

# Finding Structure in Time

Elman, J. L. (1990)  
Cognitive Science 14, 179-211

## 1 はじめに

- 認知システムにおいて時間は重要
  - ◆ 時間の認識無しに目的志向的な行動は取れない
    - 発声・運動
    - プランニング
    - 因果関係の認識
  
- 時間とニューラルネット
  - ◆ 従来、ニューラルネットで時系列データを取り扱うことは困難とされてきた
  - ◆ 「ニューラルネットにはプロセスがない」などの批判
  
- 本研究の目的
  - ◆ 時間的要因を処理に組み入れたアーキテクチャの提案
  - ◆ 自然言語処理の分野で検証

## 2 時間の表現に関する問題

- 従来 of 時間表現 (Cottellm Munro, & Zipser, 1987; Elman & Zipser, 1988)
  - ◆ 明示的・人工的なもの
  - ◆ 空間的メタファーに基づく
  - ◆ ビットベクトルの利用  
011100000 → 時間 1 に 0, 時間 2 に 1, 時間 3 に 1...
  
- 従来 of 時間表現における問題
  - ◆ 問題1: 世界とネットとのインタフェースが必要
    - ◇ 入力を蓄えるバッファを用意
    - ◇ ある程度の時間が経過した後に、バッファ中のベクトルをネットへ入力
    - ◇ 更新を決定する基準はどうなっているのか
  - ◆ 問題2: 時間枠の設定が固定
    - ◇ ベクトル長を変えることはできない
    - ◇ 多様な長さのデータ(e.g., 単語)を扱えない
  - ◆ 問題3: 絶対的な時間から相対的な時間を切り出せない

- ◇ 相対的には同じだが、絶対的には異なるパターン  
011100000  
000111000
- ◇ ベクトル空間上では全く異なるポイントに

### 3 提案アーキテクチャ

- 非明示的な時間の表現
  - ◆ 処理に付随する形で時間を認識
  - ◆ ネットワークに時系列を扱う動的特性を付与
  - ◆ ネットワークに「記憶」を持たせる
  
- Jordan (1986) の方法 (Figure 1)
  - ◆ 出力層の状態が逐次 state ユニットへコピー
  - ◆ 現在の入力直前の出力とブレンドされた後に、中間層へ
  - ◆ 「前回の行動内容を元に、次の行動のプランを立てる」ことに相当
  
- 提案手法 (Figure 2)
  - ◆ 中間ユニット(システムの内部状態)を逐次文脈ユニットへコピー
  - ◆ 入力信号と文脈ユニットのブレンドされた信号が中間層に
  
- 提案手法の使い方
  1. 事前準備
    - 時間順に並べられたデータ系列の用意
    - 文脈ユニットの初期値を 0.5 に設定
  2. 時間 t
    - 系列の一番初めのデータを入力層に
    - 入力ユニットと文脈ユニットが中間ユニットを活性化
    - 中間ユニットから出力ユニットへフィードフォワード
    - 中間ユニットから文脈ユニットへフィードバック
  3. バックプロパゲーションによる学習 (Rumelhart, Hinton, Williams, 1986)
    - 出力ユニットの状態と教師信号を比較
    - 出力ユニットの状態が教師信号に近づくようリンク強度を更新
    - 中間→文脈のリンク強度は 1.0 に固定
  4. 時刻 t+1
    - 上記 2・3 の繰り返し
    - 文脈ユニットに前回の中間ユニットの状態が保持

- 提案手法の意味
  - ◆ 通常の3層ネットの場合、中間層は刺激の内部表現に相当
  - ◆ 中間層が、直前の中間層の状態を受け取るということは、内部表現に時間的情報が非明示的に埋め込まれることを意味する

#### 4 実験 1: 排他的論理和演算 (XOR)

- XOR 課題
  - ◆ 与えられた2つの命題のいずれか1つのみが真であるときに真と判断する論理演算
  - ◆ 00,11,01,10→0,0,1,1
  
- 時間的 XOR 課題
  - ◆ 1ビットずつの提示
  - ◆ 例 101000011110101...
    - 1番目と2番目の入力によって3番目の入力値が決定
    - 4番目と5番目の入力によって6番目の入力値が決定
  
