

LSTM

吉永 塁

2020 年 6 月 17 日

内容：LSTM について，特に LSTM が教科書 [1] に掲載されている形になるまでの流れを中心に．

- 記号の使い方は [1] に準じました．
- 指摘・補足があればお願いします．

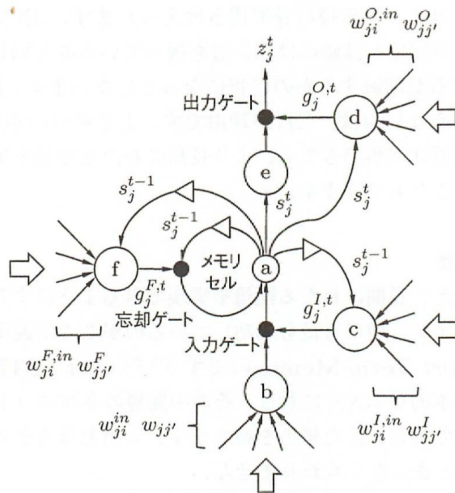
[1] 岡谷貴之．"深層学習" 第 7 章再帰型ニューラルネット

[2] "わかる LSTM ～ 最近の動向と共に"

https://qiita.com/t_Signull/items/21b82be280b46f467d1b

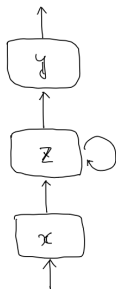
[3] S.Hochreiter and J.Schmidhuber. "LONG SHORT-TERM MEMORY"

LSTM 概要図



::(n' ~~~~~ n)::

Recurrent Neural Network

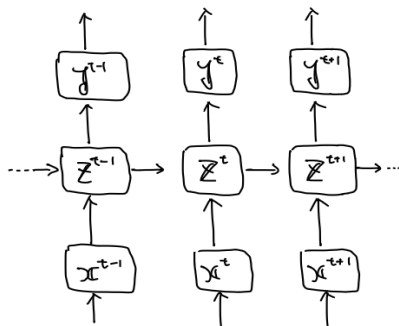


再帰型ニューラルネットワーク (reccurent neural network : RNN)

- 内部に閉路を持つニューラルネットの総称.
- (理論上) 過去の全ての入力から一つの出力への写像を表現する.
- 系列データに対応.

RNN の伝播計算

- RNN を時間方向に展開し、自己ループのない通常の順伝播型ネットワークと同様に扱う。
- 時間方向に展開すると深いネットワークとなり、勾配消失・勾配爆発につながる。
- 結果として、RNN の出力に反映できるのは高々 10 時刻程度。



Long Short-Term Memory

長・短期記憶 (long short-term memory : LSTM)

- RNN の勾配消失問題を解消して長期間の記憶の実現を目指す.
- RNN の中間層の各ユニットをメモリユニットに置き換えた構造.

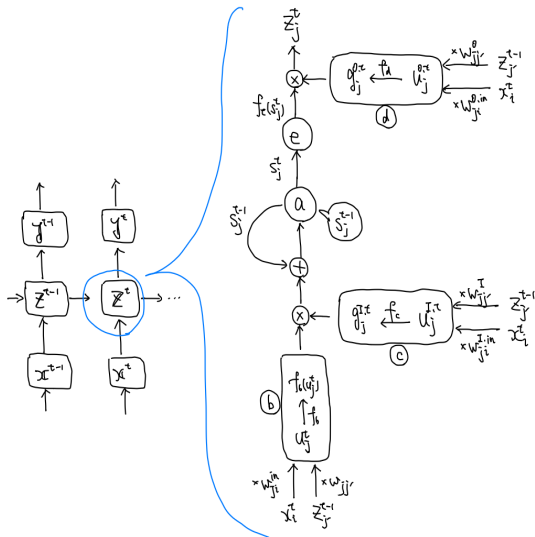
勾配消失問題について

- 解析の詳細は省略.
- 解決策 : 勾配消失回避のために自己ループの重みを 1 とする.

入出力重み衝突について

- 入力を伝達させたい vs 入力を止めたい (出力も同様)
- この競合が学習が遅れる原因になってしまう.
- 解決策 : 入出力ゲートを導入.

LSTM メモリセル/入出力ゲート

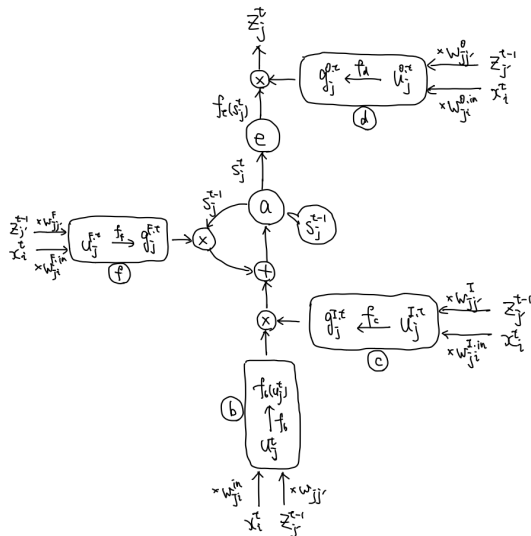


- 勾配消失問題の解決を目指した.
- メモリセル (a) : 一時刻前の記憶 s_j^{t-1} を保持.
- ユニット (e) : メモリセルの出力 s_j^t に活性化関数 f_e を適用.
- 入力ゲート (c), 出力ゲート (d) : 入出力を間引く.
- $f_c = f_d = \sigma$: ゲート値 $g_j^{I,t}, g_j^{O,t}$ を $[0, 1]$ に制限.
($\sigma(x) := (1 + \exp(x))^{-1}$)

- これが最初に提案された形 (1997 年).
- メモリセル (a) は CEC (Constant Error Carousel) とも呼ばれる.

