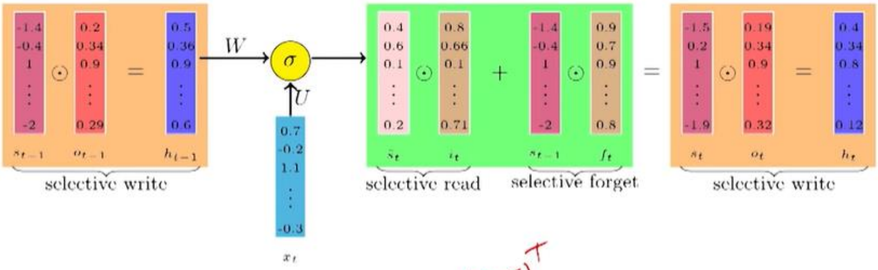
How LSTMs solve vanishing gradients problem



$(0.5)^T$

Intuition

- During forward propagation the gates control the flow of information
- They prevent any irrelevant information from being written to the state
- Similarly during backward propagation they control the flow of gradients

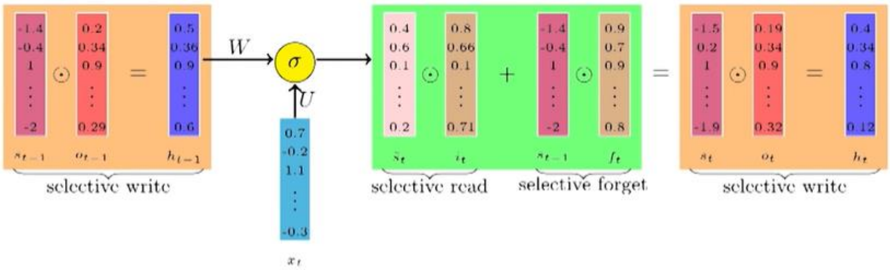
s_1 $\quad f_1 \ f_2 \ f_3 \ \dots \ f_T$ $\quad s_T$

$0.5 \ 0.5 \ 0.5 \quad 0.5 \quad \frac{\partial s_t}{\partial s_{t-1}}$

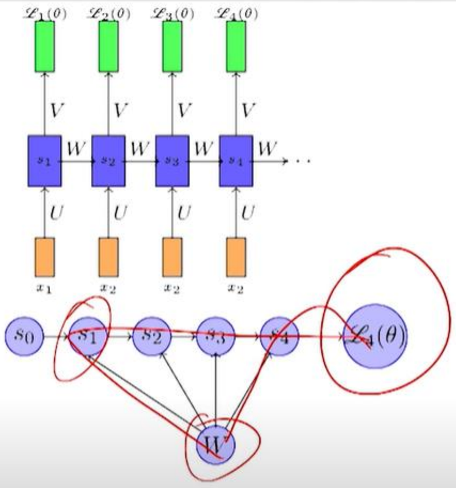
$s_t = f_t \odot s_{t-1} + i_t \odot \tilde{s}_t$

$0.5 \quad f_t$

$(0.5)^T$



- If the state at time $t-1$ did not contribute much to the state at time t (i.e., if $\|f_t\| \to 0$ and $\|o_{t-1}\| \to 0$) then during backpropagation the gradients flowing into s_{t-1} will vanish
- But this kind of a vanishing gradient is fine (since s_{t-1} did not contribute to s_t we don't want to hold it responsible for the crimes of s_t)
- The key difference from vanilla RNNs is that the flow of information and gradients is controlled by the gates which ensure that the gradients vanish only when they should (i.e., when s_{t-1} didn't contribute much to s_t)

