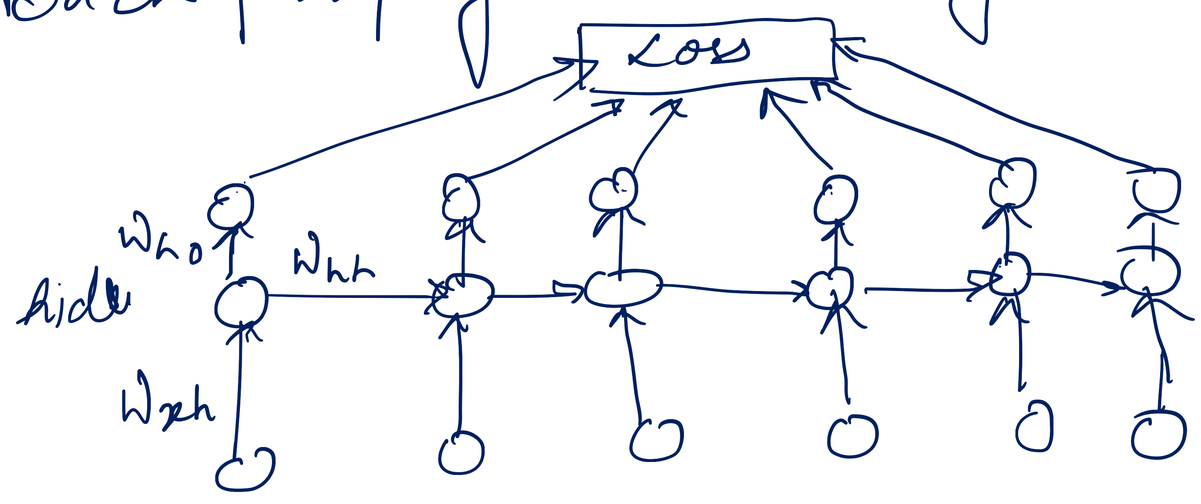
RNN's

Back propagation through time.



1) Vanishing gradient
2) Exploding gradient
   → One possible way of handling is Gradient Clipping

Difficult to handle

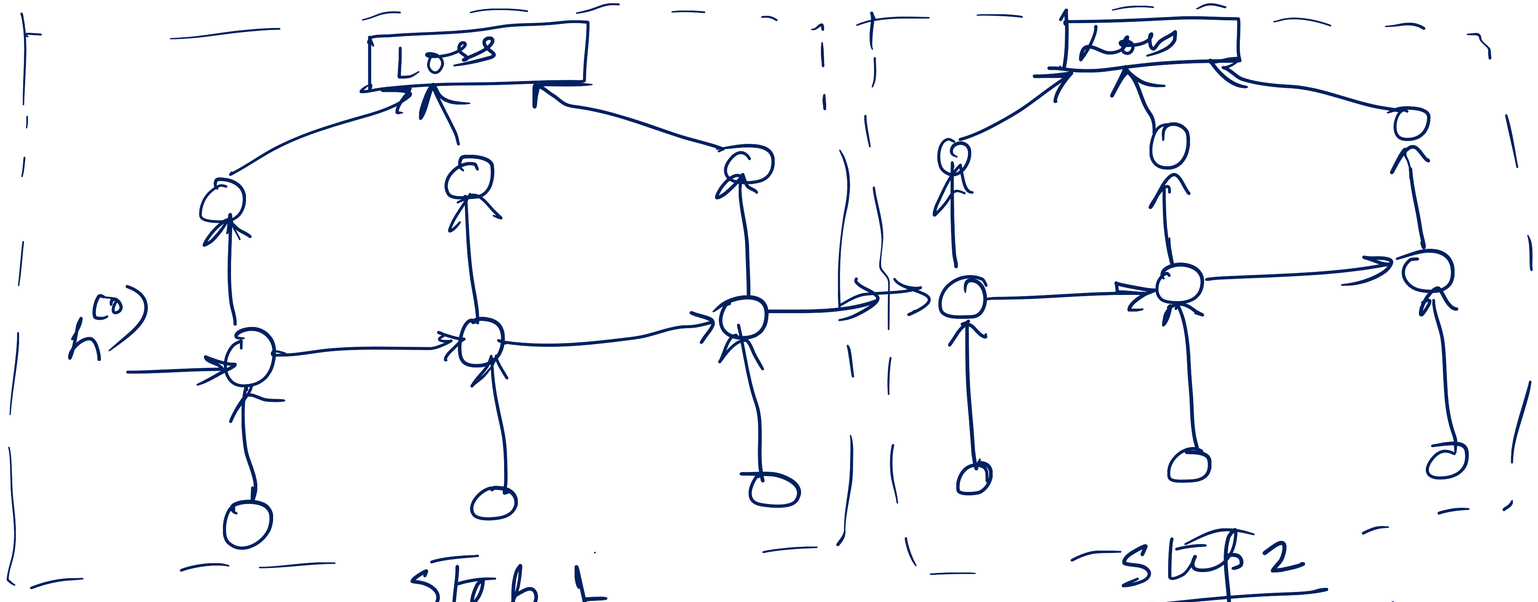
E.g. You are training a model on a set of wikipedia documents.