- 入力データ系列の用意
  - ◆ 3000ビットの系列
    - ◇ 系列1と系列2をランダムに設定
    - ◇ 系列3を系列1と系列2の値に基づき決定
    - ◇ (1000回繰り返し)
  
- ネットワークの構成
  - ◆ 入力ユニット数 1
  - ◆ 中間ユニット数 2
  - ◆ 出力ユニット数 1
  - ◆ 文脈ユニット数 2
  
- トレーニング
  - ◆ 各時点で与えられた入力を利用して、次に与えられる入力を出力
    - input 101000011110101...
    - output 01000011110101?...
  - ◆ 教師信号付
  - ◆ 3000ビットの600回トライアル
  
- 結果 (学習後のエラー率 - Figure2)
  - ◆ 系列位置 2・5・8・11→エラー小

◆ 系列位置 1・3・4・6・7・9・10・12→エラー大

■ 考察

◆ 入力データの時系列的な構造を反映した結果

◇ 1番目・2番目の値は予測できない

◇ 3番目の値は予測できる

→ネットワークが時間に含まれる構造を学習した

## 5 実験 2: 文字系列の学習

■ 目的

◆ XORよりも複雑な時系列パターンを学習することができるか検討

◆ 文字列の構造を発見する課題

■ 入力データ系列

◆ 音声情報を模した6ビットのベクトル (Table 1)

◆ 子音 (b,d,g) をランダムに結合 (1000個の系列)

◆ 各子音の後ろに母音を結合

結合ルール: b→ba, d→dii, g→guuu

例: bdgbddg→diibaguubadiidiiguuu

■ ネットワーク

◆ 入力ユニット6

◆ 中間ユニット20

◆ 出力ユニット6

◆ 文脈ユニット20

■ トレーニング

◆ 教師信号付の予測課題(次の文字を予測)

◆ 入力データ系列を200回連続提示

■ テストフェーズ

◆ トレーニングフェーズとは異なるランダム系列を用意

◆ 系列位置に応じたエラー率を算出

■ 結果1(全体的なエラー率-Figure4)

◆ 母音でのエラーが小さく子音では大きい

◇ 入力系列の構造を反映

- ◇ 子音が提示されたら, 次の母音が分かる
- ◇ 強調するべきところは, 母音が何回続くかもネットワークは予測していること

## ■ 結果 2(ユニットごとのエラー率-Figure 5)

### ◆ ユニットの区別

- ◇ ユニット 1(子音を識別するユニット)→Figure5a
- ◇ ユニット 4(高音を識別するユニット)→Figure 5b

### ◆ ユニット 1 はユニット 4 よりもエラー率が低い

- ネットワークは次に来る音の子音か母音かを正確に予測
- ◇ どの子音が来るかはわからないものの, 確実に子音が来ると予測
- ◇ 入力系列の構造を反映していない高音ユニットはエラー率が高くなる
- 逆に言えば, ネットワークは子音と母音を区別するカテゴリを形成した

## ■ 考察

### ◆ XOR 課題よりも複雑な系列でテスト

- ◇ 系列が長い, ビット長が長い, 遅延のバリエーションが豊富

### ◆ ネットワークは容易に入力系列の構造を学習

- ◇ 入力の複雑さは学習の困難さとは関係がない
- ◇ 重要なことは, 入力データがよく構造化されていること

## 6 実験 3: 単語の切り出し(形態素解析)

### ■ 目的

- ◆ 一般的な信念: 言語は単語から成り立つ
- ◆ 言語学における議論: 単語と合成語, フレーズの区別は曖昧  
エスキモーでは単語が英語のフレーズレベル
- ◆ 人間は形態素の切り出しをどのように行っているのだろうか

### ■ 入力データ系列

#### ◆ 現実の単語からなる文字系列を用意

- ◇ 15 種類の単語を用意
- ◇ 200 センテンス(4~9 語)をランダムに生成
  - 1270 単語・4963 文字
  - 入力は 5 ビットのベクトル系列 (Table 2)
  - 単語間のスペースなし(英語のリスニングを想定)

