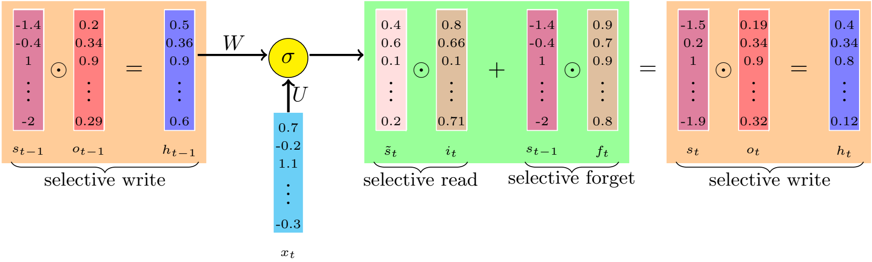
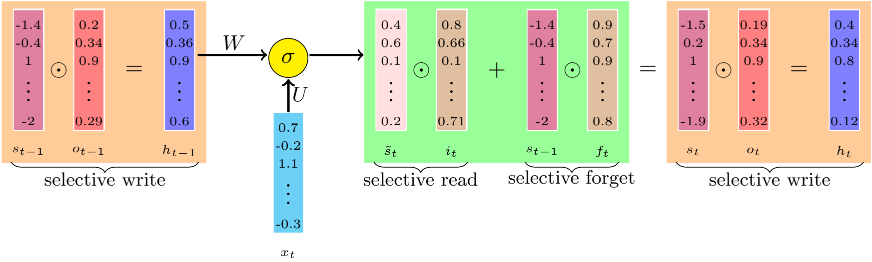
Module 15.3: How LSTMs avoid the problem of vanishing gradients



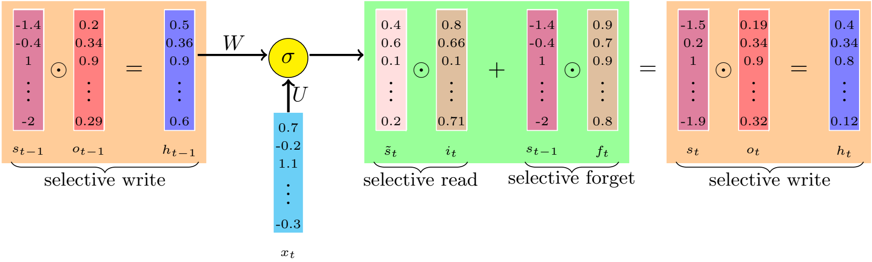
Intuition

- During forward propagation the gates control the flow of information



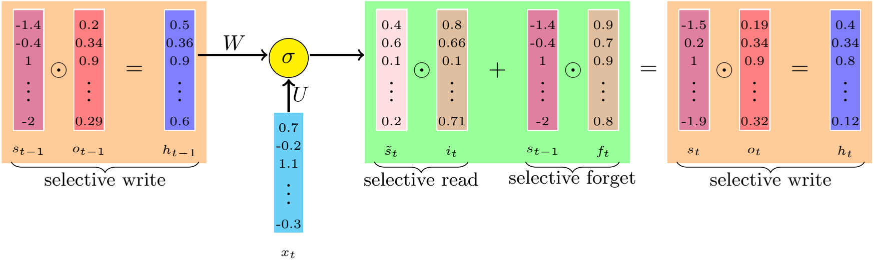
Intuition

- During forward propagation the gates control the flow of information
- They prevent any irrelevant information from being written to the state



Intuition

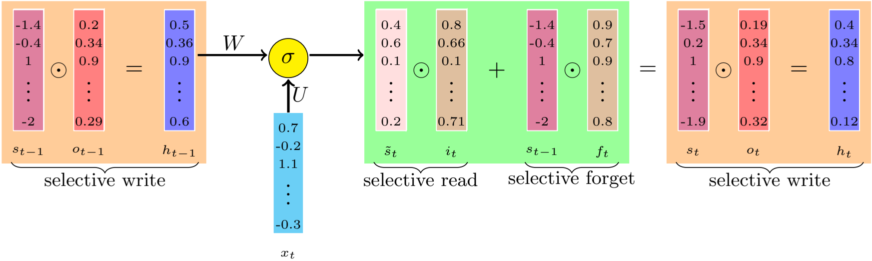
- During forward propagation the gates control the flow of information
- They prevent any irrelevant information from being written to the state
- Similarly during backward propagation they control the flow of gradients



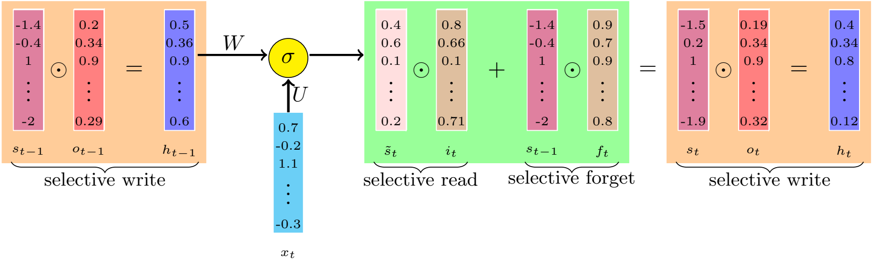
Intuition

- During forward propagation the gates control the flow of information
- They prevent any irrelevant information from being written to the state

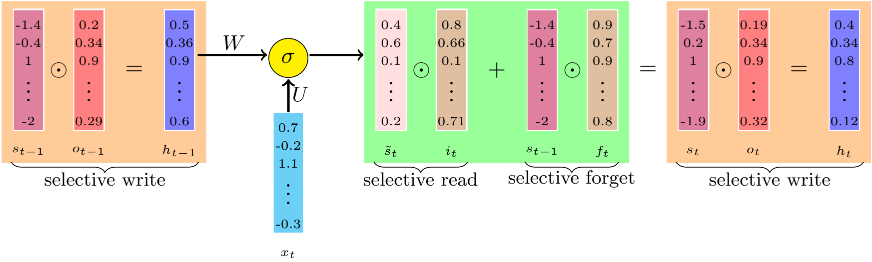
- Similarly during backward propagation they control the flow of gradients
- It is easy to see that during backward pass the gradients will get multiplied by the gate



- 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

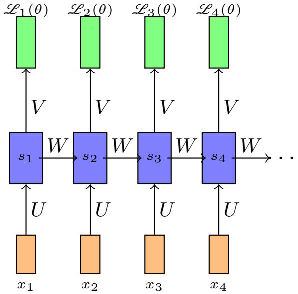


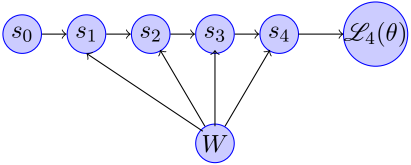
- If the state at time $t-1$ did not contribute much to the state at time t (i.e., if $\|f_t\| \to 0$ and $\|o_{t-1}\| \to 0$) then during backpropagation the gradients flowing into s_{t-1} will vanish
- But this kind of a vanishing gradient is fine (since s_{t-1} did not contribute to s_t we don't want to hold it responsible for the crimes of s_t)



- 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)

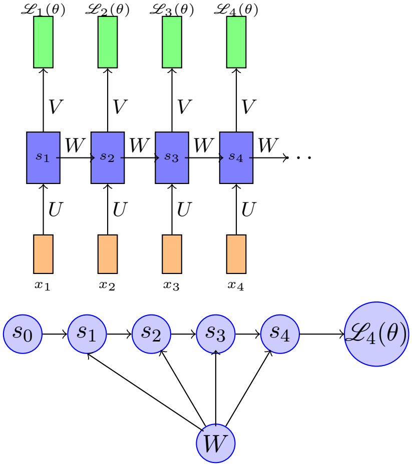
We will now see an illustrative proof of how the gates control the flow of gradients





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

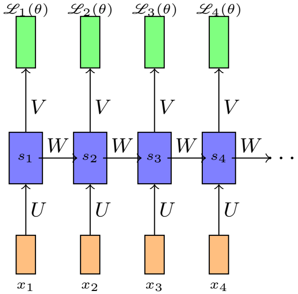
$$\frac{\partial \mathscr{L}_t(\theta)}{\partial W} = \frac{\partial \mathscr{L}_t(\theta)}{\partial s_t} \sum_{k=1}^{t} \color{red}{\prod_{j=k}^{t-1} \frac{\partial s_{j+1}}{\partial s_j}} \color{black}{\frac{\partial^+ s_k}{\partial W}}$$

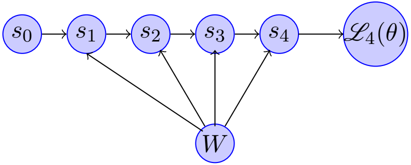


- Recall that RNNs had this multiplicative term which caused the gradients to vanish