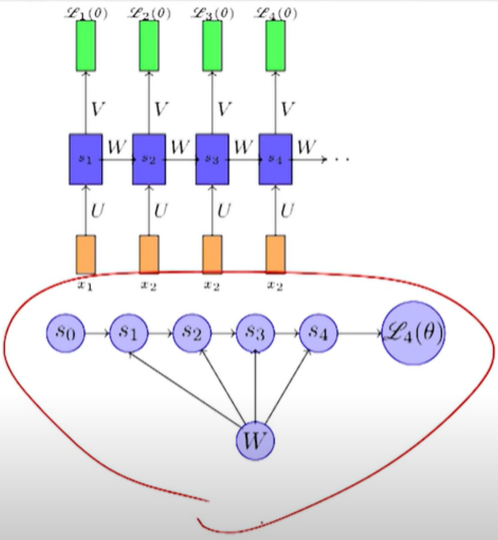


- Recall that RNNs had this multiplicative term which caused the gradients to vanish

$$\frac{\partial \mathscr{L}_t(\theta)}{\partial W} = \frac{\partial \mathscr{L}_t(\theta)}{\partial s_t} \sum_{k=1}^{t} \prod_{j=k}^{t-1} \frac{\partial s_{j+1}}{\partial s_j} \frac{\partial^+ s_k}{\partial W}$$

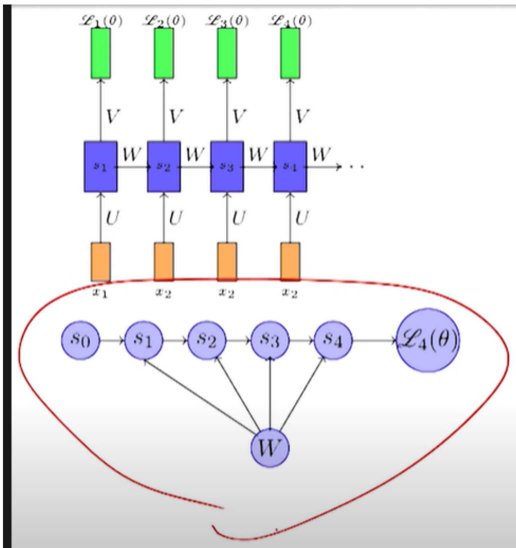
- In particular, if the loss at $\mathscr{L}_4(\theta)$ was high because W was not good enough to compute s_1 correctly then this information will not be propagated back to W as the gradient $\frac{\partial \mathscr{L}_t(\theta)}{\partial W}$ along this long path will vanish

4:41 / 23:52

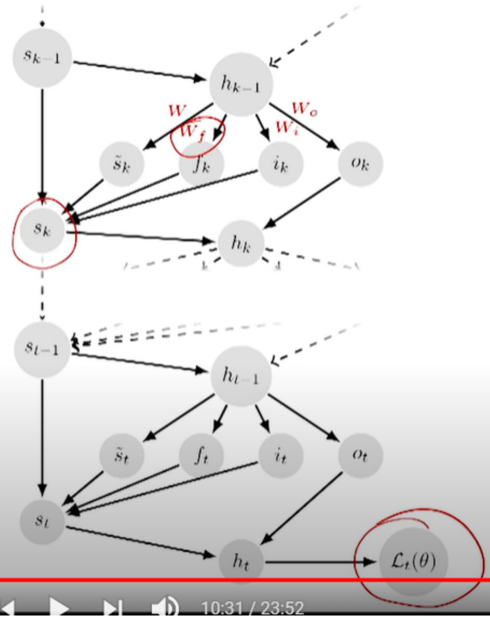


- In general, the gradient of $\mathscr{L}_t(\theta)$ w.r.t. θ_i vanishes when the gradients flowing through **each and every path** from $L_t(\theta)$ to θ_i vanish.
- On the other hand, the gradient of $\mathscr{L}_t(\theta)$ w.r.t. θ_i explodes when the gradient flowing through **at least one path** explodes.
- We will first argue that in the case of LSTMs there exists at least one path through which the gradients can flow effectively (and hence no vanishing gradients)