### ■ ネットワーク

#### ◆ 入力ユニット 5

- ◆ 中間ユニット 20
- ◆ 出力ユニット 5
- ◆ 文脈ユニット 20
  
- トレーニング
  - ◆ 系列を 10 回入力
  
- 結果 (Figure 6)
  - ◆ 単語の開始位置で高いエラー率
    - ◇ 単語の終端に向けて徐々にエラーが低下
  - ◆ 単語とフレーズの曖昧さを反映した結果
    - ◇ They は The+y として切り出される傾向
    - ◇ A boy を一単語として切り出す傾向
  
- 考察
  - ◆ 単語学習の原理は、文字間の共起性(単語内に含まれる文字間の共起頻度は高い)
    - ◇ 言語学習は入力の共起性に基づくという原理と整合
    - ◇ 意味の対応付けも共起性に基づく
  - ◆ 人間と類似した振る舞い
    - ◇ はじめの数文字を聞いた時点で単語の認識が可能
    - ◇ 子供と類似した誤り(子供は熟語を一単語と認識する傾向(MacWhinney, 1978))

## 7 実験 4: 深層構造(文法)の抽出

- 目的
  - ◆ 入力を文字系列から単語系列に拡張
  - ◆ 現実の言語では、単語の系列は文法を制約として出現
    - ◇ 深層構造にしたがって、状況に応じた表層構造が生成される (Chomsky, 1957)
    - ◇ 聞き手は表層構造を手がかりとして深層構造を構成—どうやって?
  
- 入力データ系列
  - ◆ データ
    - ◇ 単語: 13 クラスの名詞・動詞 (Table 3)
    - ◇ テンプレート: 単語間の組み合わせルール (Table 4)
  - ◆ 系列作成の手順
    - ◇ ランダムに 10,000 のテンプレートを組み合わせ
    - ◇ テンプレートに単語をランダムに埋め込み (27,534 単語)
    - ◇ 単語を 31 ビットベクトルに置き換え

- 各ベクトルは互いに直交(重なりなし=文法の情報ベクトルに含めない)
- 入力系列の例(Table 5)

■ ネットワーク

- ◆ 入力ユニット 31
- ◆ 中間ユニット 150
- ◆ 出力ユニット 31
- ◆ 文脈ユニット 150

■ トレーニング

- ◆ 入力データ系列中の次の単語を予測
- ◆ データを6回提示

■ 結果(パフォーマンス)

- ◆ RMS(実際の系列ベクトルと出力ベクトルとのズレ): 0.88
- ◆ RMS(実際の確率ベクトルと出力ベクトルとのズレ): 0.053  
ある単語の次に来る単語の確率を示すベクトル(入力データから計算)  
0, 0.2, 0, 0.05... 合計 1
- ◆ 確率ベクトルと出力ベクトルとの類似度(コサイン): 0.916  
→次に来る単語を予測することはできないが, クラスを予測することができる

■ 結果(学習後の中間ユニット(内的表現)の状態)

- ◆ 手続き
    1. 学習後, 再度系列を入力(ウェイトの更新なし)
    2. 各時点での中間ユニットの状態を保存
    3. 単語ごとに平均ベクトルを算出
    4. クラスタ分析(類似しているベクトルを近くに配置) Figure 7
  - ◆ 結果
    - ◇ 文法に即したクラスタリング
      - 動詞と名詞のクラスタを分離
      - 目的語が必要な動詞
      - 目的語があってもなくてもよい動詞
      - 目的語が不要な動詞
      - 動物ではない名詞
      - 動物である名詞
      - 人間・人間ではない(大きな動物・小さい動物)...
- 入力系列の構造から, 単語のクラスを学習

