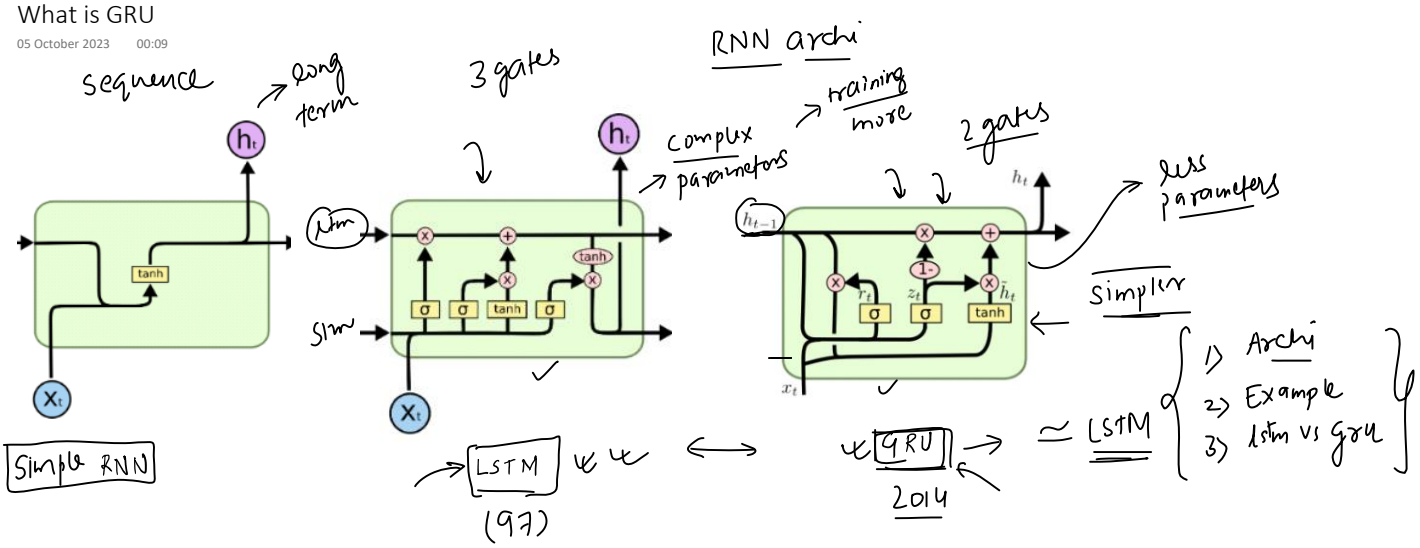
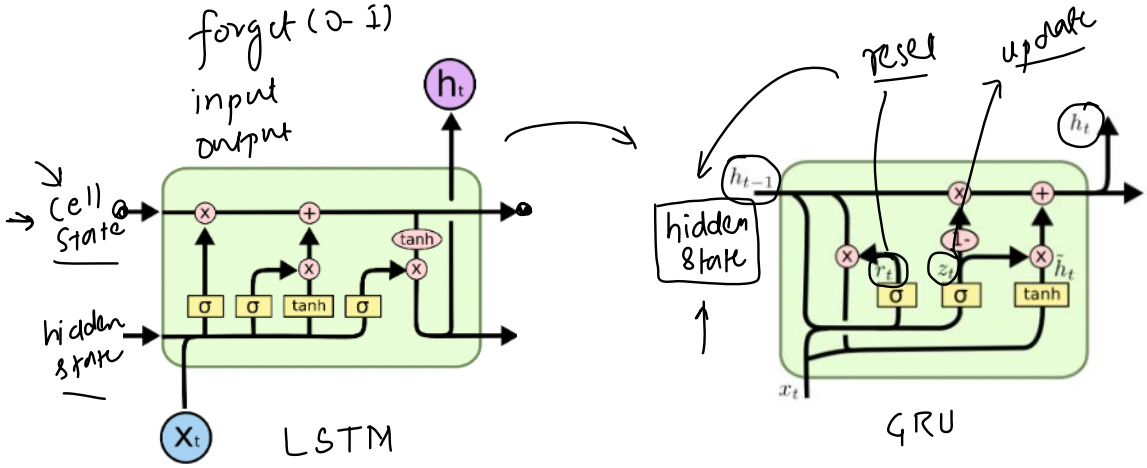
What is GRU