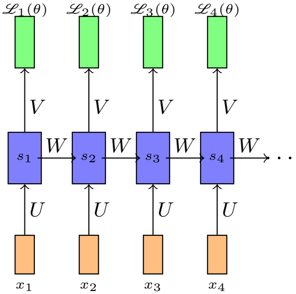
$$\frac{\partial \mathscr{L}_t(\theta)}{\partial W} = \frac{\partial \mathscr{L}_t(\theta)}{\partial s_t} \sum_{k=1}^{t} \prod_{j=k}^{t-1} \frac{\partial s_{j+1}}{\partial s_j} \frac{\partial^+ s_k}{\partial W}$$

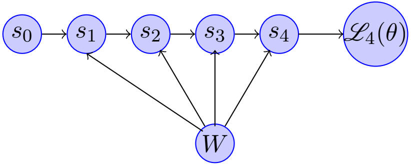
- In particular, if the loss at $\mathscr{L}_4(\theta)$ was high because W was not good enough to compute s_1 correctly then this information will not be propagated back to W as the gradient $\frac{\partial \mathscr{L}_t(\theta)}{\partial W}$ along this long path will vanish



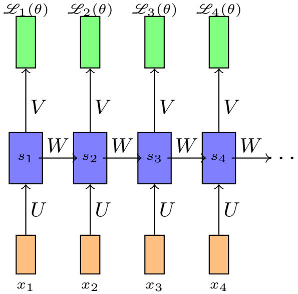


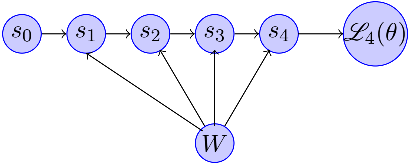
- In general, the gradient of $\mathscr{L}_t(\theta)$ w.r.t. θ_i vanishes when the gradients flowing through **each and every path** from $L_t(\theta)$ to θ_i vanish.





- 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.



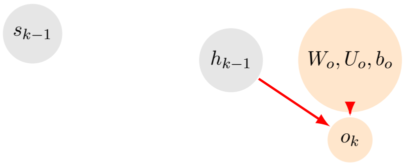


- In general, the gradient of $\mathscr{L}_t(\theta)$ w.r.t. θ_i vanishes when the gradients flowing through **each and every path** from $L_t(\theta)$ to θ_i vanish.
- On the other hand, the gradient of $\mathscr{L}_t(\theta)$ w.r.t. θ_i explodes when the gradient flowing through **at least one path** 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)

- We will start with the dependency graph involving different variables in LSTMs

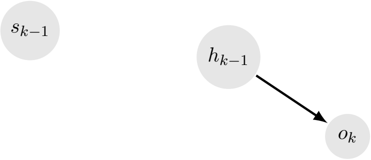
- We will start with the dependency graph involving different variables in LSTMs
- Starting with the states at timestep $k-1$

- We will start with the dependency graph involving different variables in LSTMs
- Starting with the states at timestep $k-1$



- We will start with the dependency graph involving different variables in LSTMs
- Starting with the states at timestep $k-1$

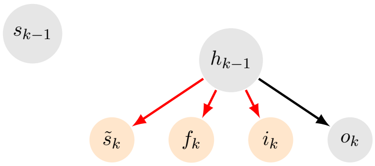
$$o_k = \sigma(W_o h_{k-1} + U_o x_k + b_o)$$



- We will start with the dependency graph involving different variables in LSTMs
- Starting with the states at timestep $k-1$

$$o_k = \sigma(W_o h_{k-1} + U_o x_k + b_o)$$

- For simplicity we will omit the parameters for now and return back to them later

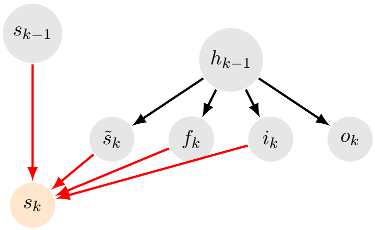


- We will start with the dependency graph involving different variables in LSTMs
- Starting with the states at timestep $k-1$

$$o_k = \sigma(W_o h_{k-1} + U_o x_k + b_o)$$

- For simplicity we will omit the parameters for now and return back to them later

$$i_k = \sigma(W_i h_{k-1} + U_i x_k + b_i)$$
$$f_k = \sigma(W_f h_{k-1} + U_f x_k + b_f)$$
$$\tilde{s_k} = \sigma(W h_{k-1} + U x_k + b)$$

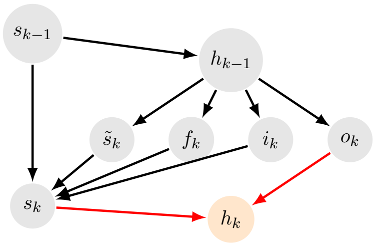


- We will start with the dependency graph involving different variables in LSTMs
- Starting with the states at timestep $k-1$

$$o_k = \sigma(W_o h_{k-1} + U_o x_k + b_o)$$

- For simplicity we will omit the parameters for now and return back to them later

$$i_k = \sigma(W_i h_{k-1} + U_i x_k + b_i)$$
$$f_k = \sigma(W_f h_{k-1} + U_f x_k + b_f)$$
$$\tilde{s_k} = \sigma(W h_{k-1} + U x_k + b)$$
$$s_k = f_k \odot s_{k-1} + i_k \odot \tilde{s_k}$$

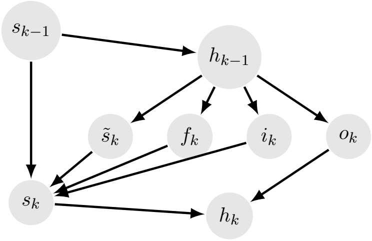


- We will start with the dependency graph involving different variables in LSTMs
- Starting with the states at timestep $k-1$

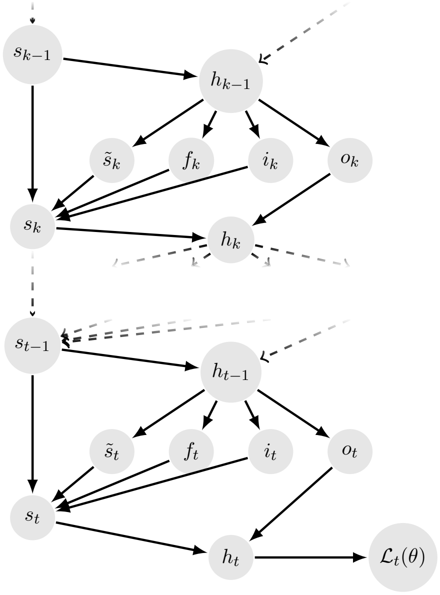
$$o_k = \sigma(W_o h_{k-1} + U_o x_k + b_o)$$

- For simplicity we will omit the parameters for now and return back to them later

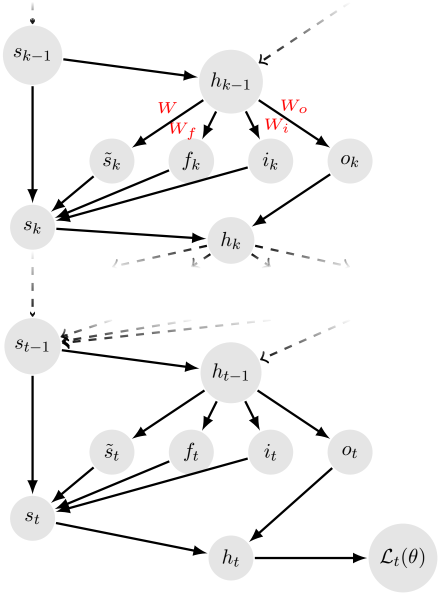
$$i_k = \sigma(W_i h_{k-1} + U_i x_k + b_i)$$
$$f_k = \sigma(W_f h_{k-1} + U_f x_k + b_f)$$
$$\tilde{s_k} = \sigma(W h_{k-1} + U x_k + b)$$
$$s_k = f_k \odot s_{k-1} + i_k \odot \tilde{s_k}$$
$$h_k = o_k \odot \sigma(s_k)$$



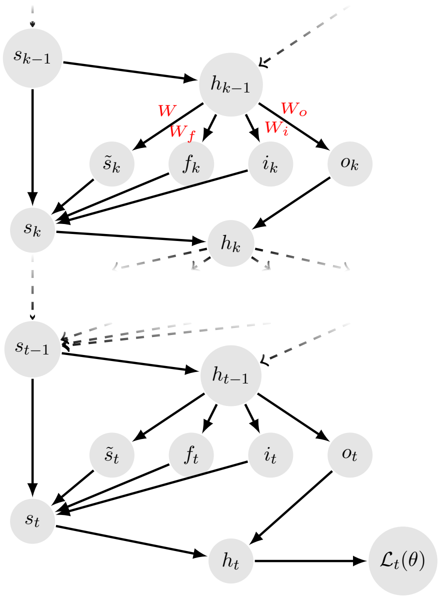
- Starting from h_{k-1} and s_{k-1} we have reached h_k and s_k



