$$h_t = f(h_{t-1}, x_t)$$
$$= f(f(h_{t-2}, x_{t-1}), x_t)$$

$$f = (\tanh(U x_t + \omega h_{t-1}))$$

Problem

(1) A very big sentence, answer question
forget stuff
carry stuff

$$h_{t+1} = \tanh(b + \omega h^t + u x^{(t)})$$

$$\frac{\partial L}{\partial h_{t+1}} \quad \omega \quad C$$

$$\left[\because \frac{\partial L}{\partial h_t} = \frac{\partial L}{\partial h_{t+1}} \frac{\partial h_{t+1}}{\partial h_t} \right]$$

$$= \frac{\partial L}{\partial h_{t+1}} \omega \boxed{}$$

$$\frac{\partial L}{\partial h_1} = \frac{\partial L}{\partial h_2} \omega \boxed{} = \frac{\partial L}{\partial h_3} \omega^2 \boxed{}$$

$$\hookrightarrow \frac{\partial L}{\partial h_3} \omega \boxed{}$$

$$W = QAQ^T \Rightarrow W^k = QA^kQ^T$$

(positive semidefinite)

orthogonal matrices

$$A = \begin{bmatrix} \lambda_1 & 0 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 0 & \cdots & 0 \\ \vdots & & & & \\ 0 & 0 & 0 & & \lambda_k \end{bmatrix}$$

$$\left[\because W^2 = QA\boxed{Q^TQ}AQ^T = QA^2Q^T\right]$$

$$\frac{\partial L}{\partial h_1} = \frac{\partial L}{\partial h_k} \cdot QA^kQ$$

$\lambda_1 \quad \lambda_2 \quad \cdots \quad \lambda_k$

$0.01 \rightarrow \lambda_1$

$\lambda^{40} \Rightarrow 0.000\cdots01$

$\lambda = 2$

$\lambda^{40} \Rightarrow 2^{40}$

→ So gradients become very small (vanish)
very large (explode)

$$h_{t+1} = \tanh(W h_t + U x_{t+1})$$

$$\begin{tikzcd} \boxed{h_t} \arrow[r] \arrow[d, dash] & \boxed{h_{t+1}} \arrow[d, dash] \\ \boxed{x_t} & \boxed{x_{t+1}} \end{tikzcd}$$

Now we want to forget some stuff as well

But how?

$f: \sigma(b^f + U^f x_t + W^f h_{t-1})$

multiply it by h^{t-1}
Now we get how much we forget

Now we need a remember function

(But why? can't we just add $b + Ux^t + Wh^{t-1}$)

$g = b^g + U^g x + W^g h^{t-1}$

→ Since it's always good to get an idea of it apparently

→ we take this and multiply it with $(b + Ux^t + Wh^{t-1})$

$$f^{(t)} h^{(t-1)} + g^t (b + Ux^t + Wh^{t-1})$$

But thoda wait karle, there's a change now

$$f_i^{(t)} = \sigma\left(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h_j^{(t-1)}\right)$$

$$s_i^{(t)} = f_i^{(t)} s_i^{(t-1)} + g_i^{(t)} \sigma\left(b_i + \sum_j U_{i,j} x_j^{(t)} + \sum_j W_{i,j} h_j^{(t-1)}\right)$$

$$g_i^{(t)} = \sigma\left(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{(t-1)}\right)$$

$$h_i^{(t)} = \tanh\left(s_i^{(t)}\right) q_i^{(t)}$$

$$q_i^{(t)} = \sigma\left(b_i^o + \sum_j U_{i,j}^o x_j^{(t)} + \sum_j W_{i,j}^o h_j^{(t-1)}\right)$$

But here, we dont do $\tanh(b+(Ux_t+Wh_{t-1}))$ here, unlike RNN

we do

$$h^{t+1} = \tanh(s^t)\underbrace{(\sigma(b+Ux_t+wh_{t-1}))}_{\text{fraction of prev person also taken}}$$

now $s(t) =$

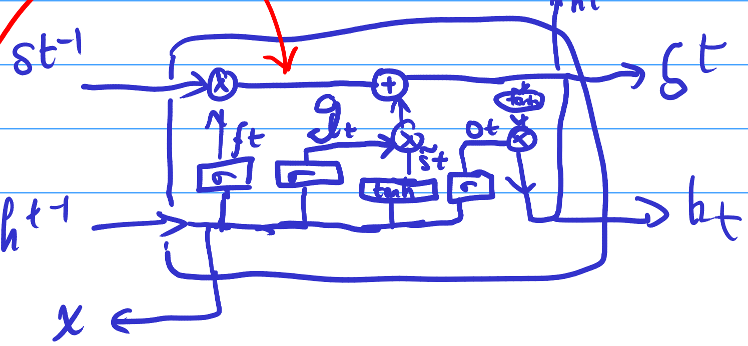
$$f^{(t)} s^{(t-1)} + g^{(t)} \sigma(b+Ux_t+Wh_{t-1})$$

So now, we need to find out

b^f U^f w^f $\quad$ b, ω, U, v $\quad$ $\underbrace{b^g \; U^g \; w^g}_{\text{fraction for remembering}}$

for decoding

Our backprop is now on 's' and is very simple!!



the derivative of the product will give a sum though instead of products

maybe thats why gradients dont explode

weighted average

$$h_i^{(t)} = u_i^{(t)} h_i^{(t-1)} + (1 - u_i^{(t-1)})\sigma\left(b_i + \sum_j U_{i,j} x_j^{(t-1)} + \sum_j W_{i,j} r_j^{(t-1)} h_j^{(t-1)}\right)$$

resets prev input

- Where ***u*** stands for the update gate and ***r*** for reset gate. Their value is defined as usual:

$$u_i^{(t)} = \sigma\left(b_i^u + \sum_j U_{i,j}^u x_j^{(t)} + \sum_j W_{i,j}^u h_j^{(t)}\right)$$

and

$$r_i^{(t)} = \sigma\left(b_i^r + \sum_j U_{i,j}^r x_j^{(t)} + \sum_j W_{i,j}^r h_j^{(t)}\right)$$

GRU