Gated Feedback Recurrent Neural Networks

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho and Yoshua Bengio

ICML 2015

組員: 張嘉哲、林暘竣

簡介

Recurrent Neural Networks (RNN) 已經被廣泛運用在sequence model相關的機器學習，特別是輸入、輸出長短不一的任務上，但RNN對於long-term dependency的輸入就不太能處理，只會根據前一個hidden state的狀態影響現在的state。

因此發展出gated activation function: Long Short-Term Memory (LSTM) 和 Gated Recurrent Unit (GRU)，可以處理long-term dependency和short-term dependency。

這一篇paper為RNN發展出新的設計Gated-feedback RNN (GF-RNN)，使用在stack RNN上面可以連接先前N個hidden state而不是只有1個。在language modeling以及python program evaluation的實驗結果裡證明了改良後的GF-RNN可以獲得較佳的成果。

接下來會先介紹RNN、LSTM以及GRU的核心架構，再來介紹GF-RNN以及如何使用在其他架構上，再來是實驗結果，最後是我們的心得感想。

架構介紹

RNN

現在的state會根據先前的state決定狀態，其數學式是

，其中常是logestic sigmoid function、W為input的權重、U是state-to-state reccurrent matrix、x為input。

LSTM

對於中間的memory cell 有3個gate在控制他分別為input gate 、forget gate 以及output gate ，這三個gate都會被之前的hidden state影響其數學式分別為

對於memory cell 有foget gate控制他是否要忘記先前的memory以及input gate是否輸入進去，其數學式為

其中

最後output出來的hidden state也有output gate控制是否輸出其式子為

GRU

GRU的設計也是有gate在控制其memory，但不同於LSTM的地方是他只有2個gate在控制，其為reset gate和update gate 。對於state 其數學式為

而其中的是介於0~1的比率，決定之前的state有多少會留下來，其數學式為

新的memory要進來會有一個reset gate控制要不要含有先前的state值，其數學式為

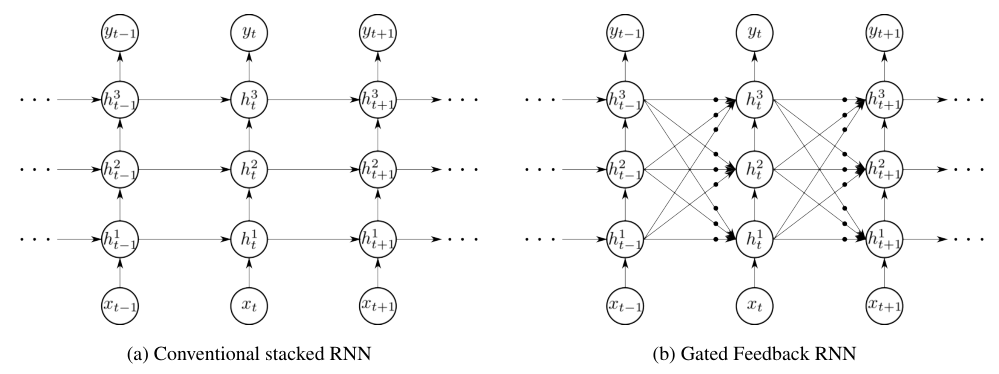
其中為

，當值逼近為0時候就reset不考慮先前state的值。

GF-RNN

GF-RNN最主要新增的gate是global reset gate，不僅僅只有前一時間的一個state的weight可以影響reset gate，而是前一時間許多層state的weight可以影響reset gate，其數學式為

與分別為hidden state和input的weight matrix，L表示第幾個hidden layer，如果今天j=1則為。圖(1)為傳統RNN與GF-RNN的比較圖。



圖(1)