05 October 2023 00:09



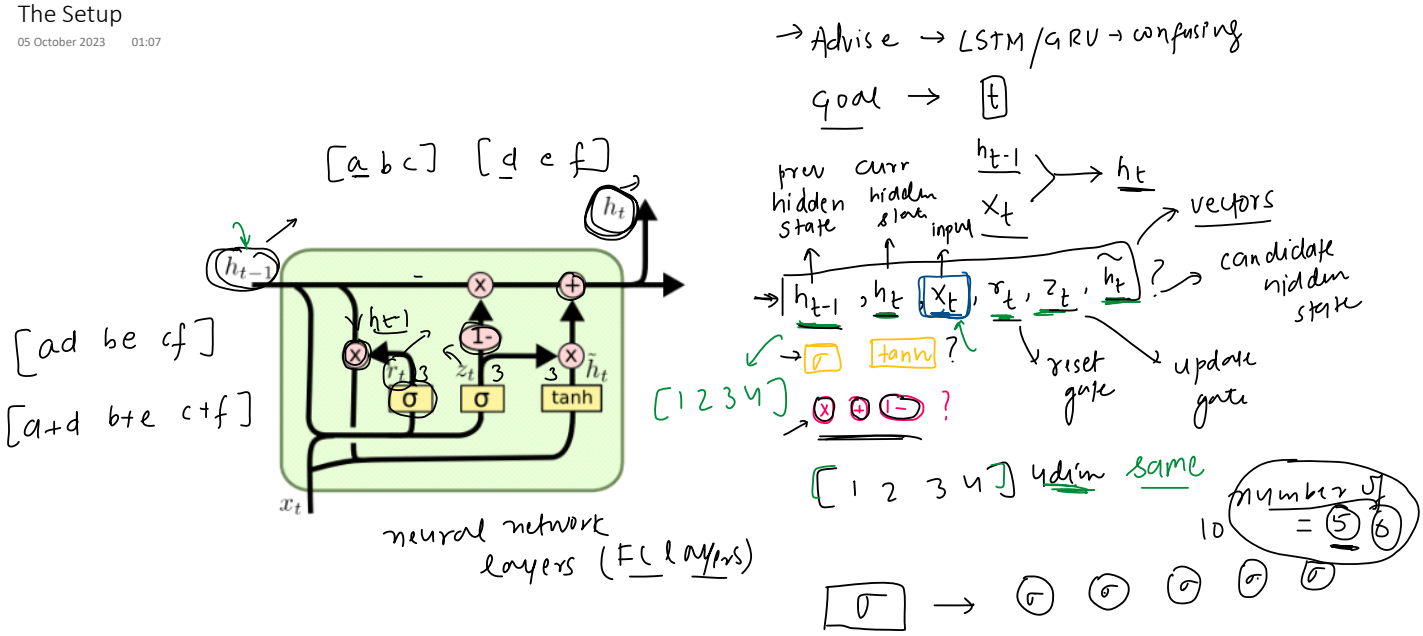
The Big Idea Behind GRU

05 October 2023 00:47



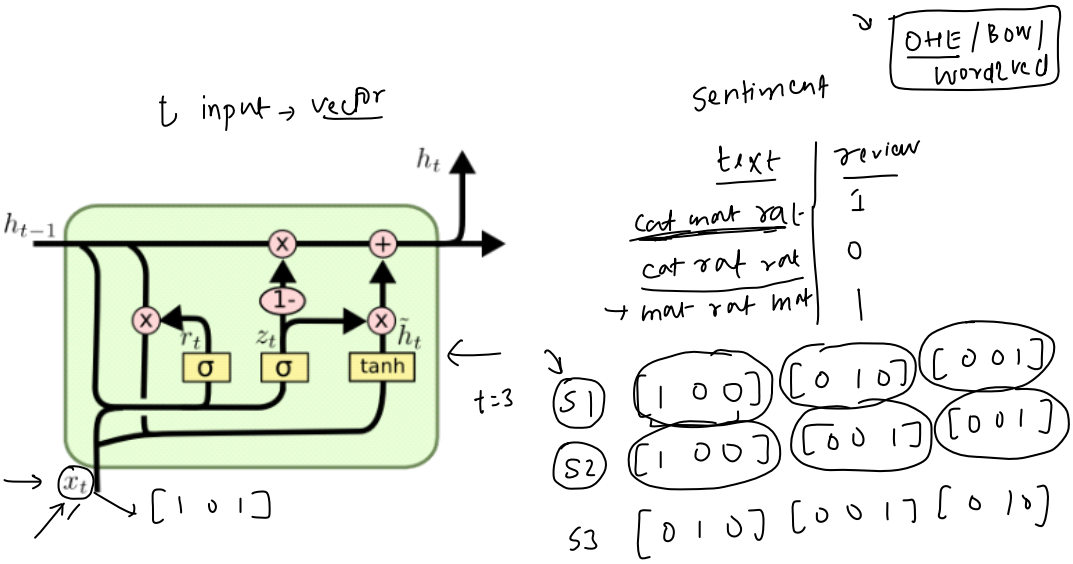
The Setup

05 October 2023 01:07



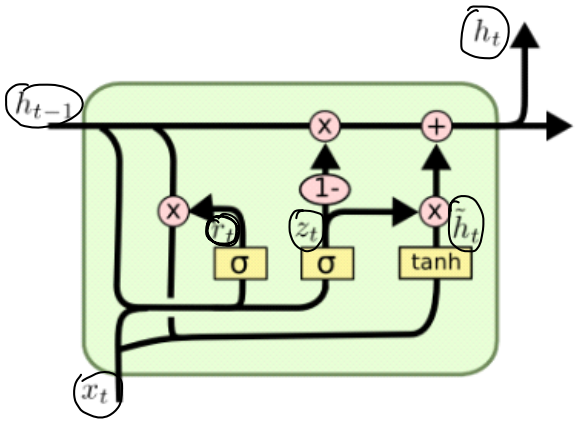
The Input Xt

05 October 2023 01:52



Architecture

05 October 2023 02:10



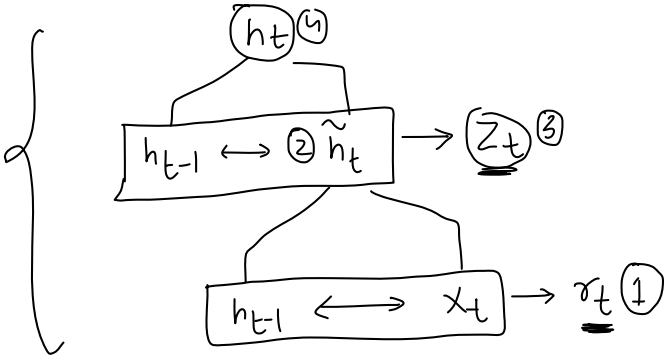
h_{t-1} , x_t ⟶ h_t

→ 1) Calculate r_t (reset gate)

→ 2) Calculate $\tilde{h}_t$ (candidate hidden state)