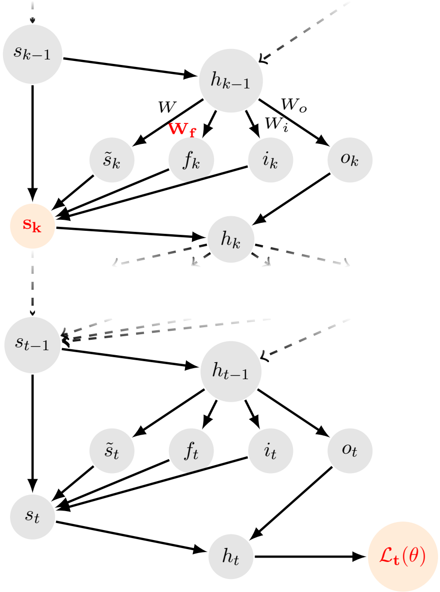
- Starting from h_{k-1} and s_{k-1} we have reached h_k and s_k
- And the recursion will now continue till the last timestep



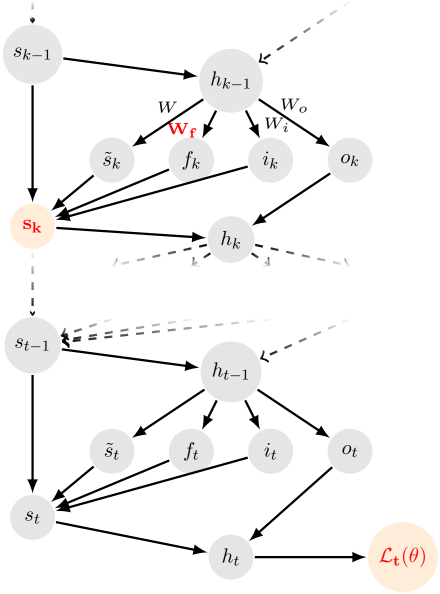
- Starting from h_{k-1} and s_{k-1} we have reached h_k and s_k
- And the recursion will now continue till the last timestep
- For simplicity and ease of illustration, instead of considering the parameters (W, W_o, W_i, W_f, U, U_o, U_i, U_f) as separate nodes in the graph we will just put them on the appropriate edges. (We show only a few parameters and not all)



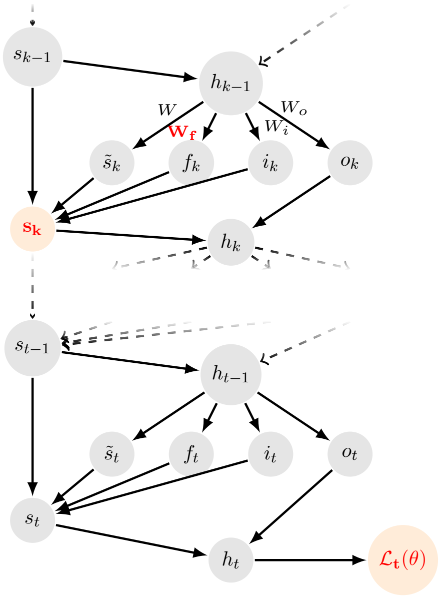
- Starting from h_{k-1} and s_{k-1} we have reached h_k and s_k
- And the recursion will now continue till the last timestep
- For simplicity and ease of illustration, instead of considering the parameters (W, W_o, W_i, W_f, U, U_o, U_i, U_f) as separate nodes in the graph we will just put them on the 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



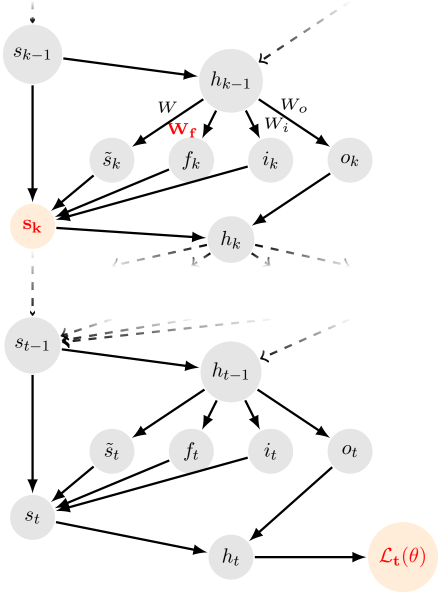
- For example, we are interested in knowing if the gradient flows to W_f through s_k



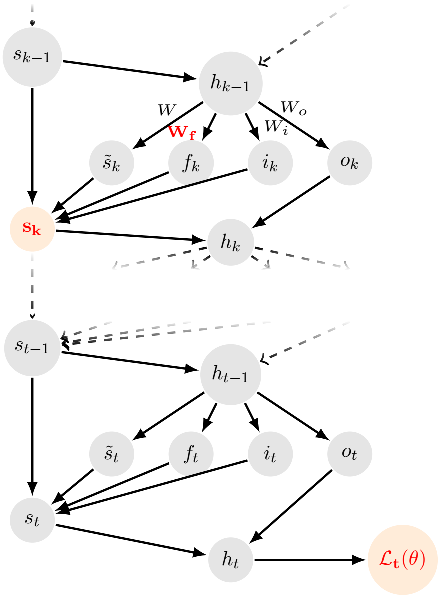
- For example, we are interested in knowing if the gradient flows to W_f through s_k
- In other words, if $\mathscr{L}_t(\theta)$ was high because W_f failed to compute an appropriate value for s_k then this information should flow back to W_f through the gradients



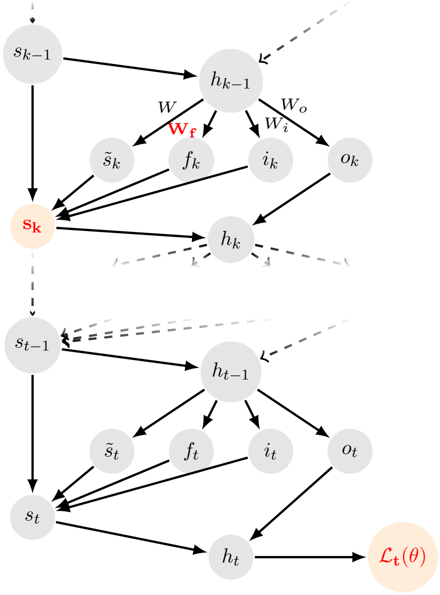
- For example, we are interested in knowing if the gradient flows to W_f through s_k
- In other words, if $\mathscr{L}_t(\theta)$ was high because W_f failed to compute an appropriate value for s_k then this information should flow back to W_f through the gradients
- We can ask a similar question about the other parameters (for example, W_i, W_o, W, etc.)



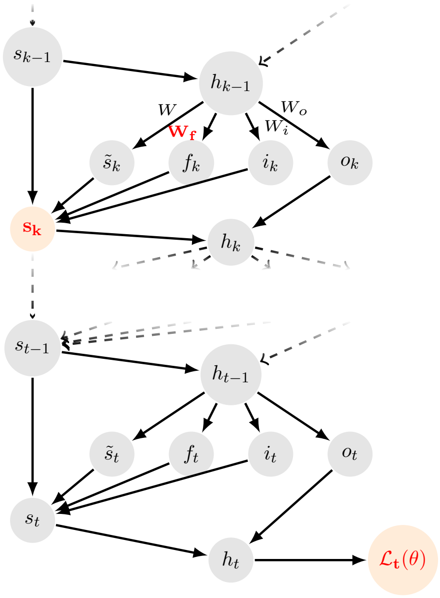
- For example, we are interested in knowing if the gradient flows to W_f through s_k
- In other words, if $\mathscr{L}_t(\theta)$ was high because W_f failed to compute an appropriate value for s_k then this information should flow back to W_f through the gradients
- We can ask a similar question about 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



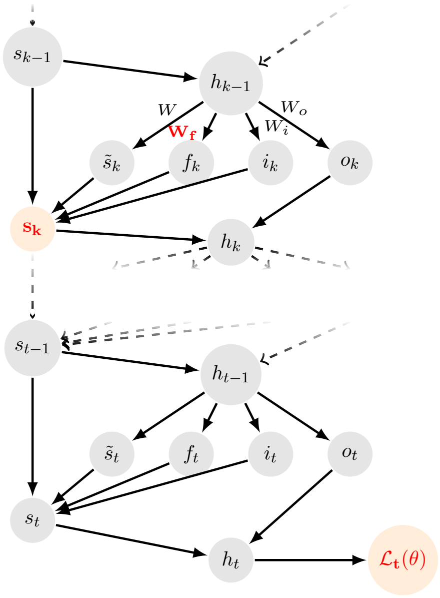
- It is sufficient to show that $\frac{\partial \mathcal{L}_t(\theta)}{\partial s_k}$ does not vanish (because if this does not vanish we can reach W_f through s_k)



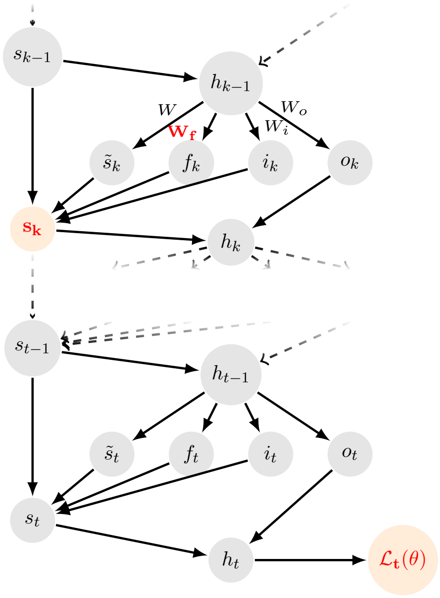
- It is sufficient to show that $\frac{\partial \mathcal{L}_t(\theta)}{\partial s_k}$ does not vanish (because if this does not vanish we can reach W_f through s_k)
- First, we observe that there are multiple paths from $\mathscr{L}_t(\theta)$ to s_k (you just need to reverse the direction of the arrows for backpropagation)