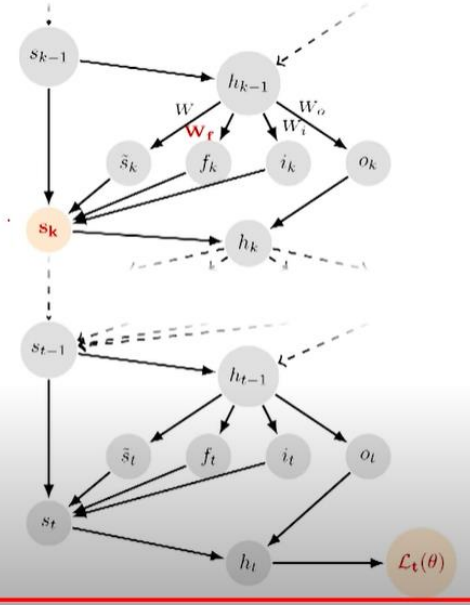
6:55 / 23:52



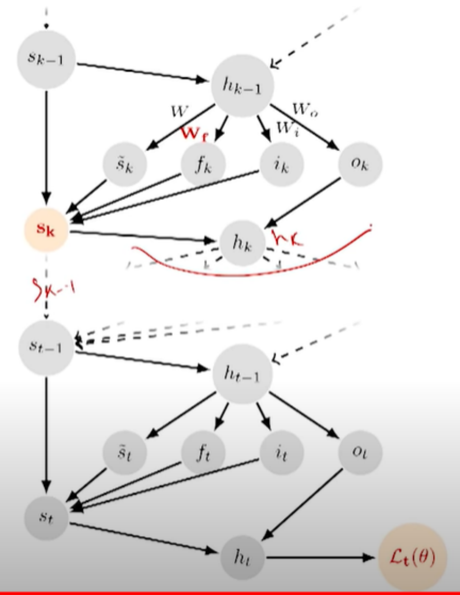
- In general, the gradient of $\mathscr{L}_t(\theta)$ w.r.t. θ_i vanishes when the gradients flowing through **each and every path** from $L_t(\theta)$ to θ_i vanish.
- On the other hand, the gradient of $\mathscr{L}_t(\theta)$ w.r.t. θ_i explodes when the gradient flowing through **at least one path** explodes.
- We will first argue that in the case of LSTMs there exists at least one path through which the gradients can flow effectively (and hence no vanishing gradients)



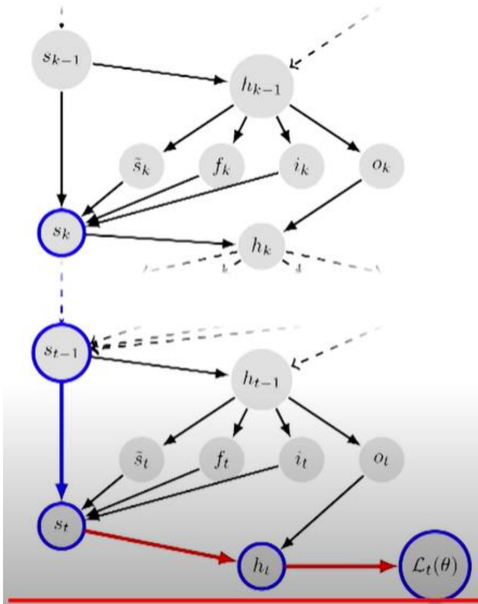
- Starting from h_{k-1} and s_{k-1} we have reached h_k and s_k
- And the recursion will now continue till the last timestep
- For simplicity and ease of illustration, instead of considering the parameters (W, W_o, W_i, W_f, U, U_o, U_i, U_f) as separate nodes in the graph we will just put them on the appropriate edges. (We show only a few parameters and not all)
- We are now interested in knowing if the gradient from $\mathscr{L}_t(\theta)$ flows back to an arbitrary timestep k



- For example, we are interested in knowing if the gradient flows to W_f through s_k
- In other words, if $\mathscr{L}_t(\theta)$ was high because W_f failed to compute an appropriate value for s_k then this information should flow back to W_f through the gradients
- We can ask a similar question about the other parameters (for example, W_i, W_o, W, etc.)
- How does LSTM ensure that this gradient does not vanish even at arbitrary time steps? Let us see