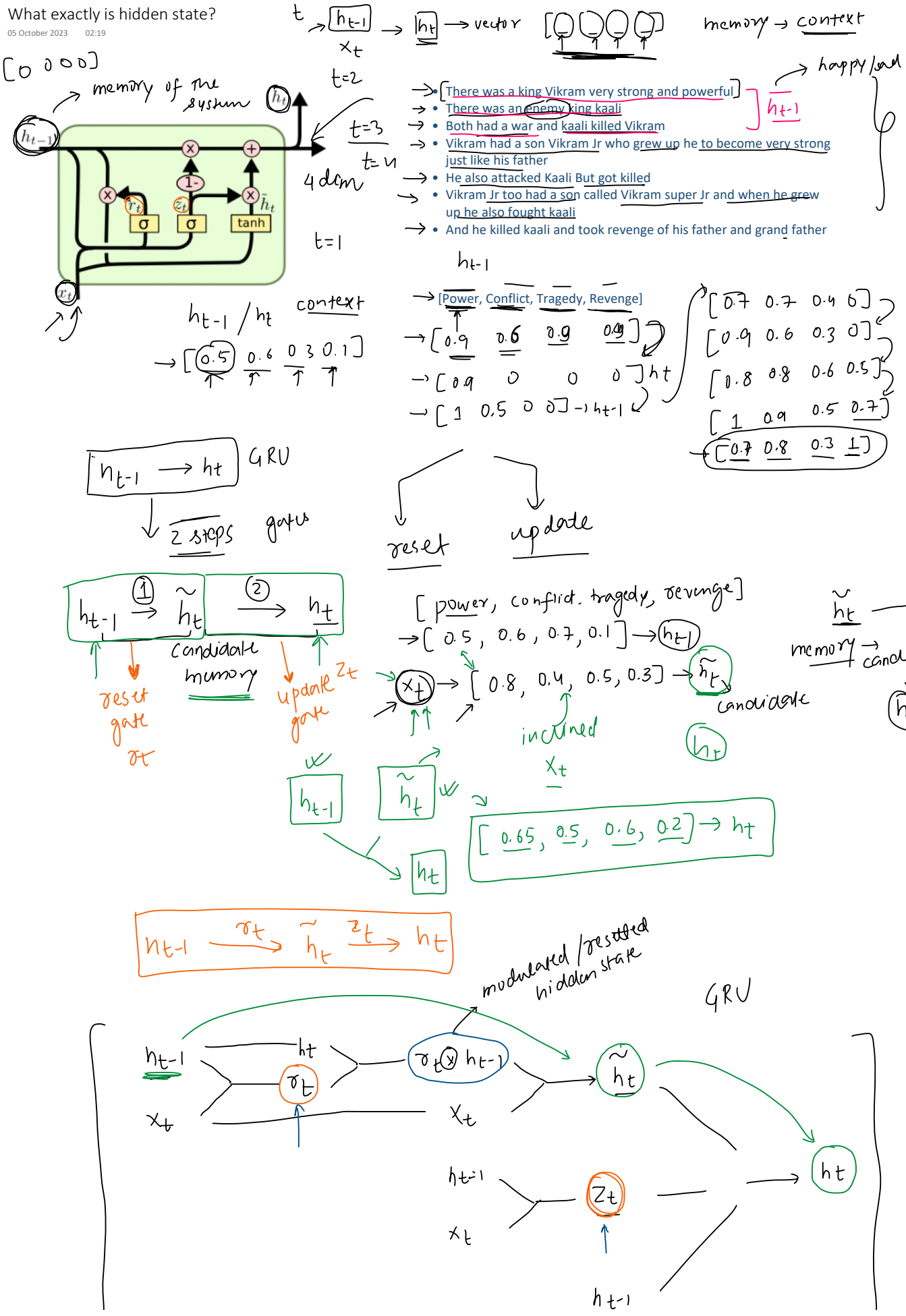
→ 3) Calculate z_t (update gate)

→ 4) Calculate h_t (current hidden state)



What exactly is hidden state?