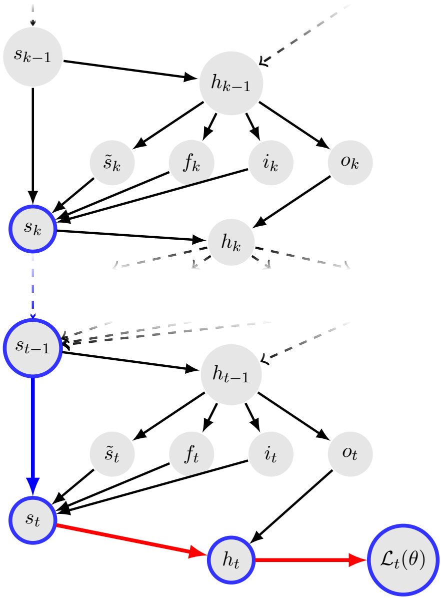
- It is sufficient to show that $\frac{\partial \mathscr{L}_t(\theta)}{\partial s_k}$ does not vanish (because if this does not vanish we can reach W_f through s_k)
- First, we observe that there are multiple paths from $\mathscr{L}_t(\theta)$ to s_k (you just need to reverse the direction of the arrows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



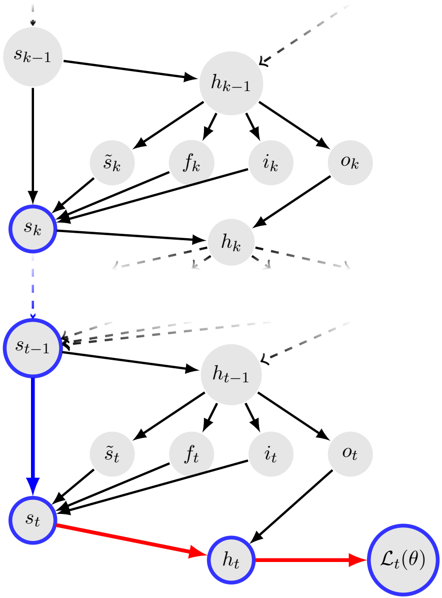
- It is sufficient to show that $\frac{\partial \mathcal{L}_t(\theta)}{\partial s_k}$ does not vanish (because if this does not vanish we can reach W_f through s_k)
- First, we observe that there are multiple paths from $\mathscr{L}_t(\theta)$ to s_k (you just need to reverse the direction of the arrows 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)



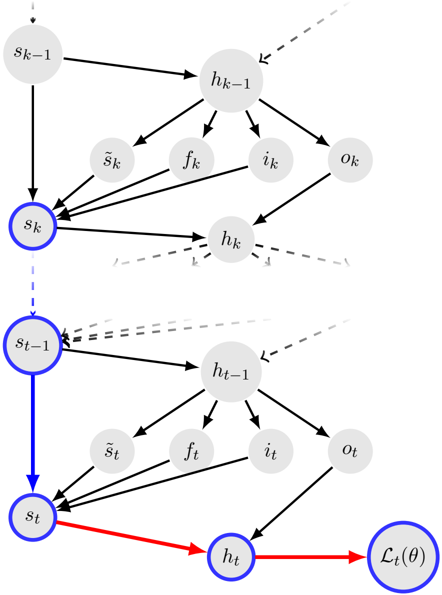
- It is sufficient to show that $\frac{\partial \mathscr{L}_t(\theta)}{\partial s_k}$ does not vanish (because if this does not vanish we can reach W_f through s_k)
- First, we observe that there are multiple paths from $\mathscr{L}_t(\theta)$ to s_k (you just need to reverse the direction of the arrows 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

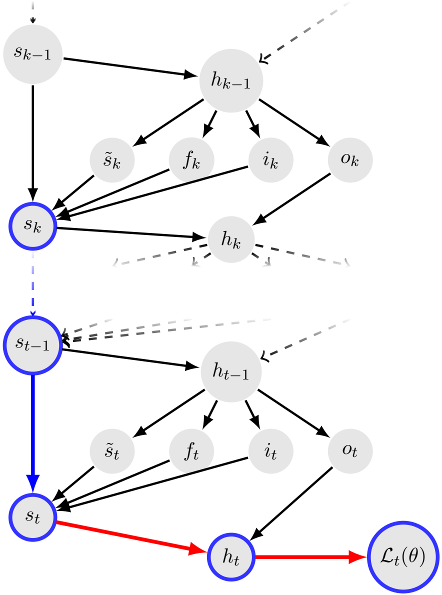


- Consider one such path (highlighted) which will contribute to the gradient
- Let us denote the gradient along this path as t_0



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

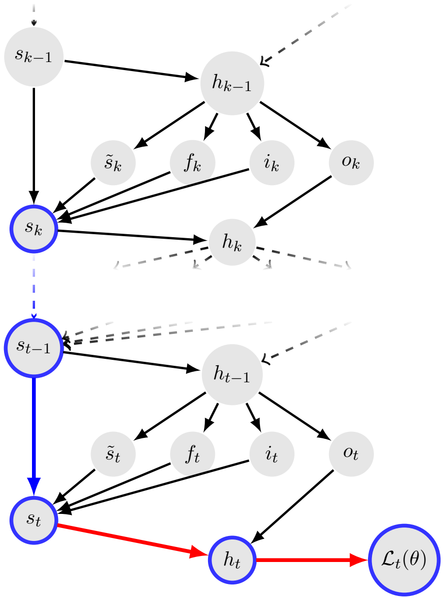
$$t_0 = \frac{\partial \mathscr{L}_t(\theta)}{\partial h_t} \frac{\partial h_t}{\partial s_t} \frac{\partial s_t}{\partial s_{t-1}} \cdots \frac{\partial s_{k+1}}{\partial s_k}$$



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

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

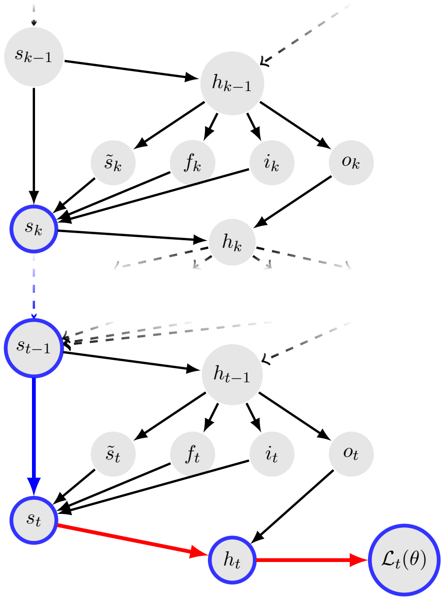
- The first term $\frac{\partial \mathscr{L}_t(\theta)}{\partial h_t}$ is fine and it doesn't vanish (h_t is directly connected to $\mathscr{L}_t(\theta)$ and there are no intermediate nodes which can cause the gradient to vanish)



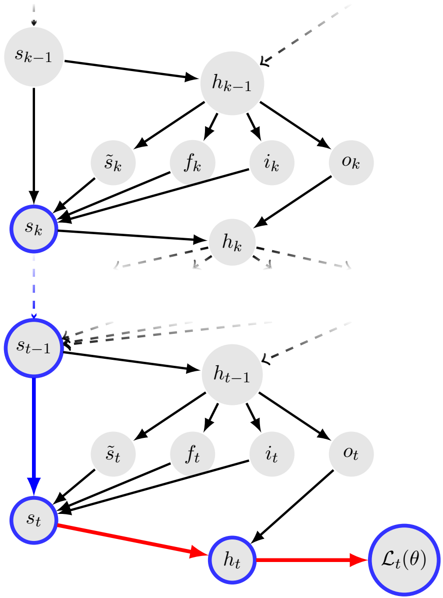
- Consider one such path (highlighted) which will contribute to the gradient
- Let us denote the gradient along this path as t_0

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

- 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 $\frac{\partial h_t}{\partial s_t}$ $\frac{\partial s_t}{\partial s_{t-1}}$ $(\forall t)$

