

深層学習

系列データのためのネットワーク

会津大学 コンピュータ理工学研究科 コンピュータ情報システム学専攻 高橋輝

系列データ

個々の要素が順序付きの集まり

$$\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \dots, \mathbf{x}^T$$

として与えられるデータを系列データと呼ぶ

約束

- 要素の並びをインデックス $1, 2, 3, \dots, T$ で表し, t を時刻と呼ぶ.
 - 時刻は物理的な時間と対応するとは限らない
- 個々のデータが順序を持っている, つまり, 並びに意味を持っていればよい.

今回扱う問題

1. テキスト to 多クラス (Text to Multi-class)

- レストランの利用客の感想を3段階で評価.
- 文を構成する各単語をベクトルで表現. (下に例示)
 $\mathbf{x}^1 = \text{'They'}$, $\mathbf{x}^2 = \text{'have'}$, \dots , $\mathbf{x}^{15} = \text{'better'}$
- データの最小単位は一つの文($\mathbf{X}_n = (\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^{T_n})$).
- 単語数は自由なので, 系列長 T_n も自由.

今回扱う問題

2. 音声認識(Speech Recognition)

- 発話を記録した時間信号から発話内容を推定する.
- 前処理
 - 信号は一定の周期で標本化され,
量子化されたデジタルデータ(=一般的な音声データ)
 - 方法の例: $10ms$ 間隔で $25ms$ 幅の窓で切り出し, 周波数スペクトルの分布情報を取り出して, 特徴ベクトルの系列($\mathbf{x}^1, \mathbf{x}^2, \dots$,)を得る.
- 入力に前処理を行ったデータを取り, 発話を構成する音素(phoneme) or 発話内容を直接表す文字列を推定する.

1, 2共に, 出力は入力と異なる長さの系列を出力できる必要がある

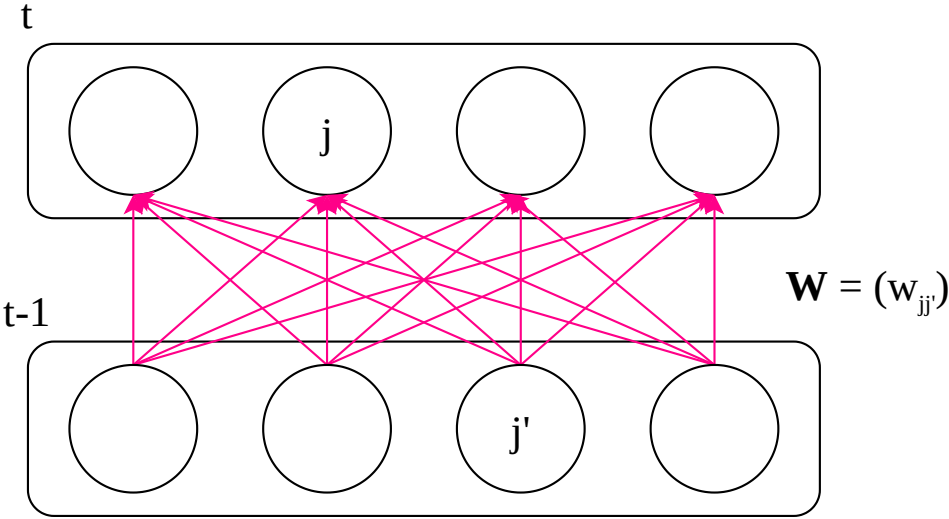
リカレントニューラルネットワーク

1. リカレントニューラルネットワーク(RNN)とはなんぞや？

A. リカレントニューラルネットワーク(Recurrent Neural Network)とは,
内部に(有向)閉路を持つニューラルネットワーク の総称である.

- 例として,
 - Elman Network
 - Jordan Network
 - Time Delay Network
 - Echo State Network など様々なものがあるが, 始めは単純なものを考える.

シンプルなRNN



前図のように,中間層のユニットの出力が自分自身に**重み付き**で戻されるRNNを考える.
この自分自身に戻ってくるパスを**帰還路**と呼ぶ.

この構造により, 中間層のユニットは, ひとつ前の状態を**覚える**ことができる.

また, このユニットは, ひとつ前の出力と, 現在の入力 of 両方を考慮して状態が変わるため, 振る舞いを**動的に変化させる**ことができる.

この二つの特性により, この単純なRNNは系列データ中の"文脈"を捉えることが期待される.

RNNは各時刻 t につき1つの入力 \mathbf{x}^t を受け取り, 出力 \mathbf{y}^t を出力する.

これは, **入力と同じ長さ**の系列を出力することを意味する.

過去に受け取った入力(理論上はすべて)が**帰還路**を通して出力に影響を与える.

順伝播型ネットワークとの比較

	順伝播型ネットワーク	RNN
帰還路	なし	あり
写像	$\mathbf{x}^t \mapsto y$	$(\mathbf{x}^0, \mathbf{x}^1, \dots, \mathbf{x}^t) \mapsto \mathbf{y}$

このRNNは, 系列データについて, 順伝播型ネットワークと同じ**万能性**を持つ.

RNNの順伝播

系列 $(\mathbf{x}^0, \mathbf{x}^1, \dots, \mathbf{x}^t)$ を順に入力すると, 系列 $(\mathbf{y}^0, \mathbf{y}^1, \dots, \mathbf{y}^t)$ を出力する.

$$y : (\mathbf{x}^0, \mathbf{x}^1, \dots, \mathbf{x}^t) \mapsto (\mathbf{y}^0, \mathbf{y}^1, \dots, \mathbf{y}^t)$$

この計算の詳細を次のスライドで説明するが, その前に定義を行う.

定義

- $\mathbf{x}^t = (x_i^t)$: ネットワークの入力
- $\mathbf{u}^t = (u_j^t), \mathbf{z}^t = (z_j^t)$: 中間層ユニットの入出力
- $\mathbf{v}^t = (v_k^t), \mathbf{y}^t = (y_k^t)$: 出力層ユニットの入出力
- $\mathbf{d}^t = (d_k^t)$: 目標とする出力

References

1. <https://qiita.com/mochimochidog/items/ca04bf3df7071041561a>
2. Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. Neural networks, 2(5), 359-366.