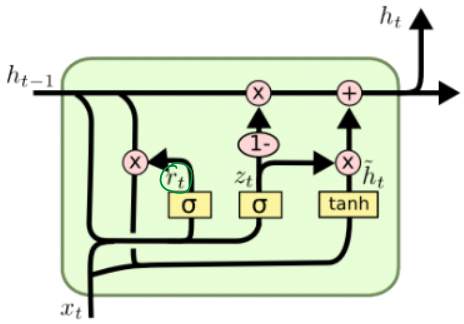
05 October 2023 02:19

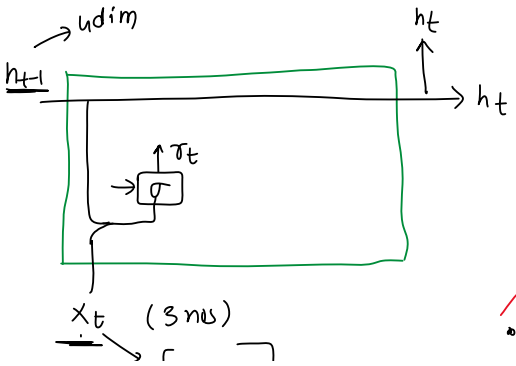


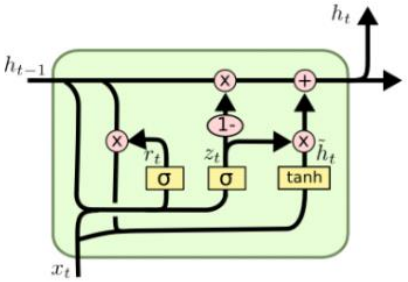

h_{t-1}

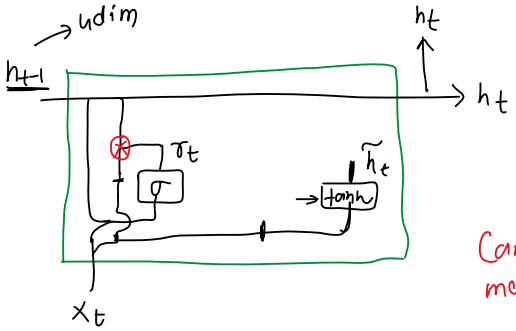
Calculating the reset gate

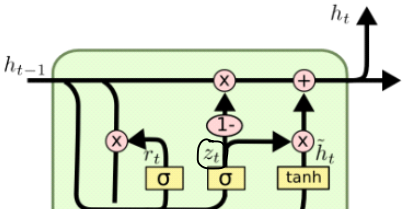
05 October 2023 14:24











weighted sum

reset

r_t → vector → h_{t-1}

↳ gate

$h_{t-1} = [0.6, 0.6, 0.7, 0.1]$ (lower lower higher)

