UE21CS343BB2
Topics in Deep Learning

Dr. Shylaja S S
Director of Cloud Computing & Big Data (CCBD), Centre for Data Sciences & Applied Machine Learning (CDSAML)
Department of Computer Science and Engineering
shylaja.sharath@pes.edu

Ack:Anirudh Chandrasekar, Teaching Assistant

Topics in Deep Learning

RNN - Backpropagation Through Time

Backpropagation Through Time

- The dimensions of the RNN components are as follows:

$x_i \in R^n$ (n-dimensional input)

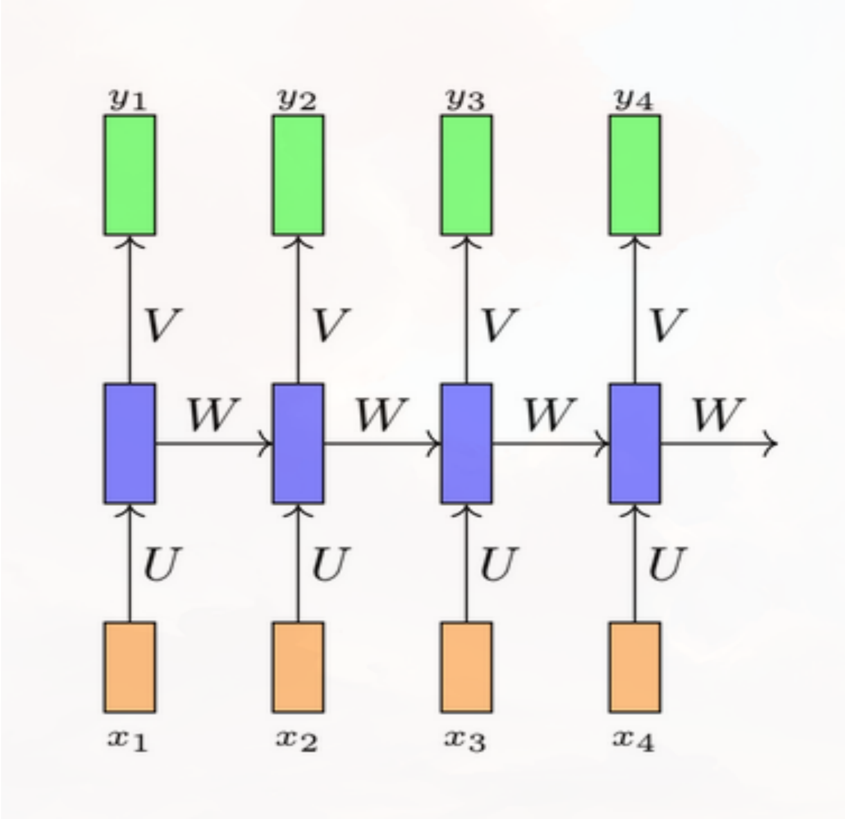
$h_i \in R^d$ (d-dimensional state)

$y_i \in R^k$ (say k classes)

