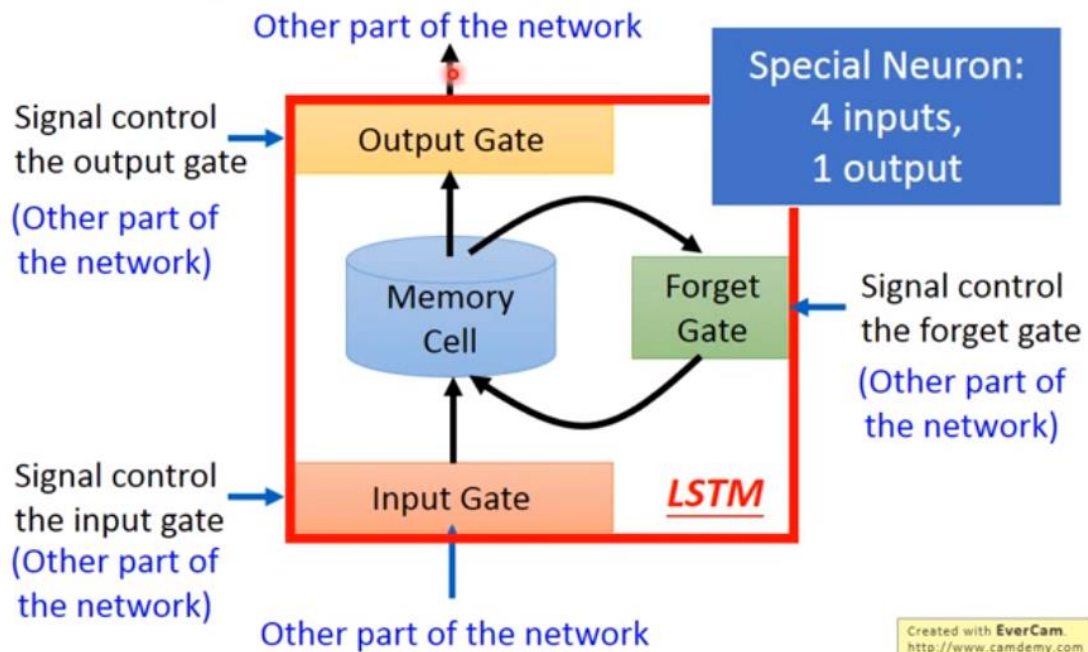
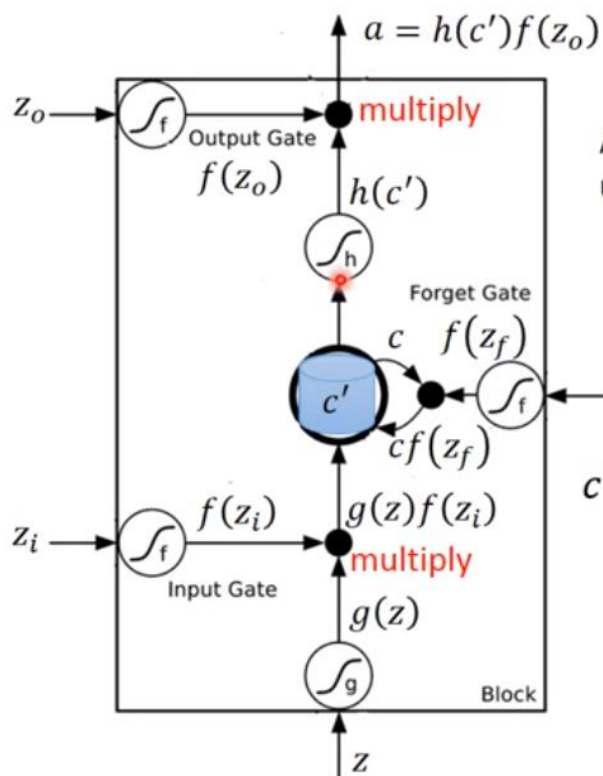


Long Short-term Memory (LSTM)





Activation function f is usually a sigmoid function

Between 0 and 1

Mimic open and close gate

$$c' = g(z)f(z_i) + cf(z_f)$$

- 符號約定：

1. Input: z
2. 操作 Input Gate: $z_i z_i$
3. 操作 Forget Gate: $z_f z_f$
4. 操作 Output Gate: $z_o z_o$
5. Output: a
6. Memory Cell: C

- 流程說明：

1. z 通過 sigmoid function 得到 $g(z)$
2. $z_i z_i$ 通過 sigmoid function 得到 $g(z_i)$
3. 1、2 所得相乘得到 $g(z)f(z_i)$
4. $z_f z_f$ 通過 sigmoid function 得到 $g(z_f)$
5. Memory Cell 乘上 Forget Gate 的值，得到 $cf(z_f)$
 - 當 forget gate 的輸入為 0，那代表之前存在 memory cell 的值會變 0
 - 當 forget gate 的輸入為 1，那代表之前存在 memory cell 的值會保留
6. 將 5、3 相加得到 c'
 - $c' = g(z)f(z_i) + cf(z_f)$
 - 當 input gate 的輸入為 0，那就相當於沒有更新
 - 當 input gate 的輸入為 1，那就相當於直接把 $g(z)$ 當做輸入
7. Memory Cell c 變為 c'