- It is sufficient to show that $\frac{\partial \mathscr{L}_t(\theta)}{\partial s_k}$ does not vanish (because if this does not vanish we can reach W_f through s_k)
- First, we observe that there are multiple paths from $\mathscr{L}_t(\theta)$ to s_k (you just need to reverse the direction of the arrows for backpropagation)
- For example, there is one path through s_{k+1}, another through h_k
- Further, there are multiple paths to reach to h_k itself (as should be obvious from the number of outgoing arrows from h_k)
- So at this point just convince yourself that there are many paths from $\mathscr{L}_t(\theta)$ to s_k

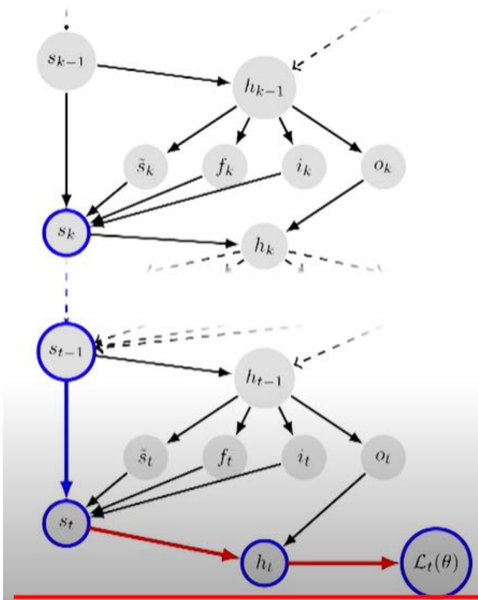


- Consider one such path (highlighted) which will contribute to the gradient
- Let us denote the gradient along this path as t_0

$$t_0 = \frac{\partial \mathscr{L}_t(\theta)}{\partial h_t} \frac{\partial h_t}{\partial s_t} \frac{\partial s_t}{\partial s_{t-1}} \cdots \frac{\partial s_{k+1}}{\partial s_k}$$

- The first term $\frac{\partial \mathscr{L}_t(\theta)}{\partial h_t}$ is fine and it doesn't vanish (h_t is directly connected to $\mathscr{L}_t(\theta)$ and there are no intermediate nodes which can cause the gradient to vanish)
- We will now look at the other terms $\frac{\partial h_t}{\partial s_t}$ $\frac{\partial s_t}{\partial s_{t-1}}$ $(\forall t)$

13:56 / 23:52

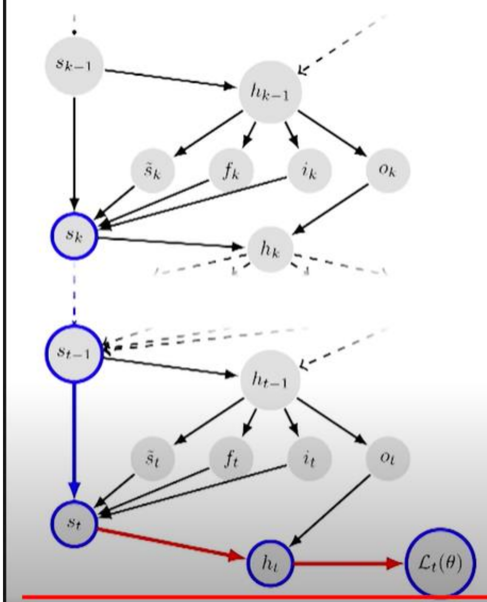


- Let us first look at $\frac{\partial h_t}{\partial s_t}$
- Recall that

$$h_t = o_t \odot \sigma(s_t)$$

- Note that h_{ti} only depends on o_{ti} and s_{ti} and not on any other elements of o_t and s_t
- $\frac{\partial h_t}{\partial s_t}$ will thus be a square diagonal matrix $\in \mathbb{R}^{d \times d}$ whose diagonal will be $o_t \odot \sigma'(s_t) \in \mathbb{R}^d$ (see slide 35 of Lecture 14)
- We will represent this diagonal matrix by $\mathcal{D}(o_t \odot \sigma'(s_t))$

