論文読みまとめ

Survey on Recurrent Neural Network in Natural Lang

井上智裕

2020 年 11 月 18 日

1　導入

1.1　☆ 扱う問題

自然言語処理（NLP）

1.2　☆ 問題意識

自然言語処理分野では、テキストを単なる記号や単語の羅列のように扱うのではなく、複数の単語が一つのフレーズを作り、複数のフレーズが一つの文を作り、最終的には文が思考やアイデアを伝えるというような言語の階層構造を考慮する必要がある。従来のニューラルネットワークでは、すべての入力（および出力）は互いに独立していると考えるが、実際には、文中の次の単語を予測するタスクなど、前の入力を考慮する必要があるタスクが多い。

2　理論

RNN は、各ニューロンに内部メモリを持たせ前の入力の情報を保持することにより、文（単語列）のような連続的なデータ列を扱えるようにしている。図 2.1 に RNN が展開されている様子を示す。

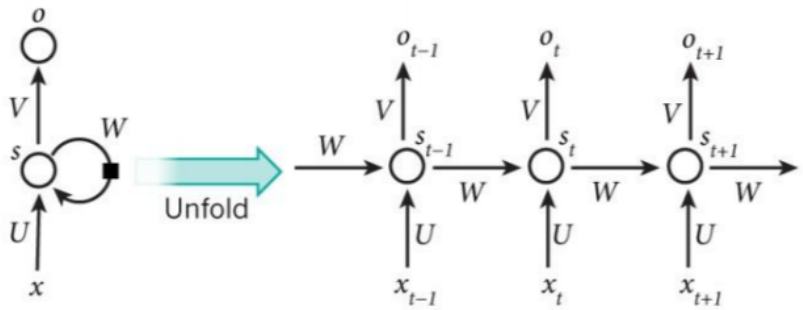

図 2.1　展開された RNN

x_t はステップ t における入力ベクトルであり、入力 x は $t = 1, \cdots, n$ に対して $x = [x_1, \cdots, x_n]$ である。例えば、$x = [\text{its}, \text{water}, \text{is}, \text{so}, \text{transparent}, \text{STOP}]$ のようになる。訓練データセット D_{train} は $D_{train} = \{x^1, \cdots, x^m\}$ なる集合を考える。RNN では、語彙の集合を V ($|V| = v$)、全ての実現しうる文を $V^{\max n}$ として、$\forall x \in V^{\max n}$ に対して、$P(x)$ を返すモデルを考える。$P(x)$ は (2.1) 式に示すように $x_1, \cdots, x_n$ の同時確率で与えられる。これに同時確率の定義を適用すれば (2.2) 式 が得られ、$P(x_2|x_1), \cdots, P(x_n|x_1)$ は互いに独立であるので (2.3) 式が得られる。

$$P(x) = P(x_1, \cdots, x_n) \tag{2.1}$$
$$= P(x_1)P(x_2, \cdots, x_n|x_1) \tag{2.2}$$
$$= P(x_1)P(x_2|x_1)\cdots P(x_n|x_1) \tag{2.3}$$

また、単語 k の重みを θ_k、単語の重みからなる行列を Θ、$t = 1$ から $t = n-1$ ステップまでの単語から抽出された特徴量を $\phi(x_{n-1}, \cdots, x_1)$ とすると、ある文脈に対して次に現れる単語の確率は (2.5) 式により求められる。

$$P(x_n = k|x_{n-1,\cdots,x_1}) = \frac{e^{\theta_k \cdot \phi(x_{n-1},\cdots,x_1)}}{\sum_{k'=1}^{V} e^{\theta_{k'} \cdot \phi(x_{n-1},\cdots,x_1)}} \tag{2.4}$$
$$P(x_n|x_{n-1,\cdots,x_1}) = \mathrm{softmax}(\Theta\phi(x_{n-1}, \cdots, x_1)) \tag{2.5}$$

$s_t \in \mathbb{R}^d$ はステップ t における隠れ状態であり、文脈情報を保持するネットワークのメモリに相当する。s_t は、入力 x_t と前の隠れ状態 s_{t-1} を用いて (2.6) 式により計算される。

$$s_t = f(Ux_t + Ws_{t-1}) \tag{2.6}$$

この関数 f としては、tanh や ReLU のような非線形関数を用いる。また、$U \in \mathbb{R}^{d\times v}$ は全単語ベクトルを含む行列であり、$W \in \mathbb{R}^{d\times d}$ はメモリがどのように渡されるかを制御する行列である。なお、初期の隠れ状態 s_0 は通常ゼロベクトルである。また、o_t はステップ t における出力である。単語列から次の単語を予測するタスクでは、各単語の出現確率を表すベクトルがこれに相当する。

$$o_t = \mathrm{softmax}(Vs_t) \tag{2.7}$$

$V \in \mathbb{R}^{v\times d}$ は文脈情報を単語の確率分布に変換する行列である。以上より、$t = 1, \cdots, n-1$ について、 (2.6) 式を順に計算することで s_{n-1} を求めることができ、(2.8) 式によって、$t = n-1$ までの文脈に対して $t = n$ に現れる単語の確率分布を求めることができる。

$$P(x_n|x_{n-1,\cdots,x_1}) = o_{n-1} = \mathrm{softmax}(Vs_{n-1}) \tag{2.8}$$

RNN では、単語ベクトルから作られる行列 U、隠れ層および出力層のパラメタ W、V を学習する必要がある。通常の誤差逆伝播法は RNN の再帰性のため機能しないため、Backpropagation Through Time (BPTT) を利用する。

なお、RNN は数ステップしか情報を保持できないという欠点があるが、Long Short Term Memory (LSTM) をネットワークに追加することでこれを改善できる。

3 結論・展望

RNN は内部メモリを持たせることで文脈情報を保持して、連続的なデータ列を扱うことができるため、近年非常に人気のある手法であり、多方面で利用されている。