8. $c'c'$ 通過 h function 得到 $h(c')h(c')$

9. $z_o z_o$ 通過 sigmoid function 得到 $f(z_o)f(z_o)$

- 當 output gate 的輸入為 1，代表可以通過
- 當 output gate 的輸入為 0，則 $f(z_o)f(z_o)$ 為 0，代表無法通過

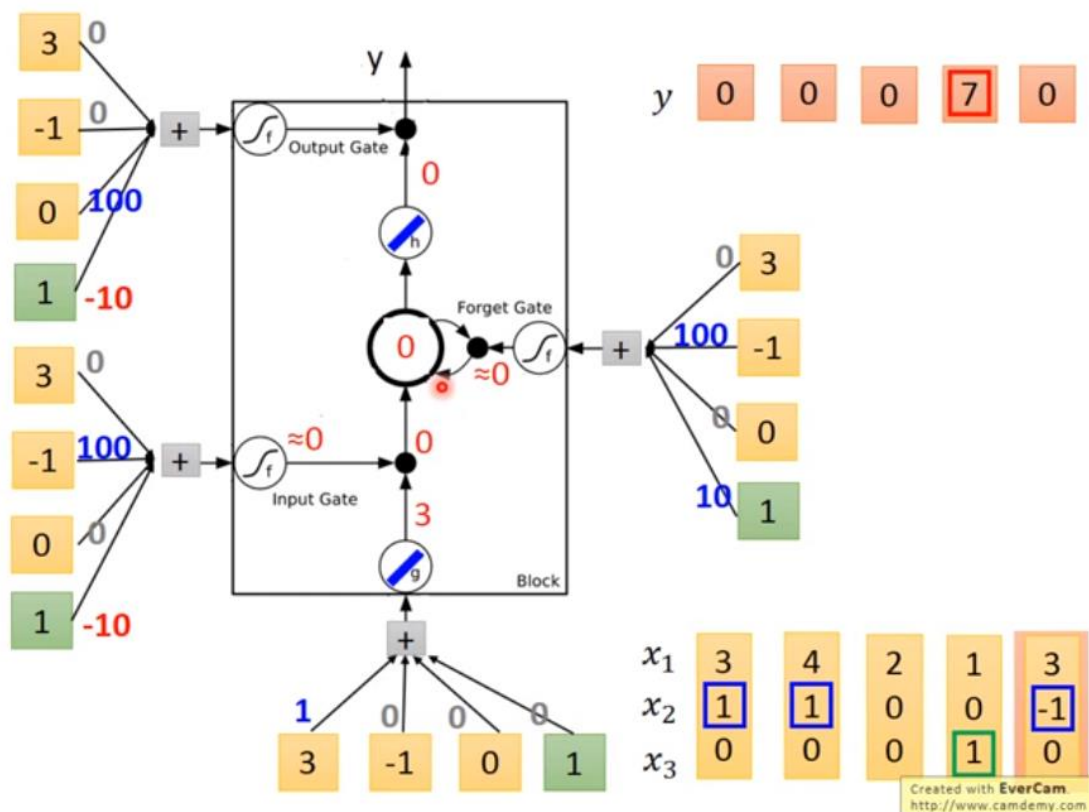
10. $9、10$ 相乘得到 $a=h(c')f(z_o)a=h(c')f(z_o)$

- activation function:

- 通常選擇使用 Sigmoid Function，因為輸出是介於 0、1 之間的值，用以代表 gate 被打開的程度

範例說明：

- input 為三維的 vector
- output 為一維的 vector
- 神經網路中只有一個 LSTM 的 cell
- x_2x_2 為 1 的時候， x_1x_1 的值就會被寫入 memory
- x_2x_2 為 -1 的時候， x_1x_1 的值就會被重置
- x_3x_3 為 1 的時候，output gate 才會開啟，才看的到 output
- 初始 memory 為 0



上排的值是代表每一個時步結果，在 timestep 為 3 的時候因為 timestep2 的 x_2x_2 為 1，因此記憶，以此類推。

上圖是實際執行一個過程，依之前所提，一個 memory cell 會有四個 input，三個 gate，一個 output，權重與偏差單元都是學習過程中學習到的，範例中假設我們已經得到相關學習參數並且初始 memory cell 為 0。

- input:
 - 僅第 1 個權重有值
- input_gate:
 - 僅第 2 個權重有值，並且偏差單元為-10，這代表 x_2x_2 為 0 則輸出為-10，經過 sigmoid 之後是接近 0，因此關閉
- forget_gate:
 - 僅第 2 個權重有值，並且偏差單元為 10，這代表平常是開啟狀態(開啟代表記憶)，在 x_2x_2 輸入是很大的負值時才會啟動重置。
- output_gate:
 - 僅第 3 個權重有值，並且偏差單元為-10，代表必需 x_3x_3 有值並大於偏差單元情況下才會輸出。
- memory_cell: init 0

計算：

1. input: (3,1,0)
 - input: 3
 - input_gate: open
 - forget_gate: open
 - memory_cell: 3
 - output_gate: close
 - output: 0
2. input: (4,1,0)
 - input: 4
 - input_gate: open
 - forget_gate: open
 - memory_cell: 7
 - output_gate: close
 - output: 0
3. input: (2,0,0)
 - input: 2
 - input_gate: close
 - forget_gate: open
 - memory_cell: 7
 - output_gate: close
 - output: 0
4. input: (1,0,1)
 - input: 1
 - input_gate: close
 - forget_gate: open
 - memory_cell: 7
 - output_gate: open
 - output: 7
5. input: (3,-1,0)
 - input: 3
 - input_gate: close
 - forget_gate: close
 - memory_cell: 0
 - output_gate: close
 - output: 0