$U \in R^{nxd}$

$V \in R^{dxk}$

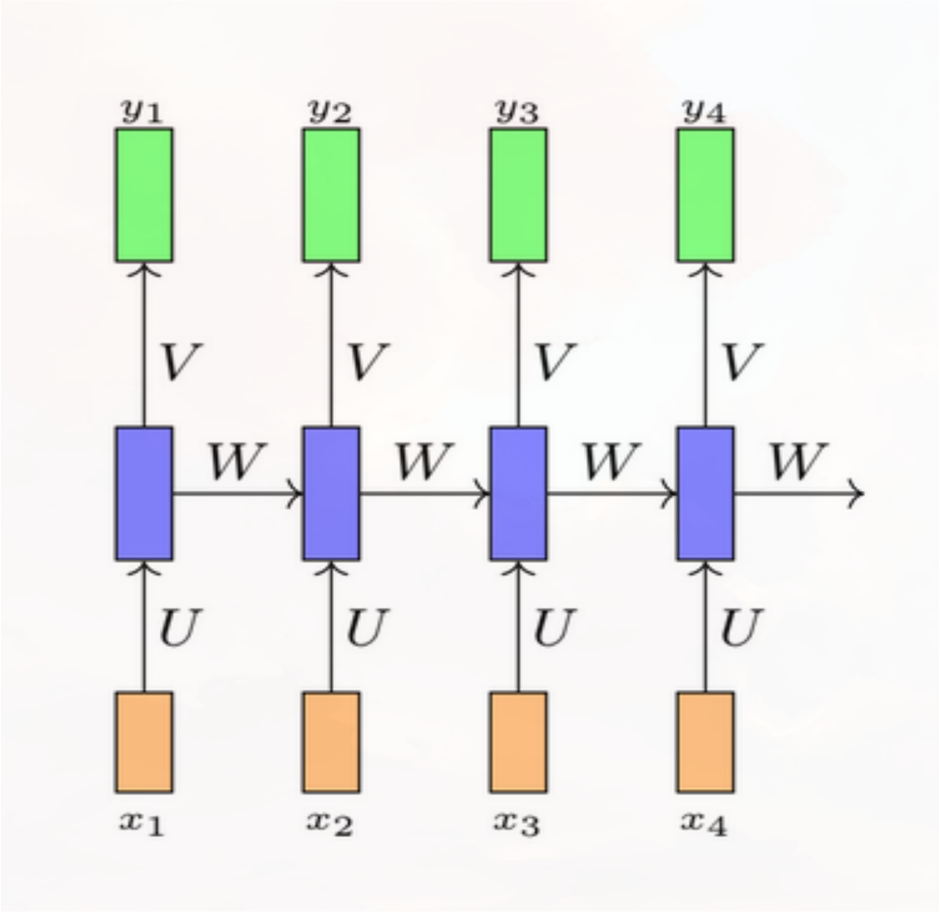
$W \in R^{dxd}$



Backpropagation Through Time

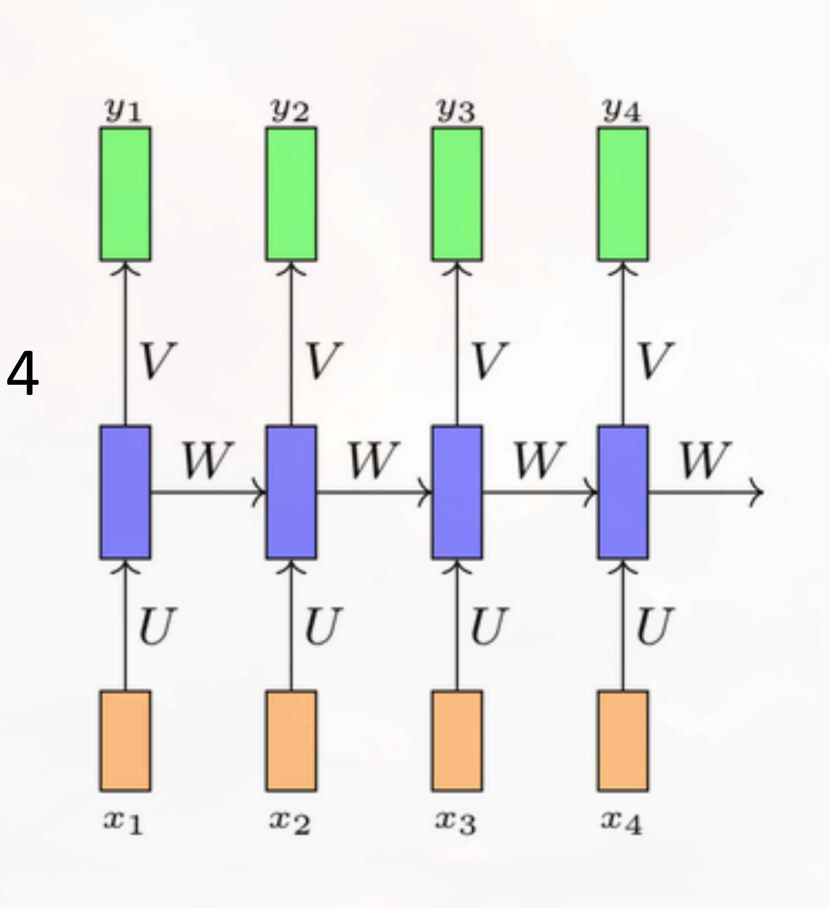
How do we train this network?

- Using Backpropagation



Backpropagation Through Time

- Suppose we consider our task of autocompletion (predicting the next character).
- For simplicity we assume that there are only 4 characters in our vocabulary (d,e,p, <stop>).
- At each timestep we want to predict one of these 4 characters.



What is a suitable output function for this task ?

Ans: Softmax

What is a suitable loss function for this task ?

