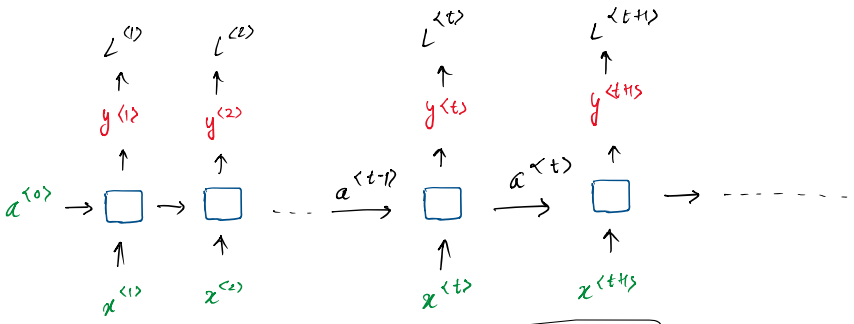
2) GRU

10 December 2025 01:51 PM



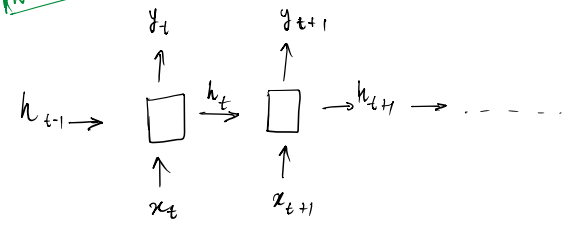
where,

$$a^{\langle t \rangle} = g_1(x^{\langle t \rangle} \cdot w_x + a^{\langle t-1 \rangle} \cdot w_a + b)$$

$$y^{\langle t \rangle} = g_2(a^{\langle t \rangle} \cdot w_y + b_y)$$

$$L(\hat{y}, y) = \sum_{i=0}^{t} L^{\langle i \rangle} \; ; \quad L^{\langle t \rangle} = L(y^{\langle t \rangle}, \hat{y})$$

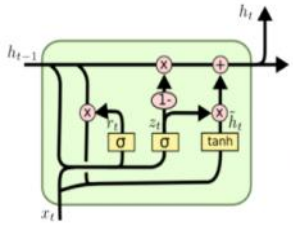
RNN's Architecture



$$h_t = \tanh(x_t \cdot w_x + h_{t-1} \cdot w_h + b_h)$$

$$y_t = h_t \cdot w_y + b_y$$

GRU's Architecture



$$r_t = \sigma(W_r x_t + U_r h_{t-1})$$

$$\tilde{h}_t = \tanh(W x_t + U(r_t \odot h_{t-1}))$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1})$$

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

↳ element wise multiplication

GRU's

reset Gate → Sigmoid function.

$$r_t = \sigma(x_t \cdot w_{r_x} + h_{r-1} \cdot w_{r_h} + b_{r_h})$$

↳ matrix multiplication

Update Gate

$$z_t = \sigma(x_t \cdot w_{z_x} + h_{r-1} \cdot w_{z_h} + b_{z_h})$$

↳ Decides how much of h_{r-1} to keep and how much of h_r to keep and how much to update the memory.

Candidate Activation → element wise multiplication.

$$\tilde{h}_t = x_t \cdot w_x + (r_t * h_{t-1}) \cdot w_h + b_h$$

↳ $\tilde{h}_t$ is just a proposal of how much of h_{t-1} is relevant to current input

final hiddenstate

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

V

final hidden state

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

↳ final memory (hidden state) to pass to next time step