再帰型ニューラルネットワーク

稲毛惇人

2018 年 2 月 20 日

1 はじめに

再帰型ニューラルネットワーク（Recurrent Neural Network, RNN）は，音声や言語，動画像といった系列データを扱う Neural Network（NN）である．これらのデータは一般にその長さがサンプルごとにまちまちで，系列内の要素の並び（文脈）に特徴を持つ．RNN はこのような特徴をうまく取り扱うことができ，さらに RNN の一種で長期の文脈をモデル化可能な “長・短期記憶（Long Short-Term Memory, LSTM）”が存在する．

ここでは，RNN のみをを取り扱う．

2 Recurrent Neural Network

RNN は内部に（有向）閉路をもつ NN の総称であり，この構造によって情報を一時的に記憶し，また振る舞いを動的に変化させる．これにより，系列データに存在する “文脈”を捉える事ができ，このことは順伝播型との大きな違いである．

RNN には，様々な構造のものがあるが，ここでは順伝播型ネットワークと同様の構造を持ち，ただし中間層のユニットの出力が自分自身に戻される「帰還路」を持つ（Elman 型の「コンテキスト層」に近い構造の）シンプルなものを考える．

RNN の出力層は順伝播型ネットワーク同様に設計する．出力層の活性化関数は，適用したい問題によって選ぶ．たとえば，分類問題では対象とするクラス数と同じ数のユニットを並べ，ソフトマックス関数を活性化関数に選択する．誤差関数も同様に選択する．

2.1 RNN の構造

RNN のノーテーションを表 1 に，モデル図を図 1 に示す．

2.2 RNN の順伝播計算

2.2.1 各層のバイアス

バイアスは仮想ニューロンを導入し，常に 1 を出力する特別なユニットを各層の 1 つ下の層に用意しこれと各ユニットとの結合重みをバイアスとする．

$$x_0^t = u_0^t = v_0^t = 1 \tag{1}$$

表 1: RNN の記法一覧

Notation of RNN	
i, j, k	: 入力層，中間層，出力層の各ユニットのインデックス
I, J, K	: 入力層，中間層，出力層のユニット数
N	: 総サンプル数
$\mathbf{x}^t = (x_i^t)$	: 時刻 t におけるネットワークへの入力
$\mathbf{u}^t = (u_j^t)$	: 時刻 t における中間層への入力
$\mathbf{z}^t = (z_j^t)$	: $\mathbf{u}^t$ を受け取ったときの中間層の出力
$\mathbf{v}^t = (v_k^t)$	: 時刻 t における出力層への入力
$\mathbf{y}^t = (y_k^t)$	: $\mathbf{v}^t$ を受け取ったときの出力層の出力
$b^{(in)}, b, b^{(out)}$	: 入力層，中間層，出力層のバイアス
$\mathbf{x^1}, ..., \mathbf{x}^T$	: T 時間までの入力系列
$\mathbf{y^1}, ..., \mathbf{y}^T$	: T 時間までの出力系列
$\mathbf{d^1}, ..., \mathbf{d}^T$	: T 時間までの目標系列
$E(\mathbf{w})$	: 誤差関数
$\mathbf{w}$	: RNN の重み
$\mathbf{W}^{(in)} = (w_{ji}^{(in)})$	: 入力層と中間層間の重み
$\mathbf{W} = (w_{jj'})$	: 中間層から中間層への帰還路の結合の重み
$\mathbf{W}^{(out)} = (w_{kj}^{(out)})$	: 中間層と出力層間の重み
$\mathbf{f}(\cdot), f(\cdot)$	: 入力層，中間層の出力が経由する活性化関数
$\mathbf{f}^{(out)}(\cdot), f^{(out)}(\cdot)$	: 出力層の出力が経由する活性化関数
$\delta_k^{out,t}$	: 時刻 t の出力層のユニット k におけるデルタ
δ_j^t	: 時刻 t の中間層のユニット j におけるデルタ
η	: 学習率