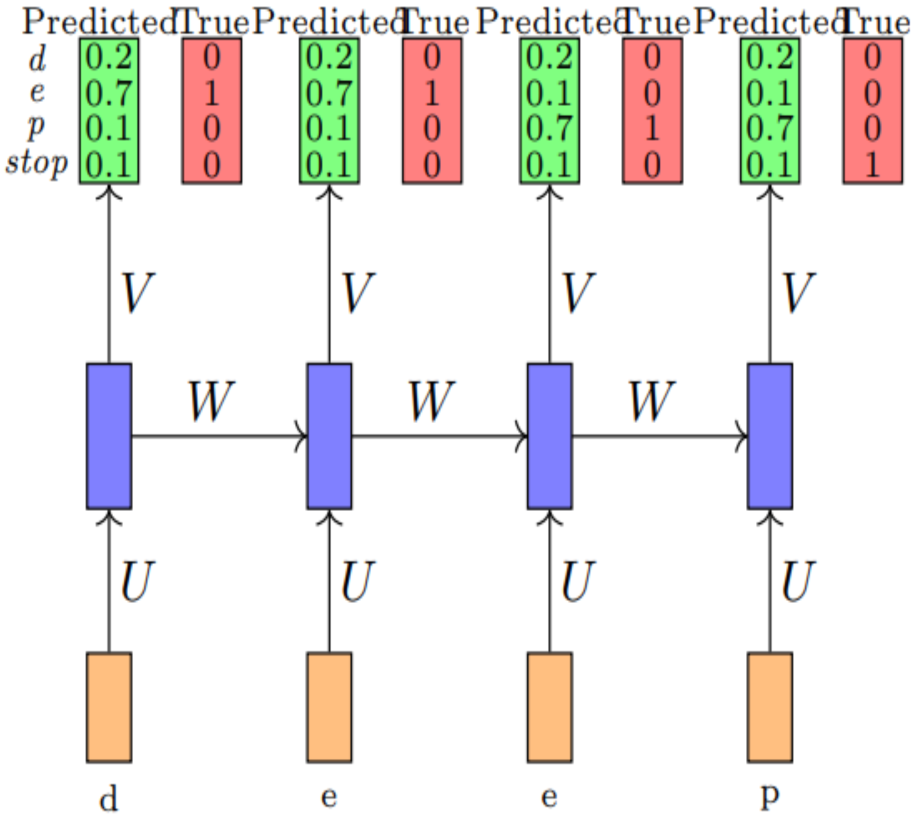
Ans: Cross Entropy

Backpropagation Through Time

- Suppose we initialize U, V, W randomly and the network predicts the probabilities as shown
- And the true probabilities are as shown in the snippet

We need to answer two questions

- What is the total loss made by the model?
- How do we backpropagate this loss and update the parameters (θ = {U, V, W, b, c}) of the network ?



Backpropagation Through Time

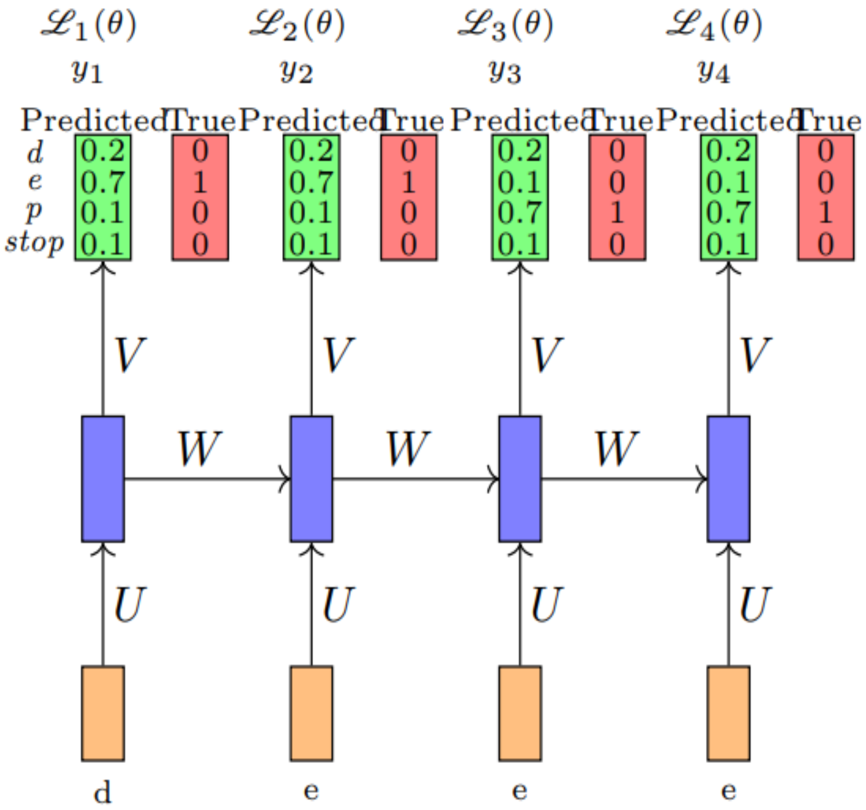
The total loss is simply the sum of the loss over all time-steps

