

# ADL—LSTM、GRU

2019年4月9日 上午 11:43

本來RNN因為多次的運算而會消失先前的數值，gradient vanishing  
RNN是只會把兩邊的input拉進來經過tanh就輸出兩邊了

- LSTM
  - 左邊變成有兩條資訊
  - ct: 希望資訊直接流過去到右邊，快速通道直接輸出(不經過 computation)
    - 加上一個gate: 會呈上一個數值決定要讓多少資訊流過去，控制大門開跟關可以流過去到下一個時間點
    - 學出開關的關係
    - Forget gate
      - sigmoid layer
      - 告訴ct-1多少資訊可以留到後面，決定有多少資訊要丟掉忘記的
      - 拿前一個state的hidden state，跟現在這個時間點的input，學出要多少的forget，學一個output要多少資訊要被忘掉
      - 去學到控制資訊對input forget情況
      - =1代表完全保留；=0表示都不管input
    - EX: 動詞會跟主詞有變化。但是stop word恐怕沒什麼用，所以可以忘掉
    - 根據input跟hidden決定資訊要否忘掉
  - Input Gate
    - 決定有多少新進來的資訊要存起來
    - 學的input多少要存到hidden state裡面。(一般RNN只會經過tanh就都存下來)
    - input gate只有部分要存近來，藉由另一個sigmoid控制
    - 是否要僅保留新subject的資訊
  - Ct
    - 把多少要留的(forget gate)，再加上新的資訊(input gate)
  - Output Gate
    - 有多少新的資訊是要輸出的
    - 根據x跟h-1，要怎麼把算出來的Ct輸出出來
- Peephole
  - 在算sigmoid的時候可以access Ct-1，可以偷看
- Coupled Forget & Input
  - 決定多少忘掉，就要新增多少Input
  - 可以share資訊，一個ft，另一個就是1-ft
  - 減少參數

- GRU
  - 比LSTM簡化
  - 把forget跟Input合再一起變成一個update gate，類似coupled input/forget
  - 現在不管兩個的關係只學一個資訊

GRU 所需的data比較少，參數比較少，不容易overfitting  
應該比LSTM還要好

#### 結論

- 解決gradient vanishing問題
- 變形效果比較好
- GRU可以一起學