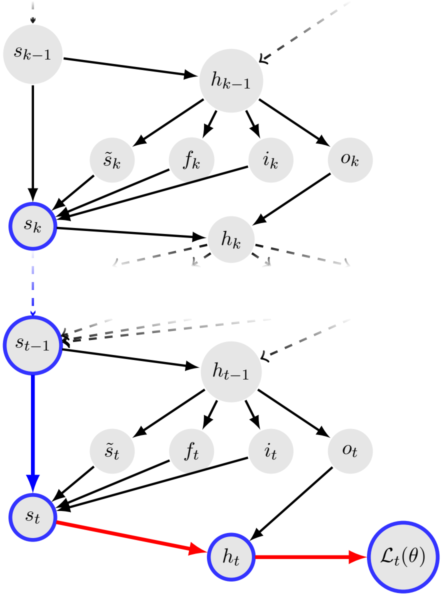


- Let us first look at $\frac{\partial h_t}{\partial s_t}$



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

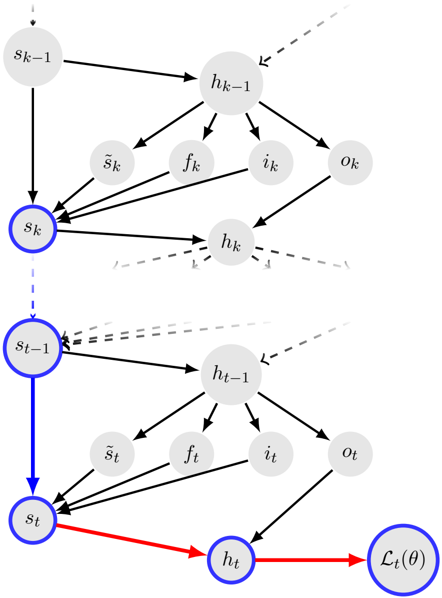
$$h_t = o_t \odot \sigma(s_t)$$



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

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

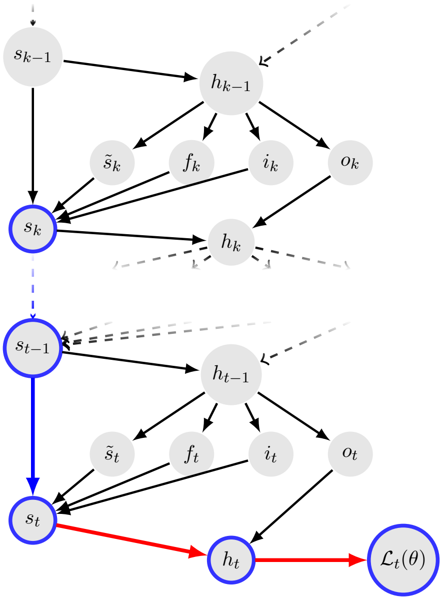
- Note that h_{ti} only depends on o_{ti} and s_{ti} and not on any other elements of o_t and s_t



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

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

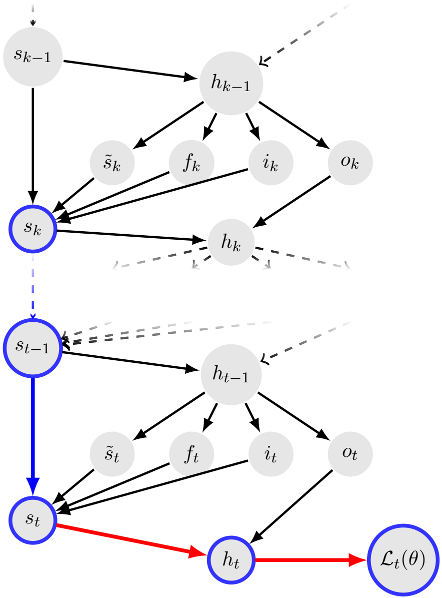
- Note that h_{ti} only depends on o_{ti} and s_{ti} and not on any other elements of o_t and s_t
- $\frac{\partial h_t}{\partial s_t}$ will thus be a square diagonal matrix $\in \mathbb{R}^{d \times d}$ whose diagonal will be $o_t \odot \sigma'(s_t) \in \mathbb{R}^d$ (see slide 35 of Lecture 14)



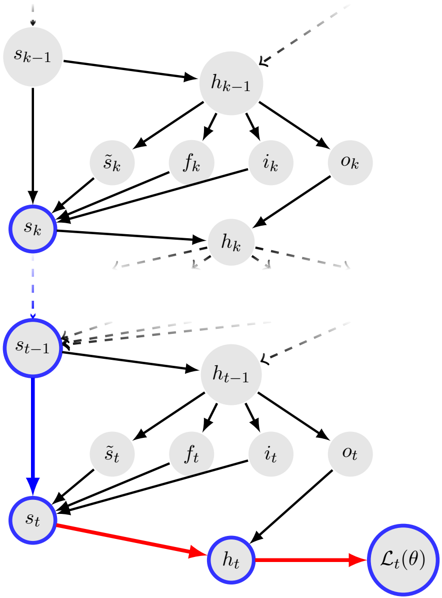
- Let us first look at $\frac{\partial h_t}{\partial s_t}$
- Recall that

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

- 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 $\mathcal{D}(o_t \odot \sigma'(s_t))$

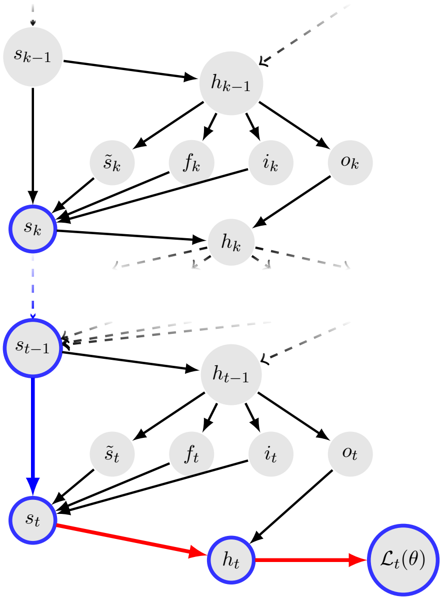


- Now let us consider $\frac{\partial s_t}{\partial s_{t-1}}$



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

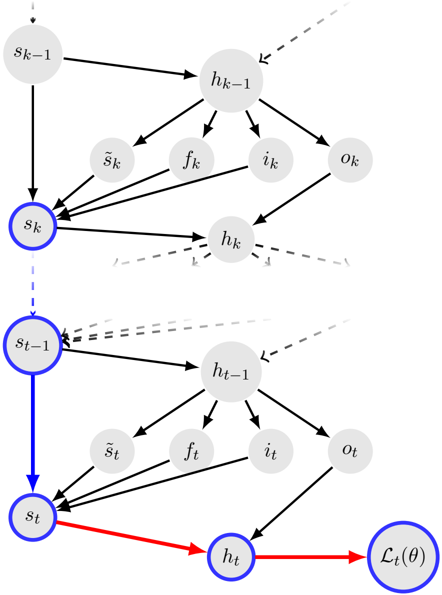
$$s_t = f_t \odot s_{t-1} + i_t \odot \tilde{s}_t$$



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

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

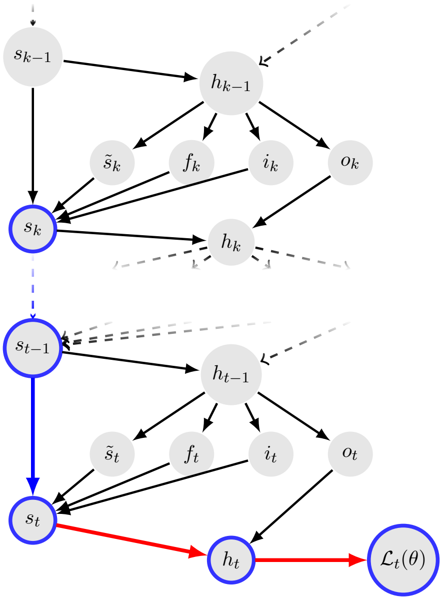
- Notice that $\tilde{s}_t$ also depends on s_{t-1} so we cannot treat it as a constant



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

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

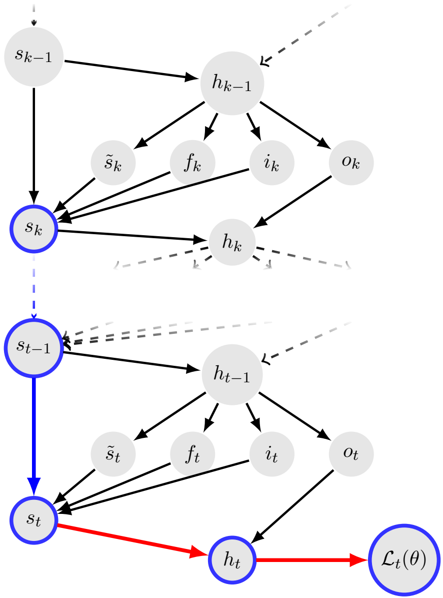
- Notice that $\tilde{s}_t$ also depends on s_{t-1} so we cannot treat it as a constant
- So once again we are dealing with an ordered network and thus $\frac{\partial s_t}{\partial s_{t-1}}$ will be a sum of an explicit term and an implicit term (see slide 37 from Lecture 14)