$$\mathscr{L}(\theta) = \sum_{t=1}^{T} \mathscr{L}_t(\theta)$$

$$\mathscr{L}_t(\theta) = -log(y_{tc})$$

$$y_{tc} = \text{predicted probability of true character at time-step } t$$

$$T = \text{number of timesteps}$$



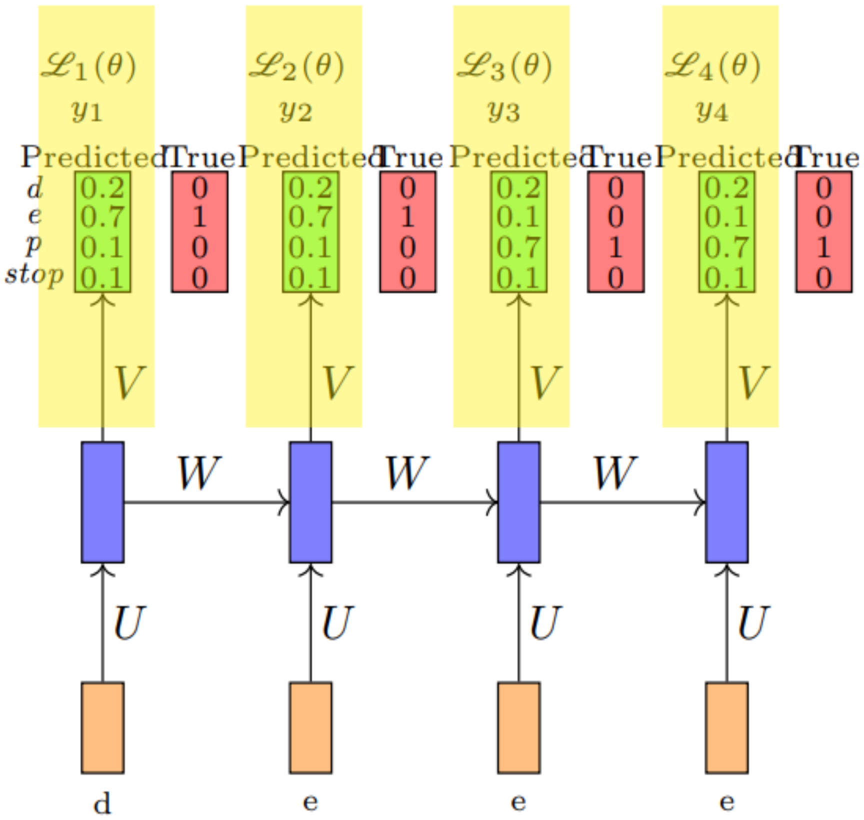
For backpropagation we need to compute the gradients w.r.t. W, U, V, b, c

Backpropagation Through Time

Let us consider the derivative wrt V:

$$\frac{\partial \mathscr{L}(\theta)}{\partial V} = \sum_{t=1}^{T} \frac{\partial \mathscr{L}_t(\theta)}{\partial V}$$

Each term is the summation is simply the derivative of the loss w.r.t. the weights in the output layer.