15:56 / 23:52

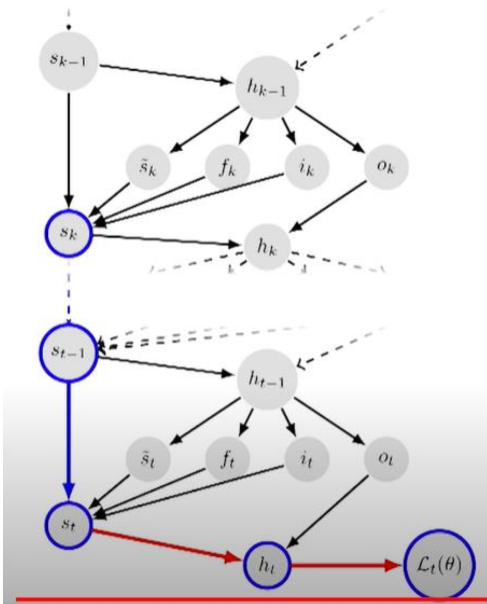


- Now let us consider $\frac{\partial s_t}{\partial s_{t-1}}$
- Recall that

$$s_t = f_t \odot s_{t-1} + i_t \odot \tilde{s}_t$$

- Notice that $\tilde{s}_t$ also depends on s_{t-1} so we cannot treat it as a constant
- So once again we are dealing with an ordered network and thus $\frac{\partial s_t}{\partial s_{t-1}}$ will be a sum of an explicit term and an implicit term (see slide 37 from Lecture 14)
- For simplicity, let us assume that the gradient from the implicit term vanishes (we are assuming a worst case scenario)
- And the gradient from the explicit term (treating $\tilde{s}_t$ as a constant) is given by $\mathcal{D}(f_t)$

17:40 / 23:52

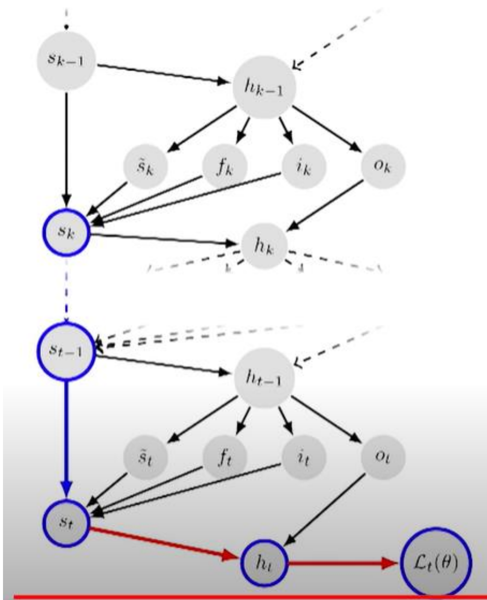


- We now return back to our full expression for t_0:

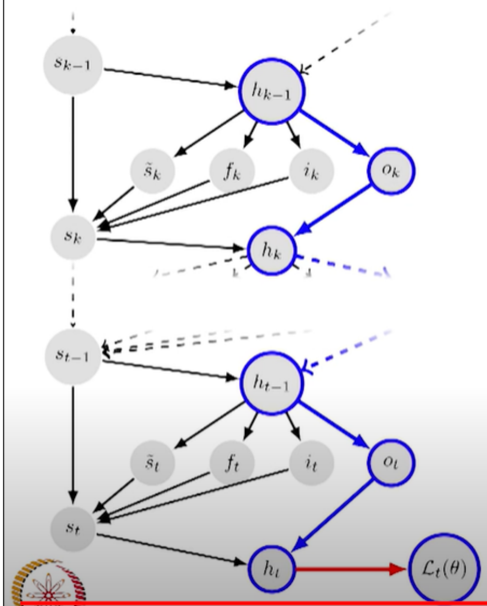
$$\begin{aligned} t_0 &= \frac{\partial \mathcal{L}_t(\theta)}{\partial h_t} \frac{\partial h_t}{\partial s_t} \frac{\partial s_t}{\partial s_{t-1}} \dots \frac{\partial s_{k+1}}{\partial s_k} \\ &= \mathcal{L}'_t(h_t).\mathcal{D}(o_t \odot \sigma'(s_t))\mathcal{D}(f_t) \dots \mathcal{D}(f_{k+1}) \\ &= \mathcal{L}'_t(h_t).\mathcal{D}(o_t \odot \sigma'(s_t))\mathcal{D}(f_t \odot \dots \odot f_{k+1}) \\ &= \mathcal{L}'_t(h_t).\mathcal{D}(o_t \odot \sigma'(s_t))\mathcal{D}(\odot_{i=k+1}^{t} f_i) \end{aligned}$$