- 補足実験
  - ◆ 目的
    - ◇ 学習後のネットワークは新規な単語にどのように反応するか
  - ◆ 手続き
    - ◇ 学習したウェイトを固定
    - ◇ Man を Zog に置き換え(新たなビットベクトルを割り当てる), 再度入力
  - ◆ 結果 (Figure 8)
    - ◇ Man が Zog に置き換わった木構造
    - ◇ ネットワークは Zog が出現するときに Man を予測
    - ◇ Zog を Man と同じものとして扱う
  - ◆ 考察
    - ◇ 人間が新規な単語の意味を文脈から推測する行動と一緒に
    - ◇ 文章理解における文脈情報の重要性を示す研究と一貫した結果 (Swinney, 1979; Tabossi, Colombo, & Job, 1987)

## 8 分析: タイプとトークンの区別

- 背景
  - ◆ 記号処理 vs. ニューラルネット
    - ◇ ニューラルネットでは記号(概念)を扱うことができない (Fodor & Pylyshyn, 1988)
  - ◆ 記号処理とニューラルネットにおける内部表現の差異
    - ◇ 記号処理モデルにおける概念の表現  
ラベル(A, x,  $\beta$ )によって表現される変数
    - ◇ PDP における心的表現  
中間層におけるユニットの活性化パターン(分散表現)
  - ◆ ニューラルネットにおける内部表現の弱点
    - ◇ 記号処理におけるラベルはカテゴリを扱う(変数束縛)
    - ◇ PDP におけるパターンは状況依存  
John という個人に対して, 状況に固有のパターンが存在してしまう
- 著者の見解
  - ◆ 弱点を逆手にとれないか
    - ◇ 分散表現と文脈ユニットを組み合わせることで, タイプとトークン(出現形)の区別を扱うことができるのでは
  - ◆ 分散表現の利点
    - ◇ 個々のユニットが連続値をとるのであれば, ユニットのセットによって, 無限の概念を表現することができる



- ◇ 本研究の中間ユニットは文脈ユニットと入力ユニットのブレンディングされたパターン  
→個々の状況に応じた概念を表現

## ■ 分析

- ◆ 実験4のデータを利用
- ◆ 単語のトークンを要素としたクラスタ分析
  - ◇ Figure 7は単語ごとの平均ベクトルを要素としたクラスタ分析
  - ◇ 27454のベクトル(中間ユニットの状態)を要素としたクラスタ分析

## ■ 結果

- ◆ Figure 7のクラスタリングの下に個々のトークンが続く
  - ◇ 個々のトークンをタイプにまとめればFigure 7と全くの同型(例外なし)
  - ◇ タイプ内のトークンはよく類似しているものの、完全に同じベクトルになったものは一つとしてない
- ◆ BoyとGirlに対応した木 (Figure 9)
  - ◇ 二つの木は酷似
  - ◇ 文の終末にきているのか、先端にきているのかでクラスタの区別
    - 終末にきているものは、直前の単語の類似性によってクラスタリング
    - 先端にきているものは、文脈によらないクラスタリング(文と文の結合はランダムであるため)

## ■ 考察

- ◆ 高次元のベクトル空間上で類似した点にトークンがクラスタリング
- ◆ タイプとトークンの区別は空間上の解像度の違い
- ◆ 変数束縛よりも自然な形でタイプとトークンを区別可能

## 9 まとめ

- 本研究の主張: 時間的特性を考慮せずに認知システムを検討することはできない
- 本研究の意義: ニューラルネットによる時系列データの処理を実現したこと
  - ◆ 本研究では、時間を処理に組み込んだ形で取り扱った
  - ◆ 提案したアーキテクチャは単純なものだが、驚くほど強力
- 得られた知見
  - ◆ エラーシグナルは時間構造を検討するうえで有効な手がかりである
  - ◆ 系列の複雑性は学習パフォーマンスを悪化させない
  - ◆ 学習される中間ユニットは高度に課題依存である
  - ◆ 学習の結果、中間ユニットは構造化される

■ 今後の課題

◆ 分析方法の改善

クラスタ分析では時間的特性を捉えられない

ネットワークの状態が時間によってどのように変化するかを捉えないと

◆ 記憶容量の検討

ネットワークによって扱うる記憶の範囲を明らかにしたい