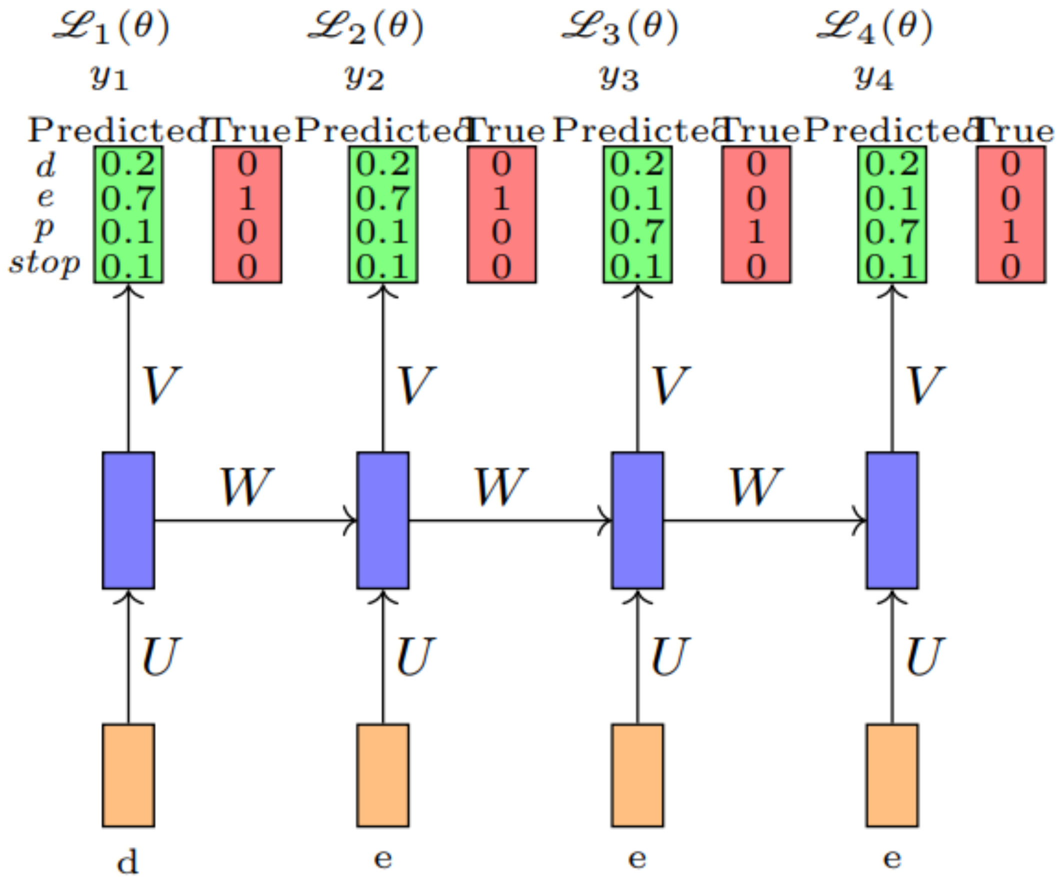


Backpropagation Through Time

- Let us consider the derivative

$$\frac{\partial \mathscr{L}(\theta)}{\partial W} = \sum_{t=1}^{T} \frac{\partial \mathscr{L}_t(\theta)}{\partial W}$$

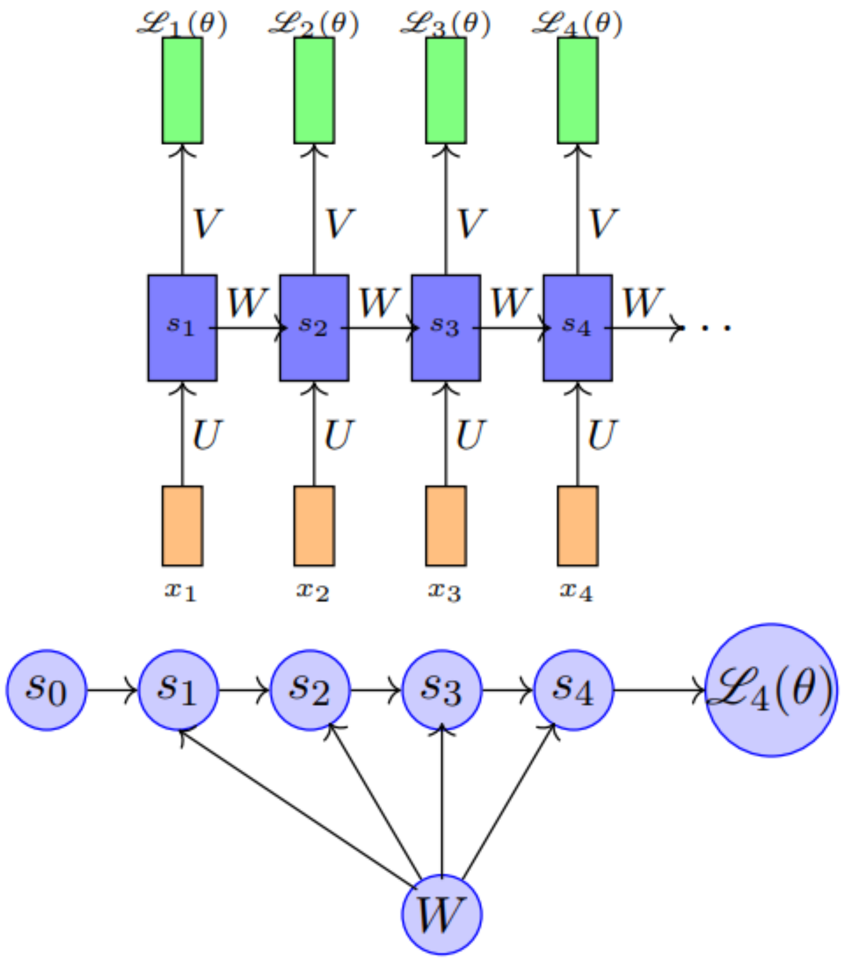
- By the chain rule of derivatives we know that $\partial L_t(\theta)/\partial W$ is obtained by summing gradients along all the paths from $L_t(\theta)$ to W
- What are the paths connecting $L_t(\theta)$ to W ?
- Let us see this by considering $L_4(\theta)$