$r_t = [0.8, 0.2, 0.1, 0.9]$

gate — 80% percent — 20% — 90%

reset

$r_t = \sigma(W_r[h_{t-1}, x_t] + b_r)$

(4) ← $(7 \times 4) = 28 + 4$

$(7) \times 1 \rightarrow (1 \times 7)$

modulated reseted

$(1 \times 7)\ (7 \times 4)$

$(1 \times 4) + (1 \times 4) \rightarrow ((1 \times 4))$

$(1 \times 4) \rightarrow r_t$

$\tilde{h}_t = [0.7\ 0.2\ 0.1\ 0.2]$

$h_{t-1} = [0.6, 0.6, 0.7, 0.1]$

$r_t = [0.8, 0.2, 0.1, 0.9]$

90%

power conflict tragedy revenge

$h_{t-1} \otimes r_t = [0.48, 0.12, 0.07, 0.09]$

$\tilde{h}_t = \tanh(W_c[h_{t-1} \otimes r_t, x_t] + b_c)$

Candidate memory

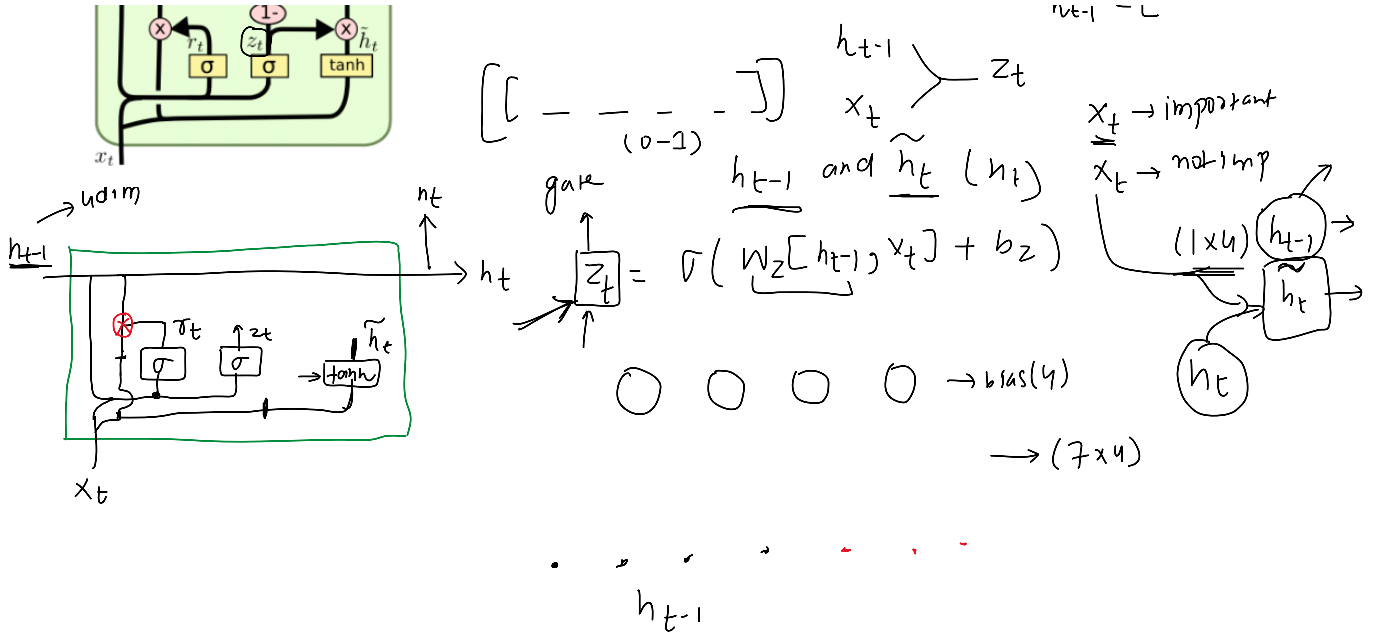
$\rightarrow b_c$

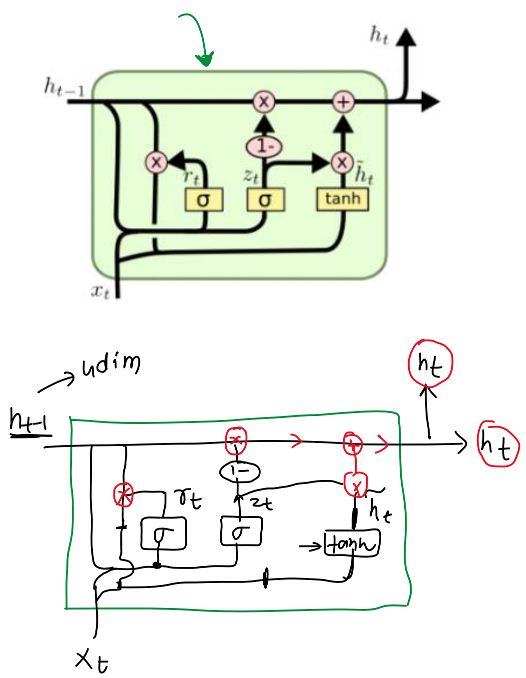
$\rightarrow W_c$ (7×4)