- The red terms don't vanish and the blue terms contain a multiplication of the forget gates
- The forget gates thus regulate the gradient flow depending on the explicit contribution of a state (s_t) to the next state s_{t+1}

19:40 / 23:52



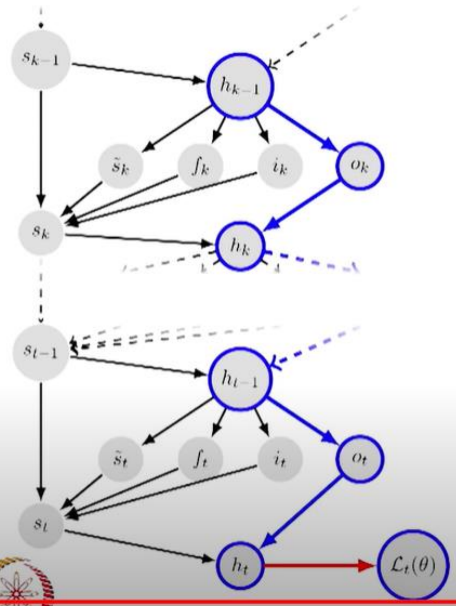
- If during forward pass s_t did not contribute much to s_{t+1} (because $f_t \to 0$) then during backpropgation also the gradient will not reach s_t
- This is fine because if s_t did not contribute much to s_{t+1} then there is no reason to hold it responsible during backpropgation (f_t does the same regulation during forward pass and backward pass which is fair)
- Thus there exists this one path along which the gradient doesn't vanish when it shouldn't
- And as argued as long as the gradient flows back to W_f through one of the paths (t_0) through s_k we are fine !
- Of course the gradient flows back only when required as regulated by f_i's (but let me just say it one last time that *this is fair*)



- Now we will see why LSTMs do not solve the problem of exploding gradients
- We will show a path through which the gradient can explode
- Let us compute one term (say t_1) of $\frac{\partial \mathcal{L}_t(\theta)}{\partial h_{k-1}}$ corresponding to the highlighted path

$$t_1 = \frac{\partial \mathcal{L}_t(\theta)}{\partial h_t}\left(\frac{\partial h_t}{\partial o_t}\frac{\partial o_t}{\partial h_{t-1}}\right)\dots\left(\frac{\partial h_k}{\partial o_k}\frac{\partial o_k}{\partial h_{k-1}}\right)$$
$$= \mathcal{L}_t'(h_t)\,(\mathcal{D}(\sigma(s_t)\odot o_t').W_o)\dots$$
$$(\mathcal{D}(\sigma(s_k)\odot o_k').W_o)$$
$$\|t_1\| \leq \|\mathcal{L}_t'(h_t)\|\,(\|K\|\|W_o\|)^{t-k+1}$$

- Depending on the norm of matrix W_o, the gradient $\frac{\partial \mathcal{L}_t(\theta)}{\partial h_{k-1}}$ may explode



- So how do we deal with the problem of exploding gradients ?
- One popular trick is to use gradient clipping
- While backpropagating if the norm of the gradient exceeds a certain value, it is scaled to keep its norm within an acceptable threshold*

*Pascanu, Razvan, Tomas Mikolov, and Yoshua Bengio. "On the difficulty of training recurrent neural networks." ICML(3)28(2013):1310-1318