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

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

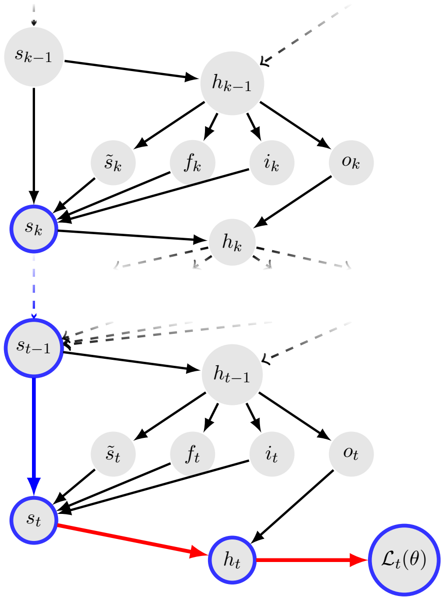
- Notice that $\tilde{s}_t$ also depends on s_{t-1} so we cannot treat it as a constant
- So once again we are dealing with an ordered network and thus $\frac{\partial s_t}{\partial s_{t-1}}$ will be a sum of an explicit term and an implicit term (see slide 37 from Lecture 14)
- For simplicity, let us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



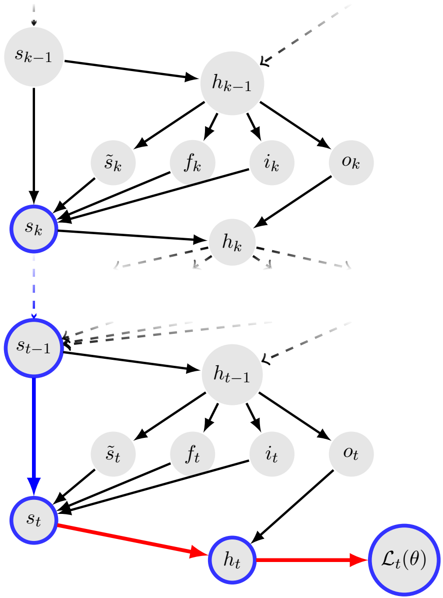
- Now let us consider $\frac{\partial s_t}{\partial s_{t-1}}$
- Recall that

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

- 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 $\mathcal{D}(f_t)$

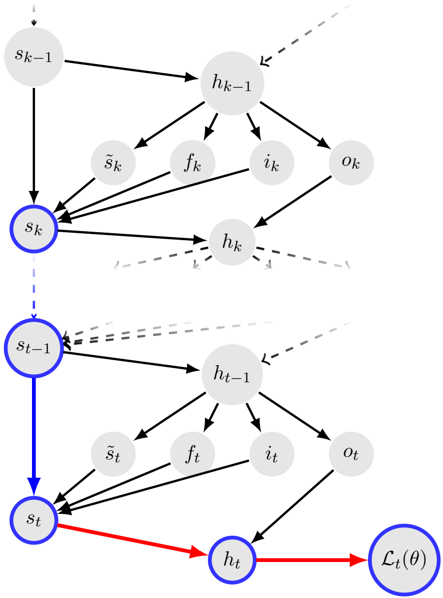


- We now return back to our full expression for t_0:



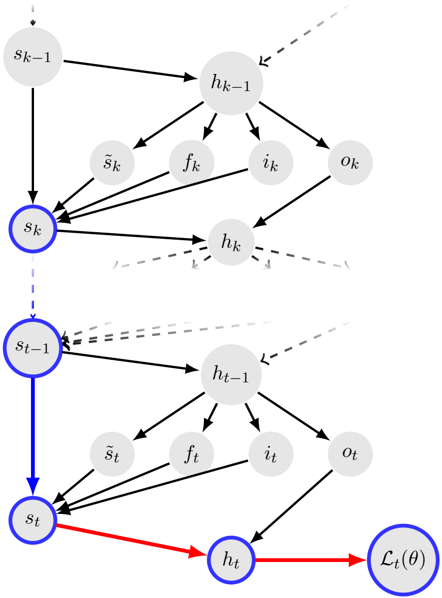
- We now return back to our full expression for t_0:

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



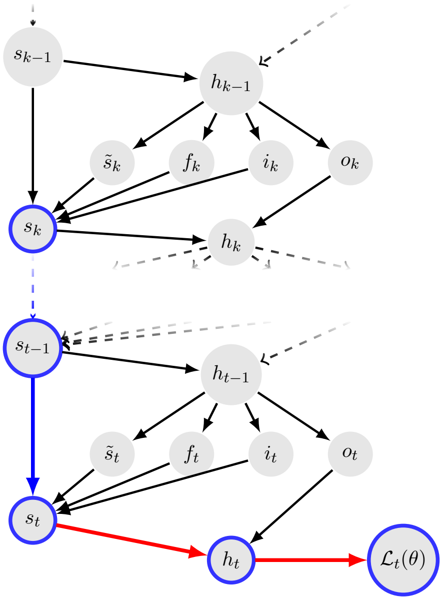
- We now return back to our full expression for t_0:

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



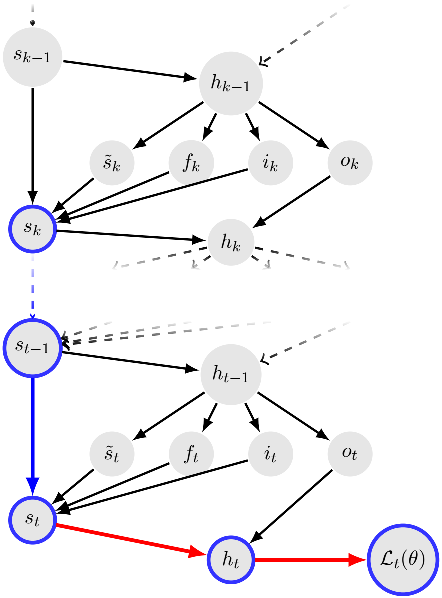
- We now return back to our full expression for t_0:

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



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

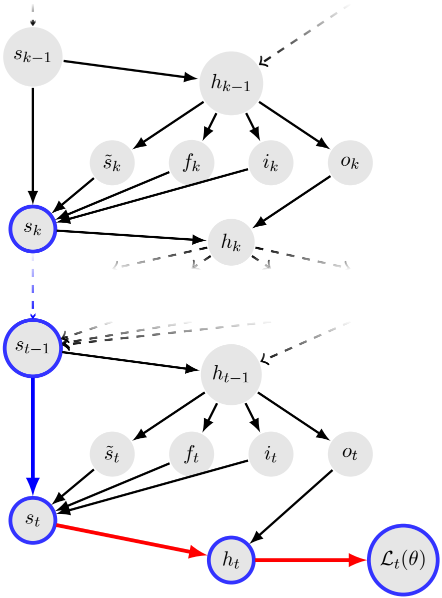
$$
\begin{aligned}
t_0 &= \frac{\partial \mathscr{L}_t(\theta)}{\partial h_t} \frac{\partial h_t}{\partial s_t} \frac{\partial s_t}{\partial s_{t-1}} \cdots \frac{\partial s_{k+1}}{\partial s_k} \\
&= \mathscr{L}'_t(h_t).\mathcal{D}(o_t \odot \sigma'(s_t))\mathcal{D}(f_t) \ldots \mathcal{D}(f_{k+1})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



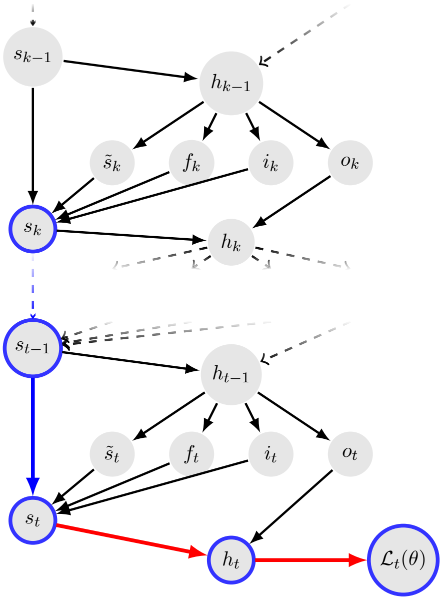
- We now return back to our full expression for t_0:

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

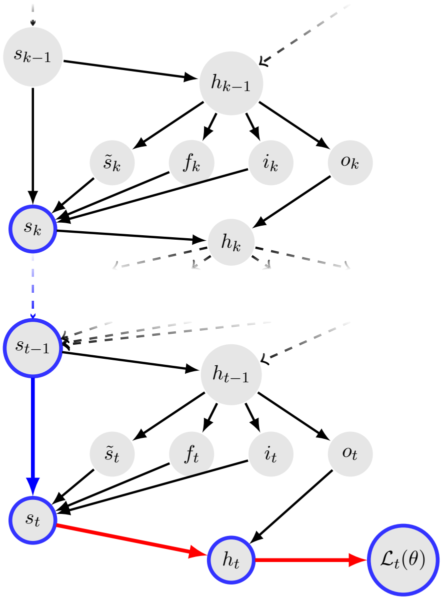
- The red terms don't vanish and the blue terms contain a multiplication of the forget gates
- The forget gates thus regulate the gradient flow depending on the explicit contribution of a state (s_t) to the next state s_{t+1}



