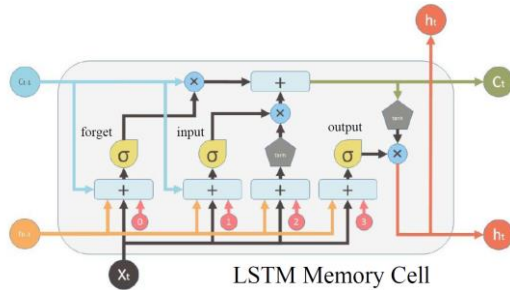


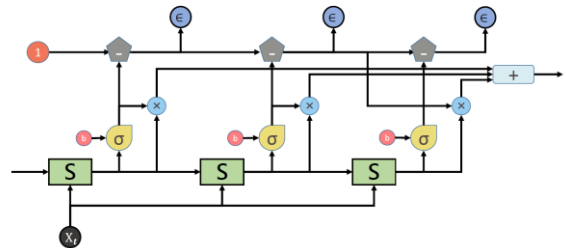
機器學習 手寫二

405235035 資工三 王博輝

1. 從資料流中說明'ACT RNN'和'LSTM'的差異



Adaptive computation Time RNN (ACT RNN)



(1)

LSTM 的輸入是 $x(t)$ 和 $C(t-1)$ 和 $h(t-1)$ ，輸出有 $h(t)$ 及 $C(t)$ 。

ACT RNN 的輸入只有 $s(t-1)$ 和 $x(t)$ 而已，ACT RNN 中並無一個獨立的 cell state 來儲存額外的狀態資訊，比較像是原本 RNN 的樣子，但也有可能是像 GRU，將 hidden state 和 cell state 合併了，總之就是沒有一個獨立的 cell state 來當作 memory 使用。

(2)

LSTM 每一次 step 的操作數的一樣，都是先決定要忘記 cell state(memory) 中的哪些資訊，再決定要新加入哪些資訊進入 cell state 中，最後再根據 cell state 中的資訊來算出當下這一 step 的 Cell state $C(t)$ 、hidden state $h(t)$ 和 output $y(t)$ 。

ACT RNN 的精神類似基本的 RNN，不過它可以動態地調整每一次 step 的計算量，每一個 S 就代表進行一次 RNN 的計算，當 input $x(t)$ 及 $s(t-1)$ (可能也會有前一個 S 計算的結果當輸入) 進行完計算後，計算出的結果會由 halting neuron (一個 sigmoid 函數) 來給出一個緩衝權重，然後此多次計算的結果會與各自的緩衝權眾相乘後再相加起來，成為下一次 step 的 hidden state $s(t)$ ，而在 ACT RNN 最上面的部分，我們計算出的緩衝權重總和會

是 1，當 1-緩衝權重小於 ϵ 時，計算就會停止，所以緩衝權重也可以當作是 ACT RNN 在當下那一步停止的機率，而剩下的緩衝值(小於 ϵ 的部分)則會經過計算後留給下一個 step 來做使用。

2. 說明 ch11 p.37 中的 S 是什麼

S 為一個相乘相加的 function，做的事基本上和普通的 RNN 一樣，function 的 input 及 output 為 $s_t = \mathcal{S}(s_{t-1}, W_x x_t)$ ，吃一個輸入 $x(t)$ 及上一次輸出的狀態 $s(t-1)$ ，輸出為現在的狀態 $s(t)$ 。

$$s_t^n = \begin{cases} \mathcal{S}(s_{t-1}, x_t^1) & \text{if } n = 1 \\ \mathcal{S}(s_t^{n-1}, x_t^n) & \text{otherwise} \end{cases}$$

從第 2 個 S 開始，則會把前一個 S 的結果當作 input 一同輸入。