Each input unit recieves a word from a wikipedia document (in vector format)

For one update of the weight, run the forward propagation to compute the loss and then backward through the entire sequence to compute the gradient

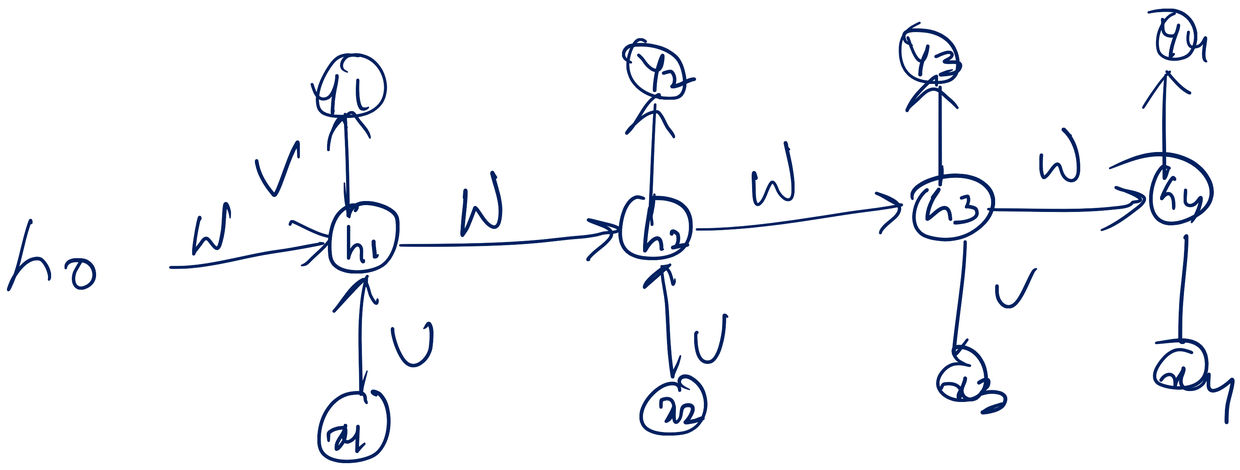
- ~~For a~~ Even for a single update the number of gradients to be calculated is equals the no. of words in the document.
  - Result is slow computation

- Solution is to use a truncated backpropagation.
  - Run forward and backward through chunks of the sequence instead of whole sequence.



- Generate subsequences and compute loss only over this subsequence of data.
- Use a batch of data as subsequence for calculating its loss
- Carry the hidden states forward in time forever, but only back propagate for some smaller no. of steps. ie back propagate through the next batch.

Derivation of the Gradient for BPTT



$$h_t = U x_t + W \phi(h_{t-1}) - (1)$$

$$y_t = V \phi(h_t) - (2)$$

Let the max stages be τ & let t be the current no. of stages.

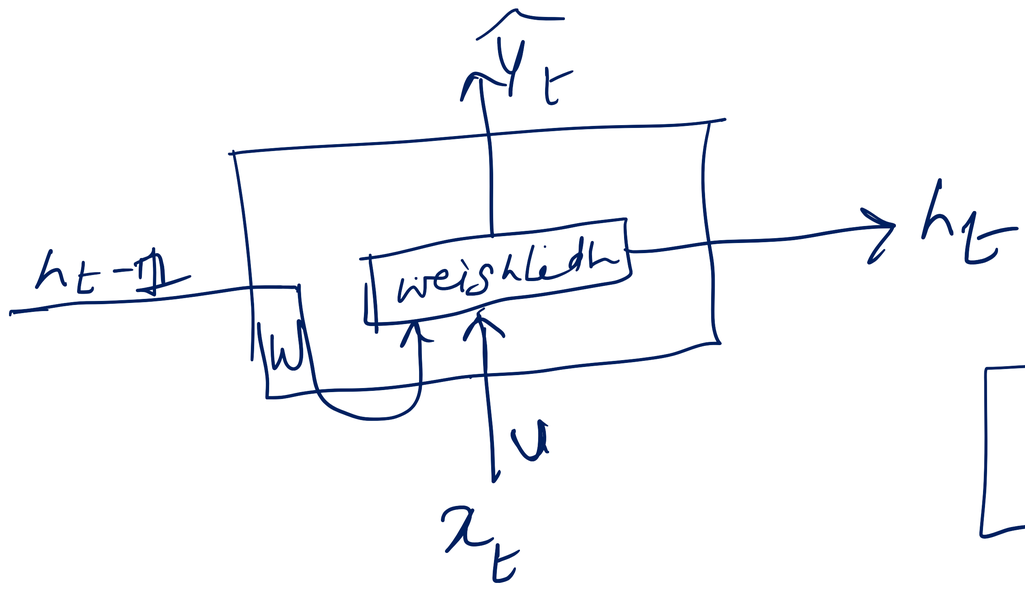
$$\frac{\partial E}{\partial W} = \sum_{t=1}^{\tau} \sum_{k=1}^{t} \frac{\partial E}{\partial y_t} \cdot \frac{\partial y_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial W}$$