使用GF-RNN

一般的RNN

只要把

改成

LSTM

把

改成

GRU

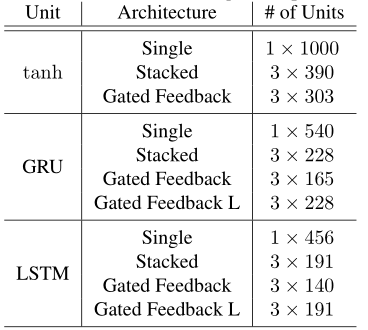
把

改成

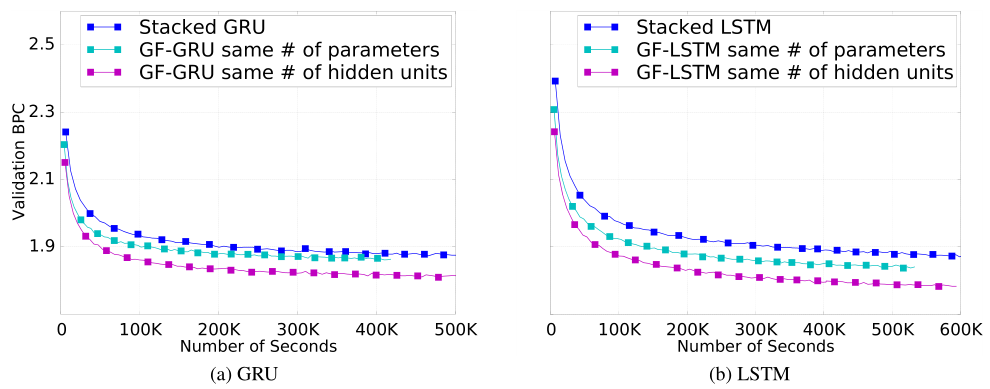
只要在hidden state上面增加global在上面即可使用。

實驗結果

使用在language model上面的話，我們會比tanh、GRU以及LSTM分別作用在Single、Stacked、Gated Feedback上面的結果，圖(2)是units數量的比較圖、圖(3)為結果圖。

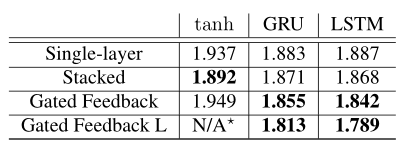


圖(2)



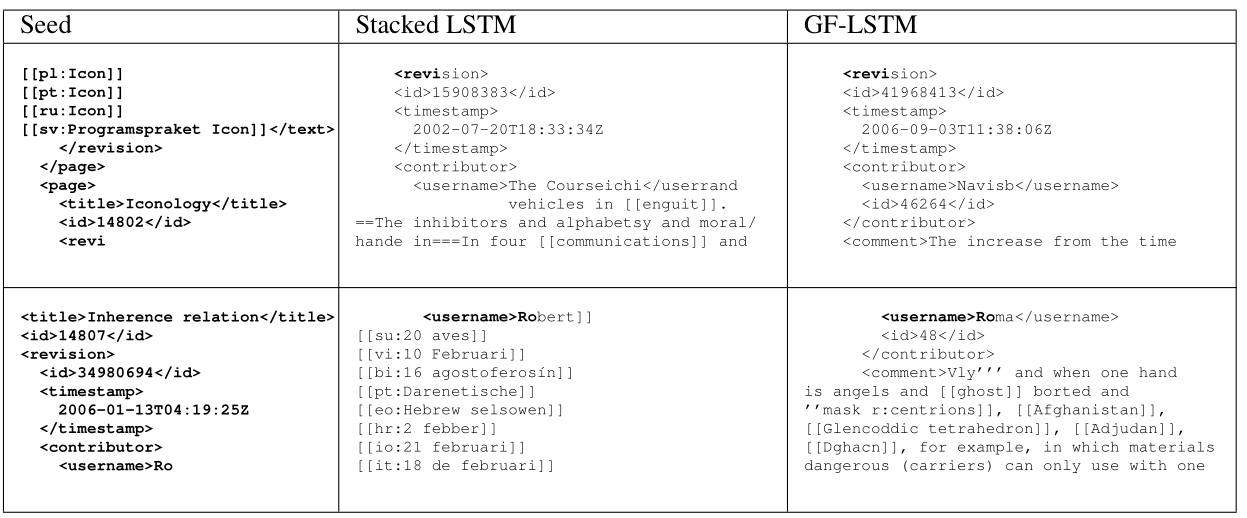
圖(3)

可以很明顯看到如果使用了GF-RNN可以使得Validation BPC有較佳的結果無論在GRU或者LSTM上面。而圖(4)為訓練100個epochs後的BPC數據圖。從結果得知，當Unit使用LSTM和GRU，並參考了前時間點每層的資訊，確實表現得比較好，而這樣的作法在使用tanh units時並沒有變好，論文雖沒點出原因，但或許是因為LSTM和GRU都有使用gate來保護他們的hidden state，Unit本身沒有加入gate的vanilla-RNN with tanh units參考了更多的資訊，反而無法控制，造成反效果。