Backpropagation Through Time

- L4(θ) depends on s4
- s4 in turn depends on s3 and W
- s3 in turn depends on s2 and W
- s2 in turn depends on s1 and W
- s1 in turn depends on s0 and W

where s0 is a constant starting state.

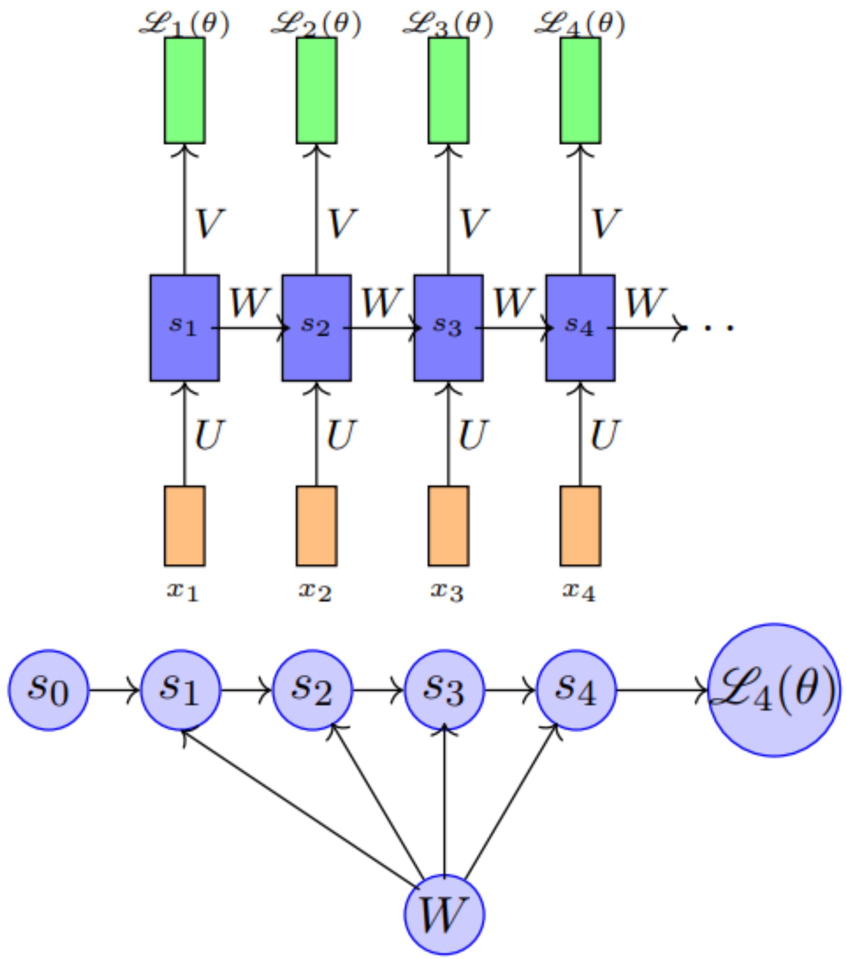


Backpropagation Through Time

- What we have here is an ordered network
- In an ordered network each state variable is computed one at a time in a specified order (first s1, then s2 and so on)
- Now we have

$$\frac{\partial \mathscr{L}_4(\theta)}{\partial W} = \frac{\partial \mathscr{L}_4(\theta)}{\partial s_4} \frac{\partial s_4}{\partial W}$$

- How do we compute $\partial s_4/\partial W$?