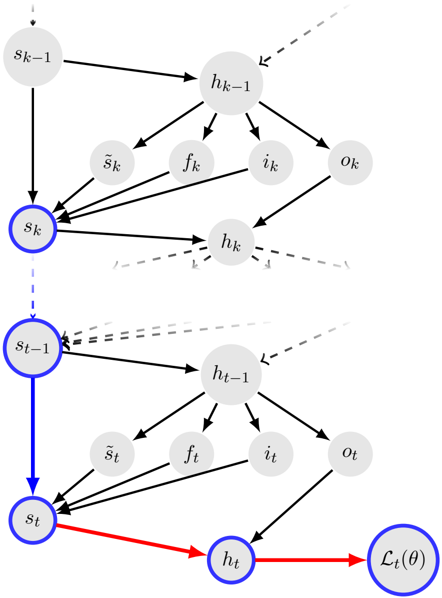
- If during forward pass s_t did not contribute much to s_{t+1} (because $f_t \to 0$) then during backpropgation also the gradient will not reach s_t



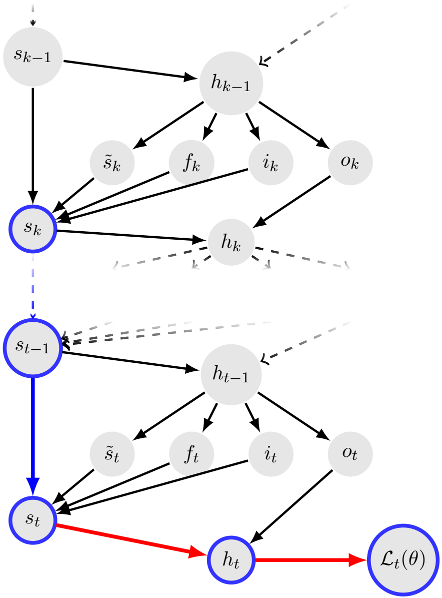
- If during forward pass s_t did not contribute much to s_{t+1} (because $f_t \to 0$) then during backpropgation also the gradient will not reach s_t
- This is fine because if s_t did not contribute much to s_{t+1} then there is no reason to hold it responsible during backpropgation



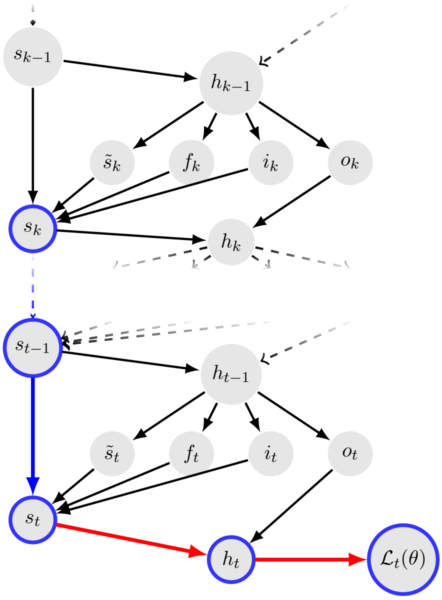
- If during forward pass s_t did not contribute much to s_{t+1} (because $f_t \rightarrow 0$) then during backpropgation also the gradient will not reach s_t
- This is fine because if s_t did not contribute much to s_{t+1} then there is no reason to hold it responsible during backpropgation (f_t does the same regulation during forward pass and backward pass which is fair)



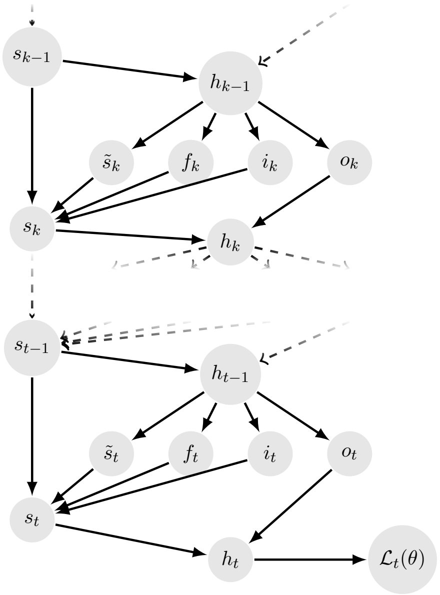
- If during forward pass s_t did not contribute much to s_{t+1} (because $f_t \to 0$) then during backpropgation also the gradient will not reach s_t
- This is fine because if s_t did not contribute much to s_{t+1} then there is no reason to hold it responsible during backpropgation (f_t does the same 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



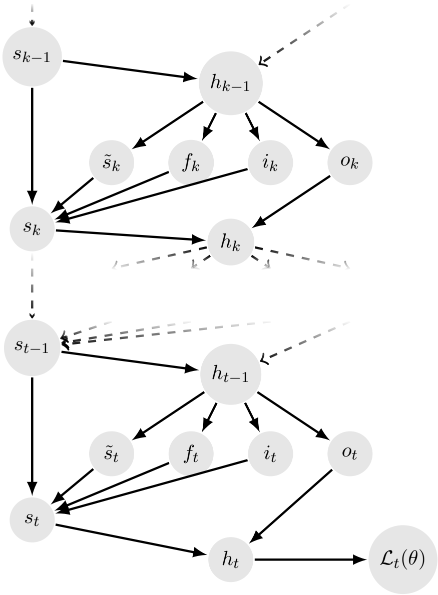
- If during forward pass s_t did not contribute much to s_{t+1} (because $f_t \to 0$) then during backpropgation also the gradient will not reach s_t
- This is fine because if s_t did not contribute much to s_{t+1} then there is no reason to hold it responsible during backpropgation (f_t does the same 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 !



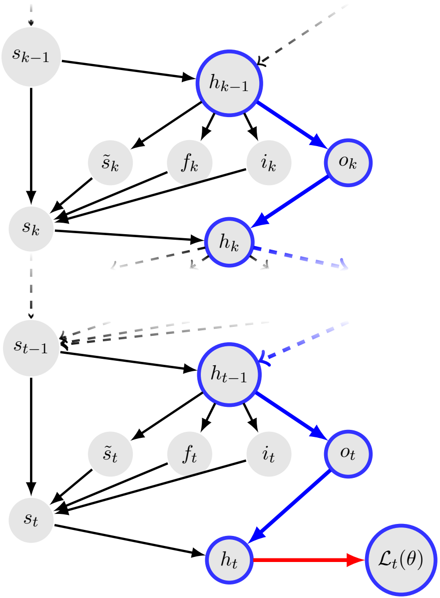
- If during forward pass s_t did not contribute much to s_{t+1} (because $f_t \to 0$) then during backpropgation also the gradient will not reach s_t
- This is fine because if s_t did not contribute much to s_{t+1} then there is no reason to hold it responsible during backpropgation (f_t does the same 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*)



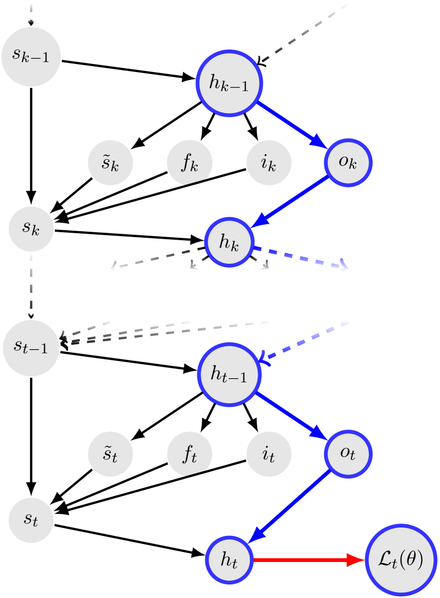
- Now we will see why LSTMs do not solve the problem of exploding gradients



- 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

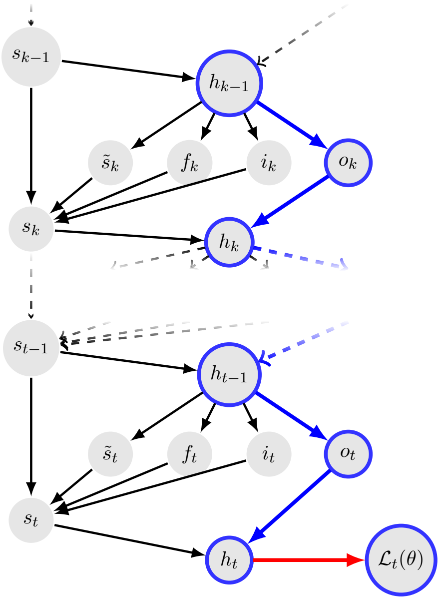


- Now we will see why LSTMs do not solve the problem of exploding gradients
- We will show a path through which the gradient can explode
- Let us compute one term (say t_1) of $\frac{\partial \mathcal{L}_t(\theta)}{\partial h_{k-1}}$ corresponding to the highlighted path



- 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

$$t_1 = \frac{\partial \mathcal{L}_t(\theta)}{\partial h_t} \left(\frac{\partial h_t}{\partial o_t} \frac{\partial o_t}{\partial h_{t-1}} \right) \dots \left(\frac{\partial h_k}{\partial o_k} \frac{\partial o_k}{\partial h_{k-1}} \right)$$