LSTM(メモリセル/入出力ゲート) の問題点

- メモリセルは記憶を保持し続ける.
- 入力系列のパターンが大きく変動したときに, 不要な過去の記憶が残ったままになる.
- 不要になった過去の記憶を自動的に忘れるような構造にする.
- 解決策: 忘却ゲートの導入.



- 入力系列のパターンが大きく変動する場合に対応.
- 忘却ゲート (f) : メモリセルに記憶した情報 s_j^{t-1} を「忘れる」.
- 今までの記憶が不要になる (入力パターンが変動する) ときにメモリセルを初期化できる.
- $f_f = \sigma$: ゲート値 $g_j^{F,t}$ を $[0, 1]$ に制限.
 $(\sigma(x) := (1 + \exp(x))^{-1})$

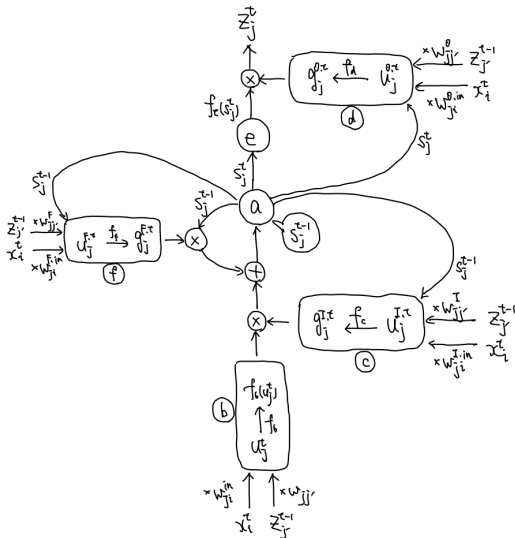
- 忘却ゲートによって、不要になったメモリセルの内容を忘れることができるようになった。

LSTM(忘却ゲート) の問題点

- 各ゲートの役割は,
 - 入力ゲート：メモリセルの内容を書き換えるか
 - 忘却ゲート：メモリセルの内容を忘れるか (保持するか)
 - 出力ゲート：メモリセルの内容を出力するか

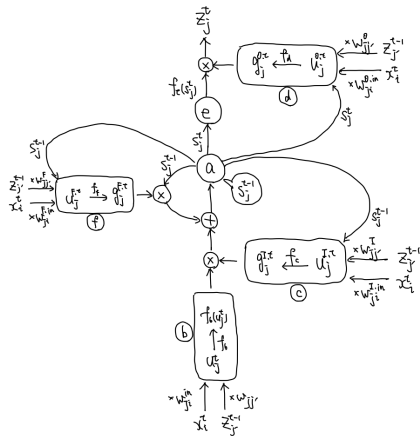
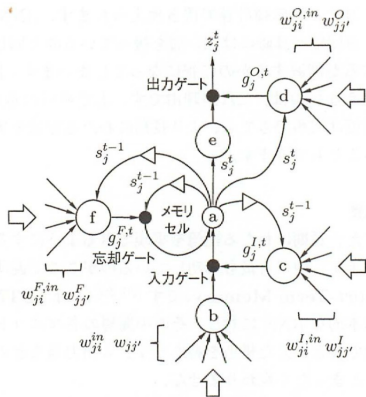
であって、何れもメモリセルの制御が役割。

- ゲートは制御対象であるメモリセルの状態を直接参照できない。
- 解決策：覗き穴結合の導入。



- ゲートの制御にメモリセルの内容を利用.
- これが教科書 [1] に載っている形.
- 構造の詳細は異なることがある.
 - 活性化関数 f_e の省略
 - 覗き穴結合にも重み付け

LSTM 概要図



順伝播計算は次のようになる.

$$z_j^t = g_j^{O,t} f(s_j^t)$$

$$g_j^{O,t} = f_d(u_j^{O,t}) = f_d \left(\sum_i w_{ji}^{O,in} x_i^t + \sum_{j'} w_{jj'}^O z_{j'}^{t-1} + s_j^t \right)$$

$$s_j^t = g_j^{F,t} s_j^{t-1} + g_j^{I,t} f_b(u_j^t)$$

$$g_j^{F,t} = f_f(u_j^{F,t}) = f_f \left(\sum_i w_{ji}^{F,in} x_i^t + \sum_{j'} w_{jj'}^F z_{j'}^{t-1} + s_j^{t-1} \right)$$

$$g_j^{I,t} = f_c(u_j^{I,t}) = f_c \left(\sum_i w_{ji}^{I,in} x_i^t + \sum_{j'} w_{jj'}^I z_{j'}^{t-1} + s_j^{t-1} \right)$$

$$u_j^t = \sum_i w_{ji}^{in} x_i^t + \sum_{j'} w_{jj'} z_{j'}^{t-1}$$

(逆伝播は省略. 教科書 [1] を参照.)