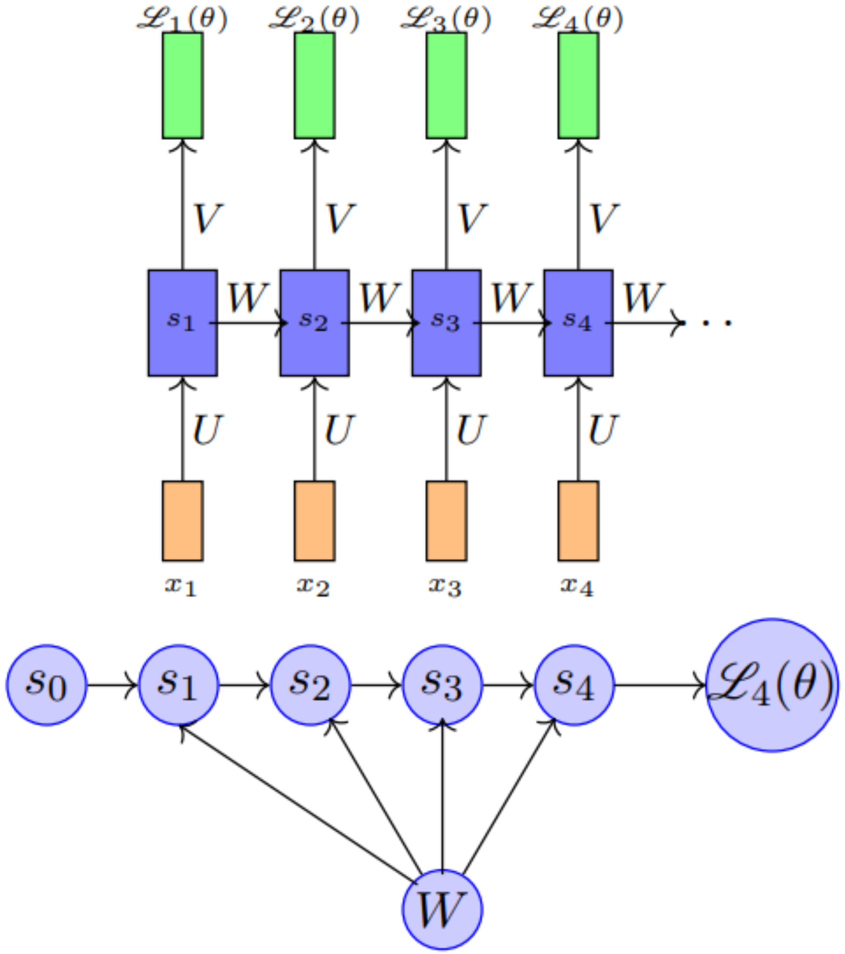


Backpropagation Through Time

- Recall that

$$s_4 = \sigma(W s_3 + b)$$

- In such an ordered network, we can't compute $\partial s_4/\partial W$ by simply treating s_3 as a constant (because it also depends on W)
- In such networks the total derivative $\partial s_4/\partial W$ has two parts
- Explicit : $\partial^+ s_4/\partial W$, treating all other inputs as constant
- Implicit : Summing over all indirect paths from s_4 to W



Backpropagation Through Time

$$\frac{\partial s_4}{\partial W} = \underbrace{\frac{\partial^+ s_4}{\partial W}}_{\text{explicit}} + \underbrace{\frac{\partial s_4}{\partial s_3}\frac{\partial s_3}{\partial W}}_{\text{implicit}}$$

$$= \frac{\partial^+ s_4}{\partial W} + \frac{\partial s_4}{\partial s_3}\left[\underbrace{\frac{\partial^+ s_3}{\partial W}}_{\text{explicit}} + \underbrace{\frac{\partial s_3}{\partial s_2}\frac{\partial s_2}{\partial W}}_{\text{implicit}}\right]$$

$$= \frac{\partial^+ s_4}{\partial W} + \frac{\partial s_4}{\partial s_3}\frac{\partial^+ s_3}{\partial W} + \frac{\partial s_4}{\partial s_3}\frac{\partial s_3}{\partial s_2}\left[\frac{\partial^+ s_2}{\partial W} + \frac{\partial s_2}{\partial s_1}\frac{\partial s_1}{\partial W}\right]$$

$$= \frac{\partial^+ s_4}{\partial W} + \frac{\partial s_4}{\partial s_3}\frac{\partial^+ s_3}{\partial W} + \frac{\partial s_4}{\partial s_3}\frac{\partial s_3}{\partial s_2}\frac{\partial^+ s_2}{\partial W} + \frac{\partial s_4}{\partial s_3}\frac{\partial s_3}{\partial s_2}\frac{\partial s_2}{\partial s_1}\left[\frac{\partial^+ s_1}{\partial W}\right]$$

For simplicity we will short-circuit some of the paths

$$\frac{\partial s_4}{\partial W} = \frac{\partial s_4}{\partial s_4}\frac{\partial^+ s_4}{\partial W} + \frac{\partial s_4}{\partial s_3}\frac{\partial^+ s_3}{\partial W} + \frac{\partial s_4}{\partial s_2}\frac{\partial^+ s_2}{\partial W} + \frac{\partial s_4}{\partial s_1}\frac{\partial^+ s_1}{\partial W} = \sum_{k=1}^{4}\frac{\partial s_4}{\partial s_k}\frac{\partial^+ s_k}{\partial W}$$