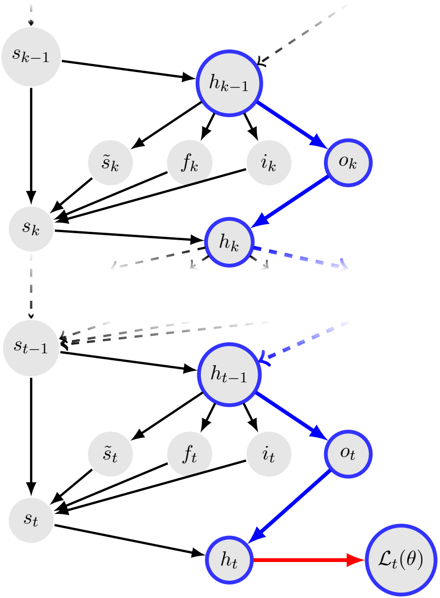


- Now we will see why LSTMs do not solve the problem of exploding gradients
- We will show a path through which the gradient can explode
- Let us compute one term (say t_1) of $\frac{\partial \mathcal{L}_t(\theta)}{\partial h_{k-1}}$ corresponding to the highlighted path

$$t_1 = \frac{\partial \mathcal{L}_t(\theta)}{\partial h_t} \left(\frac{\partial h_t}{\partial o_t} \frac{\partial o_t}{\partial h_{t-1}} \right) \dots \left(\frac{\partial h_k}{\partial o_k} \frac{\partial o_k}{\partial h_{k-1}} \right)$$

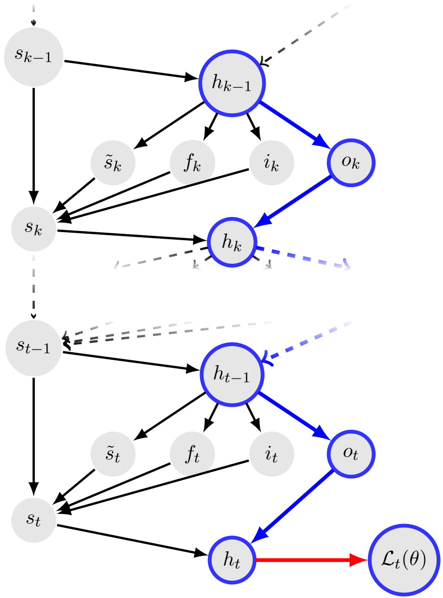
$$= \mathcal{L}'_t(h_t) \left(\mathcal{D}(\sigma(s_t) \odot o'_t).W_o \right) \dots$$

$$\left(\mathcal{D}(\sigma(s_k) \odot o'_k).W_o \right)$$



- 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

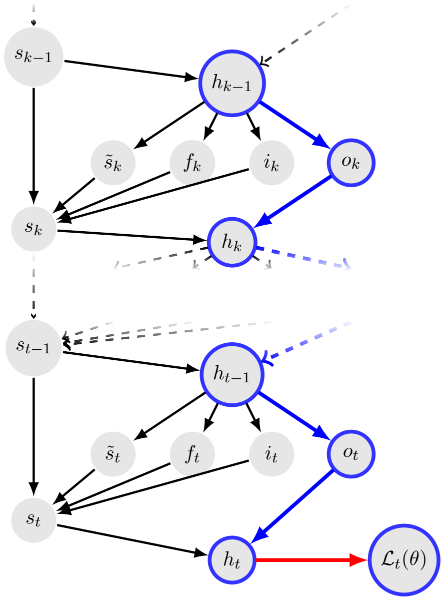
$$
\begin{aligned}
t_1 &= \frac{\partial \mathcal{L}_t(\theta)}{\partial h_t} \left(\frac{\partial h_t}{\partial o_t} \frac{\partial o_t}{\partial h_{t-1}} \right) \dots \left(\frac{\partial h_k}{\partial o_k} \frac{\partial o_k}{\partial h_{k-1}} \right) \\
&= \mathcal{L}'_t(h_t) \left(\mathcal{D}(\sigma(s_t)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



- 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 $\frac{\partial \mathscr{L}_t(\theta)}{\partial h_{k-1}}$ corresponding to the highlighted path

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

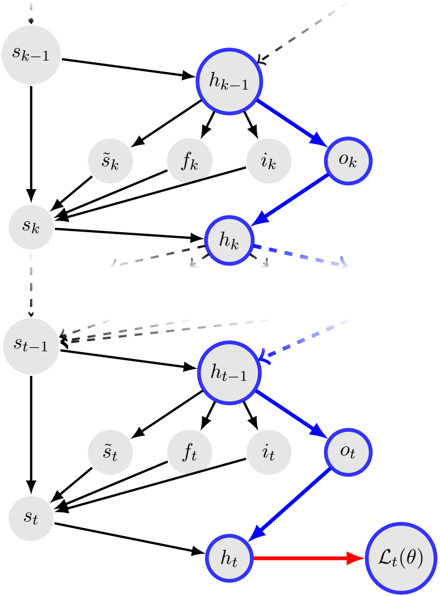
- Depending on the norm of matrix W_o, the gradient $\frac{\partial \mathscr{L}_t(\theta)}{\partial h_{k-1}}$ may explode



- 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 $\frac{\partial \mathscr{L}_t(\theta)}{\partial h_{k-1}}$ corresponding to the highlighted path

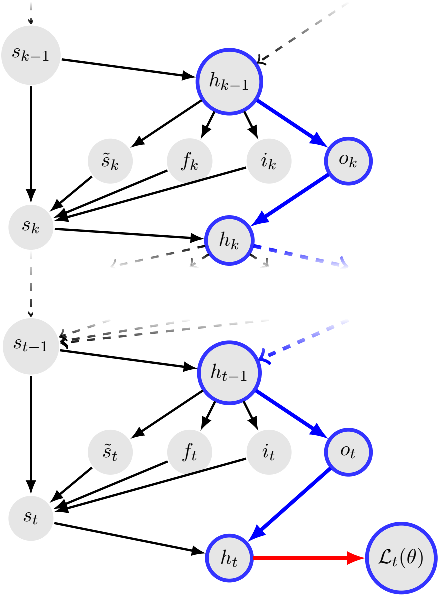
$$t_1 = \frac{\partial \mathscr{L}_t(\theta)}{\partial h_t} \left(\frac{\partial h_t}{\partial o_t} \frac{\partial o_t}{\partial h_{t-1}} \right) \dots \left(\frac{\partial h_k}{\partial o_k} \frac{\partial o_k}{\partial h_{k-1}} \right)$$
$$= \mathscr{L}'_t(h_t) \left(\mathcal{D}(\sigma(s_t) \odot o'_t).W_o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

- Depending on the norm of matrix W_o, the gradient $\frac{\partial \mathscr{L}_t(\theta)}{\partial h_{k-1}}$ may explode
- Similarly, W_i, W_f and W can also cause the gradients to explode



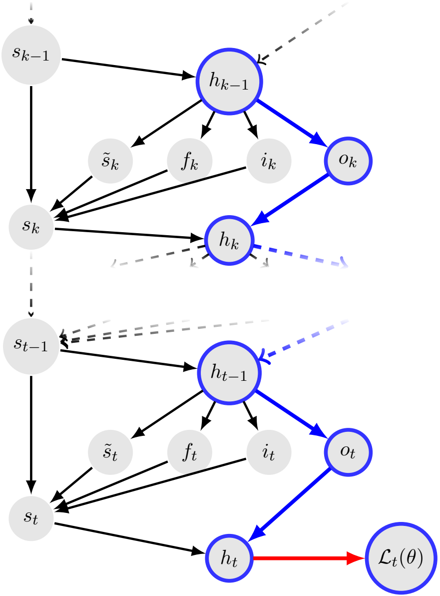
- So how do we deal with the problem of exploding gradients ?

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



- So how do we deal with the problem of exploding gradients ?
- One popular trick is to use gradient clipping

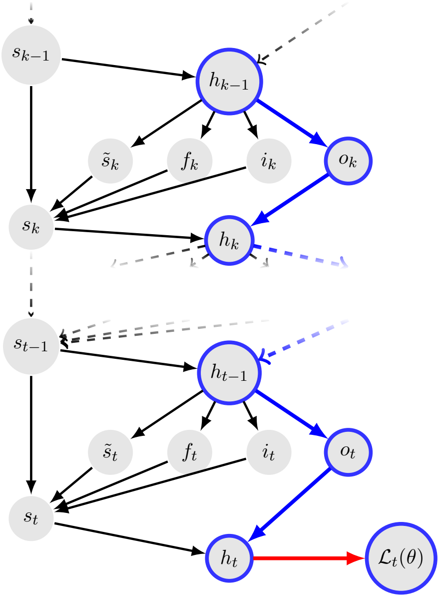
*Pascanu, Razvan, Tomas Mikolov, and Yoshua Bengio. "On the difficulty of training recurrent neural networks." ICML(3)28(2013):1310-1318



- 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



- 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*
- Essentially we retain the direction of the gradient but scale down the norm

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