$$\frac{\partial h_t}{\partial h_k} = \frac{\partial h_t}{\partial h_{t-1}} \cdot \frac{\partial h_{t-1}}{\partial h_{t-2}} \cdots \frac{\partial h_{k+1}}{\partial h_k}$$

$$= \prod \frac{\partial h_{ik}}{\partial h_k} = \prod_{i=k}^{t-1} \frac{\partial h_{i+1}}{\partial h_i} \quad - (3)$$

from eq (1) $\frac{\partial h_{i+1}}{\partial h_i} = W^T \phi'(h_{i-1}) = W^T \text{diag}(\phi' h_i)$

so $\frac{\partial h_t}{\partial h_k} = \prod_{i=k}^{t-1} W^T \text{diag}(\phi'(h_i))$



$$h_t = W h_{t-1} + U x_t$$

$$[W \; U]$$

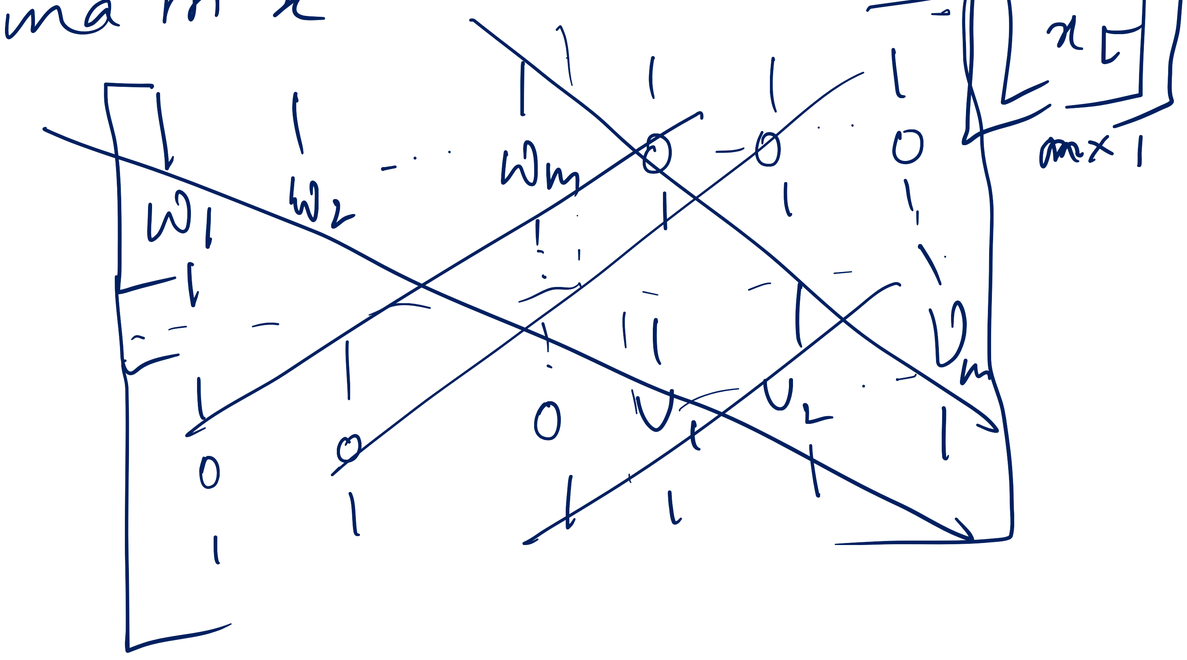
Create a new input as say $[h_{t-1}, x_t]$

ie by stacking h_{t-1} & x_t

$$\begin{bmatrix} h_{t-1} \\ m \times 1 \\ x_t \\ m \times 1 \end{bmatrix}$$

Create a new weight matrix

$$W_a = \begin{bmatrix} | & | & & | & | & | & | \\ w_1 & w_2 & \cdots & w_m & v_1 & v_2 & \cdots & v_m \\ | & | & & | & | & | & | \end{bmatrix}$$



$$W_a [h_{t-1}, x_t] = a_t$$

$$h_t = \phi (W_a [h_{t-1}, x_t] + b_a)$$ — bias

For the detailed derivation refer to slides 13&14.

Applications of RNN

Language model and Sequence Generation.

Generate text by training a model.

What is language modeling?

Given a vocabulary of words. $v_1, v_2 \ldots v_T$

Given a sequence of words $y_1, y_2 \ldots y_m$

Eg. "I am a good boy"

For a language model, the task is

Consider a language with a