Backpropagation Through Time

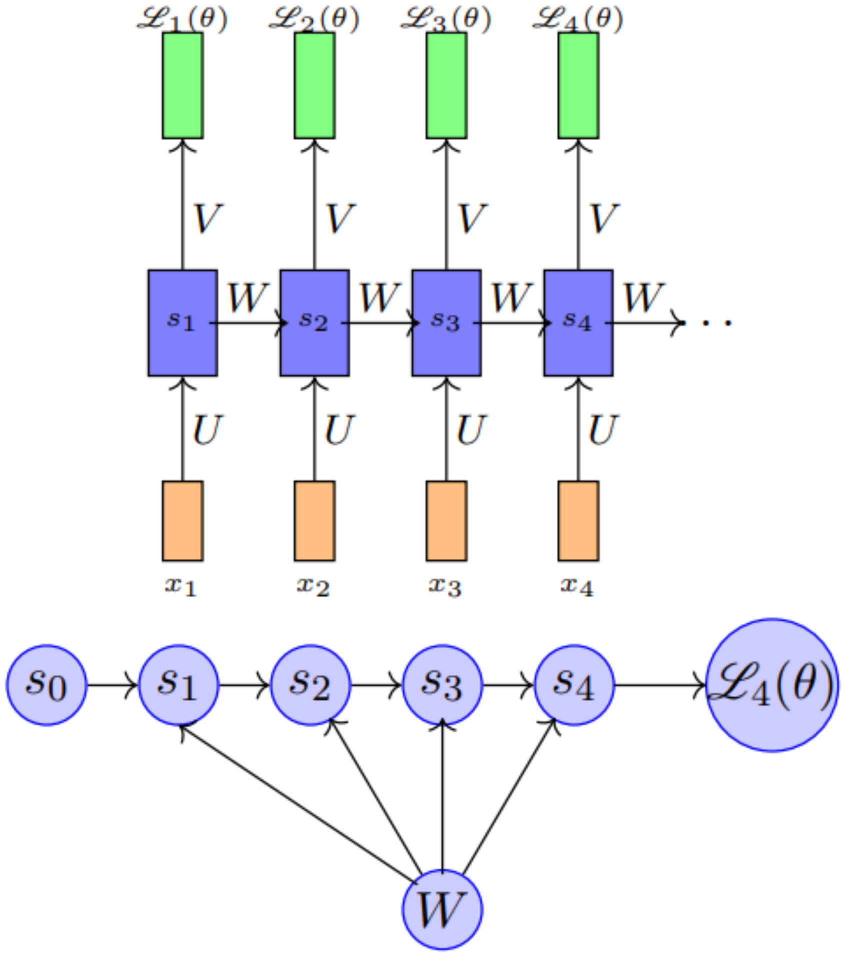
- Finally we have

$$\frac{\partial \mathscr{L}_4(\theta)}{\partial W} = \frac{\partial \mathscr{L}_4(\theta)}{\partial s_4} \frac{\partial s_4}{\partial W}$$

$$\frac{\partial s_4}{\partial W} = \sum_{k=1}^{4} \frac{\partial s_4}{\partial s_k} \frac{\partial^+ s_k}{\partial W}$$

$$\therefore \frac{\partial \mathscr{L}_t(\theta)}{\partial W} = \frac{\partial \mathscr{L}_t(\theta)}{\partial s_t} \sum_{k=1}^{t} \frac{\partial s_t}{\partial s_k} \frac{\partial^+ s_k}{\partial W}$$

- This algorithm is called backpropagation through time (BPTT) as we backpropagate over all previous time steps.



Acknowledgements & References

- https://deeplearning.ai
- https://youtu.be/OvCz1acvt-k?si=KW8X6Oave7QmcdH5
- https://youtu.be/Xeb6OjnVn8g?si=qdjNp0-LM2DvFkRR
- http://www.cse.iitm.ac.in/~miteshk/CS7015/Slides/Handout/Lecture14.pdf

UE21CS343BB2
Topics in Deep Learning

Dr. Shylaja S S
Director of Cloud Computing & Big Data (CCBD), Centre for Data Sciences & Applied Machine Learning (CDSAML)
Department of Computer Science and Engineering
shylaja.sharath@pes.edu

Ack:Anirudh Chandrasekar, Teaching Assistant