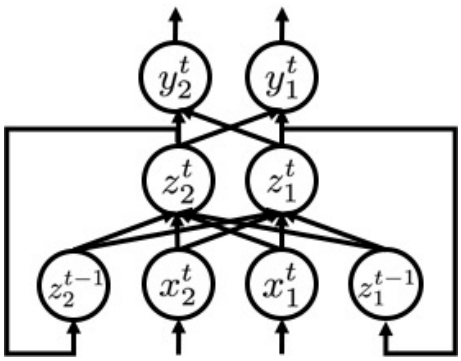

図 1: RNN のモデル図

$$\begin{cases} w_{j0}^{(in)} & = b^{(in)} \\ w_{j0} & = b \\ w_{k0}^{(out)} & = b^{(out)} \end{cases} \tag{2}$$

2.2.2 中間層の入力

$$\begin{aligned} u_j^t &= \sum_{i=1}^{I} w_{ji}^{(in)} x_i^t + b^{(in)} + \sum_{j'=1}^{J} w_{jj'} z_{j'}^{t-1} + b \\ &= \sum_{i=1}^{I} w_{ji}^{(in)} x_i^t + w_{j0}^{(in)} x_0^t + \sum_{j'=1}^{J} w_{jj'} z_{j'}^{t-1} + w_{j0} z_0^{t-1} \\ &= \sum_{i=0}^{I} w_{ji}^{(in)} x_i^t + \sum_{j'=0}^{J} w_{jj'} z_{j'}^{t-1} \end{aligned} \tag{3}$$

$$\mathbf{u}^t = \mathbf{W}^{(in)}\mathbf{x}^t + \mathbf{W}\mathbf{z}^{\mathbf{t-1}} \tag{4}$$

2.2.3 中間層の出力

$$z_j^t = f(u_j^t) \tag{5}$$

$$\mathbf{z}^t = \mathbf{f}(\mathbf{u}^t) = \mathbf{f}(\mathbf{W}^{(in)}\mathbf{x}^t + \mathbf{W}\mathbf{z}^{\mathbf{t-1}}) \tag{6}$$

$t = 1$ から始め，t を 1 ずつ増やしながら，入力系列 $\mathbf{x^1}, \mathbf{x^2}$... を用いて式 (6) を繰り返し計算することで，任意の時刻 t における中間層の状態 $\mathbf{z}^t$ を求めることができる．ただし，初期値 $\mathbf{z}^0 = (z_j^0)$ を与える必要があり，通常は $z_j^0 = 0$ とする．

2.2.4 出力層の入力

$$v_k^t = \sum_{j=1}^{J} w_{kj}^{(out)} z_j^t + b^{(out)} = \sum_{j=1}^{J} w_{kj}^{(out)} z^{t_j + w_{k0}^{(out)} v_0^t} = \sum_{j=0}^{J} w_{kj}^{(out)} z_j^t \tag{7}$$

$$\mathbf{v}^t = \mathbf{W}^{(out)}\mathbf{z}^t \tag{8}$$

2.2.5 出力層の出力

$$y_k^t = f^{(out)}(v_k^t) \tag{9}$$

$$\mathbf{y}^t = \mathbf{f}^{(out)}(\mathbf{v}^t) = \mathbf{f}^{(out)}(\mathbf{W}^{(out)}\mathbf{z}^t) \tag{10}$$

2.3 RNN の逆伝播計算

RNN の学習には，順伝播型ネットワーク同様に確率的勾配降下法が使われ，各層の重みについての誤差の微分を計算する必要がある．２つの方法が知られており，1 つは **RTRL 法**（**realtime reccurent learning**），もう 1 つは **BPTT 法**（**backpropagation through time**）である．前者はメモリ効率が良い一方，後者は計算速度が速く，またよりシンプルである．ここでは BPTT 法について説明する．

BPTT 法は，RNN を時間方向に展開し，順伝播型ネットワークに書き換えて誤差逆伝播計算を行う．具体的には図 2 に示すように，各時刻の RNN を横に並べ，中間層の帰還路を連続時刻での中間層間のユニットの結合として表現する．こうして展開したネットワークは，循環の少ない純粋な順伝播型ネットワークになり，誤差逆伝播による勾配の計算が可能になる．