$(h_{t-1} \otimes r_t)$

$\tilde{h}_t = [0.7\ 0.2\ 0.1\ 0.2]$

$h_{t-1} = [0.6\ 0.6\ 0.7\ 0.1]$





$\tilde{h}_t = [0.7 \;\; 0.2 \;\; 0.1 \;\; 0.2]$

$h_{t-1} = [0.6 \;\; 0.6 \;\; 0.7 \;\; 0.1]$

$z_t = [0.1 \;\; 0.7 \;\; 0.8 \;\; 0.2]$

$h_t \; [0.07 \;\; 0.14 \;\; 0.08 \;\; 0.04]$

$$h_t = (1-z_t) \otimes h_{t-1} \oplus z_t \otimes \tilde{h}_t$$

This equation decides like what to keep from past and current input .

$(1-z_t) \to 0.9$

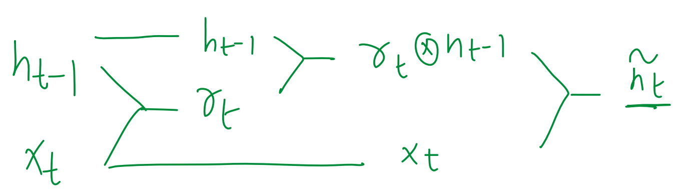
$(1-z_t) \; [0.9 \;\; 0.3 \;\; 0.2 \;\; 0.8] \; [0.6 \;\; 0.6 \;\; 0.7 \;\; 0.1]$

$[0.54 \;\; 0.18 \;\; 0.14 \;\; 0.08]$

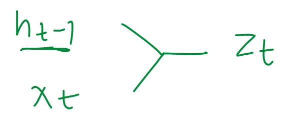
$\to h_t \; [0.61 \;\; 0.32 \;\; 0.22 \;\; 0.12]$

power conflict tragedy revenge

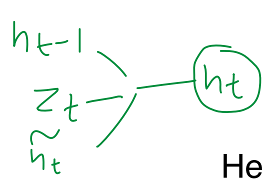
What rt is doing it is reseting the previous hidden state .



ht bar , Potential hidden state , like it is influenced by input.



zt : like how much to keep from input (you can think of in this direction.



Here it is updating the hidden state

LSTM vs GRU

05 October 2023 16:45

Here are the main differences between LSTM and GRU:

1. Number of Gates:
   - LSTM: Has three gates — input (or update) gate, forget gate, and output gate.
   - GRU: Has two gates — reset gate and update gate.
2. Memory Units:
   - LSTM: Uses two separate states - the cell state (ct) and the hidden state (ht). The cell state acts as an "internal memory" and is crucial for carrying long-term dependencies.
   - GRU: Simplifies this by using a single hidden state (ht) to both capture and output the memory.
3. Parameter Count:
   - LSTM: Generally has more parameters than a GRU because of its additional gate and separate cell state. For an input size of d and a hidden size of h, the LSTM has 4 ×((d×h)+(h×h)+h)4×((d×h)+(h×h)+h)parameters.
   - GRU: Has fewer parameters. For the same sizes, the GRU has 3×((d×h)+(h×h)+h)3 ×((d×h)+(h×h)+h)parameters.
4. Computational Complexity:
   - LSTM: Due to the extra gate and cell state, LSTMs are typically more computationally intensive than GRUs.
   - GRU: Is simpler and can be faster to compute, especially on smaller datasets or when computational resources are limited.
5. Empirical Performance:
   - LSTM: In many tasks, especially more complex ones, LSTMs have been observed to perform slightly better than GRUs.
   - GRU: Can perform comparably to LSTMs on certain tasks, especially when data is limited or tasks are simpler. They can also train faster due to fewer parameters.
6. Choice in Practice:
   - The choice between LSTM and GRU often comes down to empirical testing. Depending on the dataset and task, one might outperform the other. However, GRUs, due to their simplicity, are often the first choice when starting out.