圖(4)

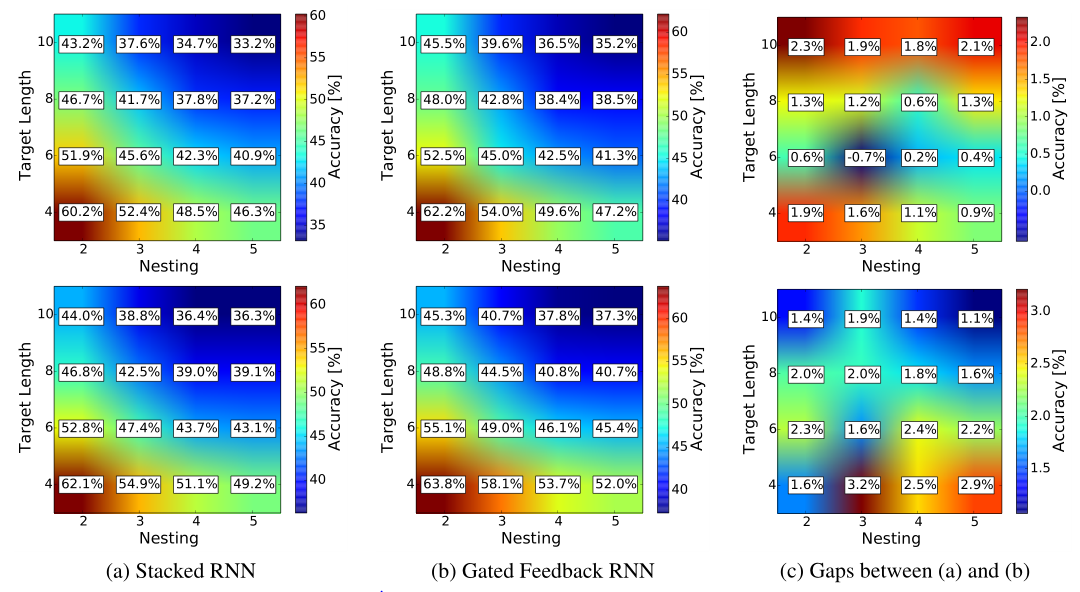
使用在python program evaluation上面的話，結果如圖(5)。



圖(5)

會發現如果使用Gated Feedbacck的話，結果會比較接近seed。

在來使用熱度圖來看python program evaluation，如圖(6)。



圖(6)

會發現當nesting與target length越大的時候accuray會逐漸下降，而圖(6-c)比較了使用gated feedback前後的正確率差距，也明顯看到使用gated feedbacck後正確率明顯提高了。

結論與心得感想

Gated feedback使用在原有的RNN、LSTM以及GRU上面，的確提高了正確率，讓結果變得更好。global reset gate得控制前層資訊的強度，因為這些資訊對於下個時間點可能無關或影響很大，所以加入gate來限制他們，經由作者的實驗也可知，使用了global reset gate比沒使用時效果更好。另外，加入了全連接後，參數量勢必大增，但實驗也顯示，應用了GF-RNN的結果和一般堆疊的GRU和LSTM比較下，相近參數的架構，及相同層數而且相同Unit數的架構，其收斂速度並沒有比較慢，甚至結果還更好，由此可知GF-RNN雖然參數量多，但訓練的過程快，代表參考了更多的資訊後，參數更新的方向可能也比較明確。

本篇論文本身即是提出一個很容易應用及加入的方法。一些像是QA方面的work，在問題也就是句子部分的feature，有時候使用CNN的表現比LSTM好，或許應用了堆疊的LSTM，並全連接不同時間的Unit後，能取出更好的特徵，也就是本篇論文的做法，在處理language model的問題或其他序列性的資料的問題時，能帶來更好的效果。

目前deep learning深受世人的重視，而有蓬勃的發展，並持續改良現有的架構，試著讓deep learning更接近人腦，也期望未來會有更多良好的架構與機制被發展出來，讓我們真正迎向AI的未來。