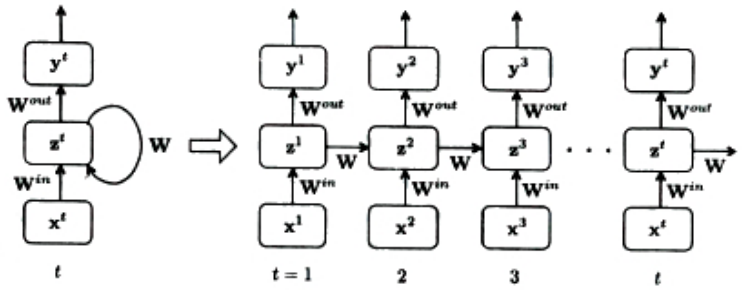

図 2: RNN を順伝播型ネットワークに書き換える時間方向への展開 [1]

2.3.1 出力層のデルタ

$$\delta_k^{out,t} \equiv \frac{\partial E}{\partial v_k^t} = \frac{\partial E}{\partial y_k^t}\frac{\partial y_k^t}{\partial v_k^t} = \frac{\partial E}{\partial y_k^t} f'^{(out)}(v_k^t) \tag{11}$$

2.3.2 出力層の誤差の勾配

式 (7) より，$w_{kj}^{(out)}$ は各 t の出力層のユニット k への入力 v_k^t のみに含まれる．

$$\frac{\partial v_k^t}{\partial w_{kj}^{(out)}} = z_j^t \tag{12}$$

$$\frac{\partial E}{\partial w_{kj}^{(out)}} = \sum_{t=1}^{T} \frac{\partial E}{\partial v_k^t}\frac{\partial v_k^t}{\partial w_{kj}^{(out)}} = \sum_{t=1}^{T} \delta_k^{out,t} z_j^t \tag{13}$$

2.3.3 帰還路内のデルタ

時刻 t におけるユニット j は，t での出力層のユニットと $t+1$ の中間層のユニットと全てにつながりがあるため，δ_j^t は式 (15) で表す．

$$\frac{\partial v_k^t}{\partial z_j^t} = w_{kj}^{(out)}, \frac{\partial u_{j'}^{t+1}}{\partial z_j^t} = w_{j'j} \tag{14}$$

$$
\begin{aligned}
\delta_j^t \equiv \frac{\partial E}{\partial u_j^t} &= \left(\sum_k \frac{\partial E}{\partial u_j^t} \frac{\partial u_j^t}{\partial v_k^t} \frac{\partial v_k^t}{\partial z_j^t} + \sum_{j'} \frac{\partial E}{\partial u_{j'}^{t+1}} \frac{\partial u_{j'}^{t+1}}{\partial v_k^{t+1}} \frac{\partial v_k^{t+1}}{\partial z_{j'}^{t+1}} \frac{\partial z_{j'}^{t+1}}{\partial u_{j'}^{t+1}} \frac{\partial u_{j'}^{t+1}}{\partial z_j^t} \right) \frac{\partial z_j^t}{\partial u_j^t} \\
&= \left(\sum_k \delta_k^{out,t} w_{kj}^{(out)} + \sum_{j'} \left(\delta_k^{out,t+1} w_{kj'}^{(out)} f'(u_{j'}^{t+1}) \right) w_{j'j} \right) \frac{\partial z_j^t}{\partial u_j^t} \\
&= \left(\sum_k w_{kj}^{(out)} \delta_k^{out,t} + \sum_{j'} \frac{\partial E}{\partial u_{j'}^{t+1}} w_{j'j} \right) \frac{\partial z_j^t}{\partial u_j^t} \\
&= \left(\sum_k w_{kj}^{(out)} \delta_k^{out,t} + \sum_{j'} w_{j'j} \delta_{j'}^{t+1} \right) f'(u_j^t)
\end{aligned}
\tag{15}
$$

