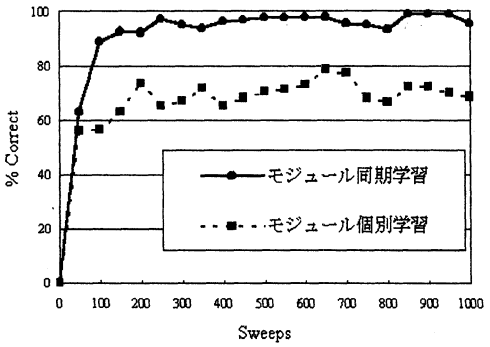

図 2: 深さ 1 の複文（学習文）に対する正解率

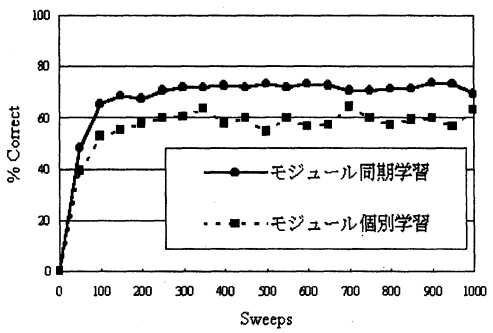

図 3: 単文（未学習文）に対する正解率

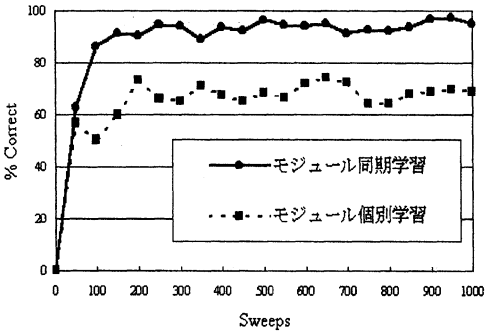

図 4: 深さ 1 の複文（未学習文）に対する正解率

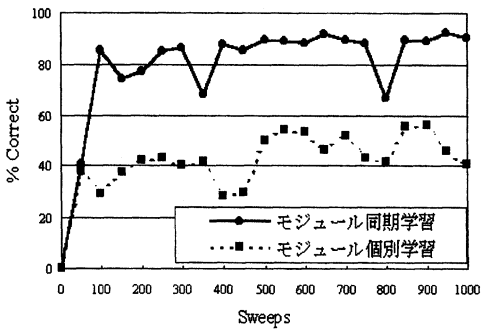

図 5: 深さ 2 の複文（未学習文）に対する正解率

今回，モジュール同期学習方式において，文の構造上の一般化能力を 92.2%という高い精度で実現することができた．これは，文献 [1] の 79.2%に比べると格段に高い結果となっている．その理由として，今回用いた学習文とテスト文の文パターン種別数は 28 であり，前稿の文パターン種別数 45 と比べて少なく，比較的容易なタスクであったことが考えられる．しかし，本稿の文は主文の時制に過去，現在進行，現在完了を入れるなど，前稿のより自然な文となっている．今後，必須格がない文，例えば「雪割草を（花壇に）植えている子供に庭を走っている犬が吠えている。」などの文の解析を試み，よりオープンなテストを行う必要がある．

6. おわりに

コネクショニスト日本語複文解析システムにおいて，修飾節の深さが 1 の複文のみを学習させ，修飾節の深さが 2 の複文を 92.2%の正解率で格解析が可能であった．また，モジュール同期学習方式とモジュール個別学習方式とでは，前者が後者より高い正解率を示した．

参考文献

[1] 本木実, 嶋津好生: 統合コネクショニストモデルによる日本語複文解析システムの構築, 言語処理学会第 6 回年次大会発表論文集, pp. 111–114 (2000).

[2] Miikkulainen,R.: “ Subsymbolic Parsing of Embedded Structure,” *Computational Architecutures Integrating Neural and Symbolic Processes, edited by R.Sun and L.A.Bookman*, pp.153–186 (1996).

[3] Rumellhart,D.E., Hinton, G.E., and Williams, R.J.: *Learning Internal Representations by Error Propagation*, in Rumelhart,D.E., McClelland,J.L., and the PDP Research Group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* Volume 1, pp. 318–362, The MIT Press, Massachusetts, (1986).