各時刻 $t = 1, ..., T$ における出力層のデルタ $\delta_k^{out,t}$ が式 (11) によって与えられれば，$t = T$ から始め，1 つずつ t を小さくしながら式 (15) を繰り返し計算することで t における中間層のデルタ δ_j^t を求めることができる．ただし，$t = T + 1$ におけるデルタは $\delta_j^{T+1} = 0$ とする．各時刻における出力層のデルタ $\delta_k^{out,t}$ は順伝播時に計算済みの RNN の出力 $\mathbf{y}^t$ と，問題に応じた表現を持つ目標出力 $\mathbf{d}^t$ から誤差関数により計算される．

2.3.4 帰還路内の誤差の勾配

式 (3) より，$w_{jj'}$ は各 t の中間層のユニット j への入力 u_j^t のみに含まれる．

$$
\frac{\partial u_j^t}{\partial w_{jj'}} = z_j^{t-1} \tag{16}
$$

$$
\frac{\partial E}{\partial w_{jj'}} = \sum_{t=1}^{T} \frac{\partial E}{\partial u_j^t} \frac{\partial u_j^t}{\partial w_{jj'}} = \sum_{t=1}^{T} \delta_j^t z_j^{t-1} \tag{17}
$$

2.3.5 入力層の誤差の勾配

式 (3) より，$w_{ji}^{(in)}$ は各 t の中間層のユニット j への入力 u_j^t のみに含まれる．

$$
\frac{\partial u_j^t}{\partial w_{ji}^{(in)}} = x_i^t \tag{18}
$$

$$
\frac{\partial E}{\partial w_{ji}^{(in)}} = \sum_{t=1}^{T} \frac{\partial E}{\partial u_j^t} \frac{\partial u_j^t}{\partial w_{ji}^{(in)}} = \sum_{t=1}^{T} \delta_j^t x_i^t \tag{19}
$$

2.4 RNN の精度

RNN は，順伝播型ネットワーク同様に，中間層に十分な数のユニットがあれば，任意の系列から系列への写像を，任意の精度で近似できることが証明されている．また，系列データの全体が一括して与えられる場合は，その系列データを逆向き（$t = T, T-1, ..., 1$）に RNN に入力することもできる．

順伝播型ネットワークが入力 1 つに対して 1 つの出力を与える写像を表現したのに対し，ネットワーク内部にある帰還路によって，理論上過去のすべての入力から 1 つの出力への写像を表現する．しかし，実際には高々過去の 10 時刻分程度と言われている．この限界は，層数の多い深いネットワークでは，誤差逆伝播法によって勾配を計算する際，層を遡るにつれ，勾配の値が爆発的に増加するか，あるいは 0 に消滅しやすいためである．

上記の RNN が持つ勾配消失問題を踏まえ，長期間にわたる記憶を実現する方法の一つに**長・短期記憶**（**Long Short-Term Memory, LSTM**）がある．LSTM は，基本的な RNN に対し，その中間層の各ユニットをメモリユニットと呼ぶ要素で置き換えた構造を持つ．その他の構造は RNN と変わらない．

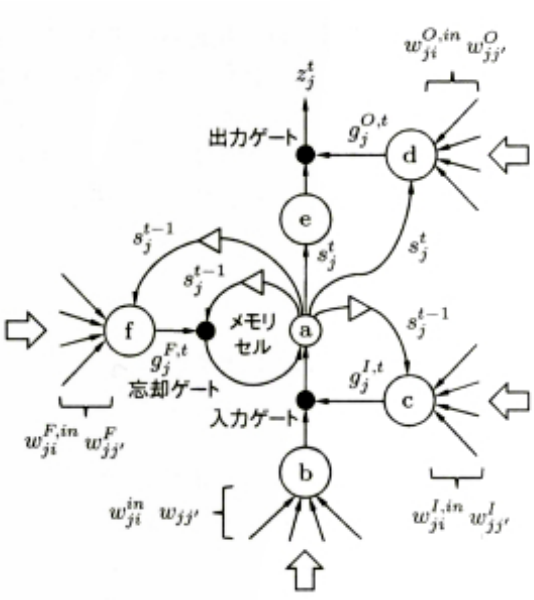

図 3: LSTM のメモリユニット [1]

参考文献

[1] 岡谷貴之, “深層学習 Deep Learning”, 講談社, 2015.