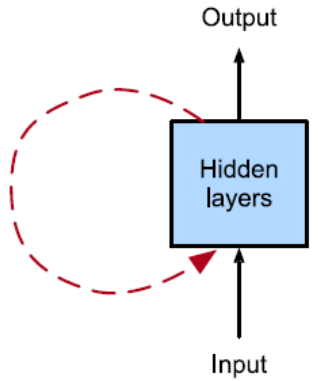
DSCI 565: RECURRENT NEURAL NETWORKS

This content is protected and may not be shared, uploaded, or distributed.

Ke-Thia Yao

Lecture 12: 2025 October 6

Recurrent Neural Networks



MLP without Hidden State

- MLP hidden layer with
  - Input: $\mathbf{X} \in \mathbb{R}^{n \times d}$, n batch size, d inputs
  - Output: $\mathbf{H} \in \mathbb{R}^{n \times h}$, h hidden units
  - Parameter: $\mathbf{W}_{\mathrm{xh}} \in \mathbb{R}^{d \times h}$
  - Parameter: $\mathbf{b}_{\mathrm{h}} \in \mathbb{R}^{1 \times h}$

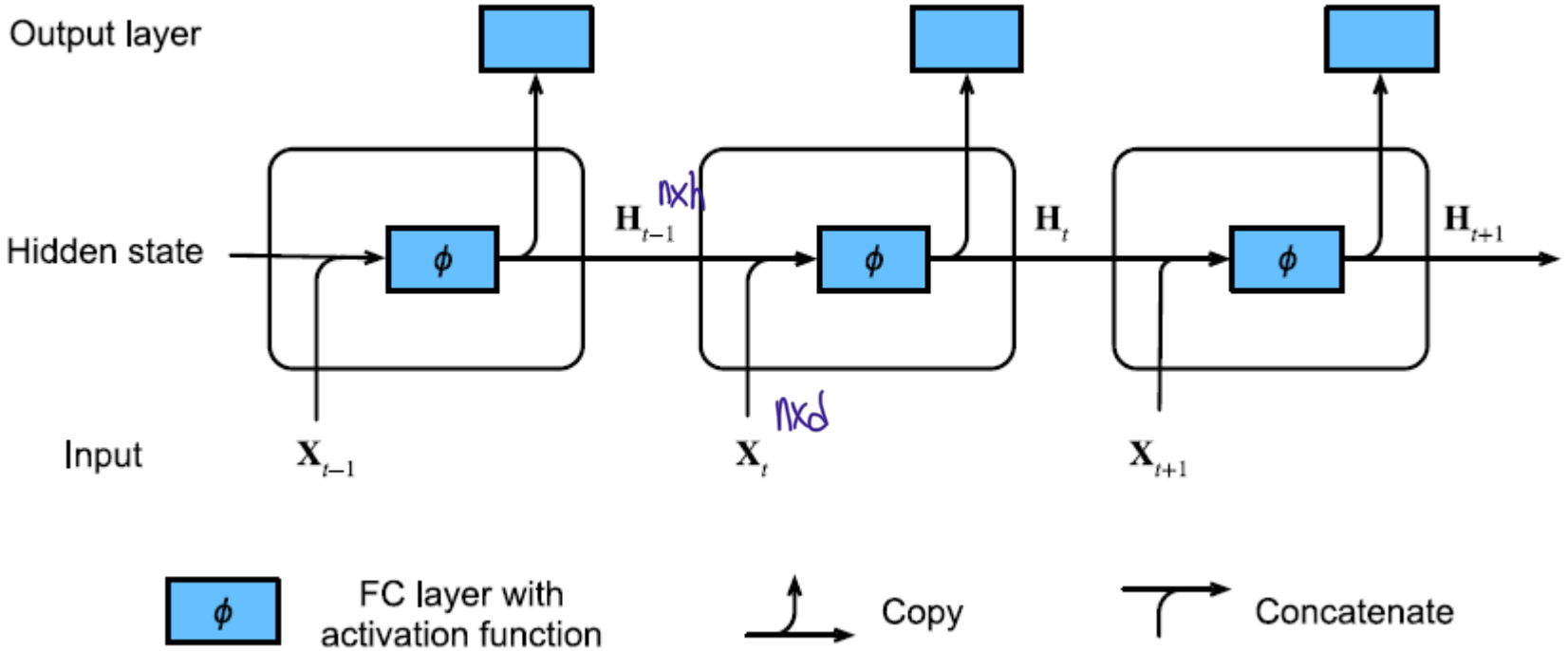
$$\underset{n\times h}{\mathbf{H}} = \phi(\underset{n\times d}{\mathbf{X}}\underset{d\times h}{\mathbf{W}_{\mathrm{xh}}} + \underset{n\times h}{\mathbf{b}_{\mathrm{h}}})$$

RNN with Hidden State

- RNN hidden layer with
  - Input: $\mathbf{X}_{\mathrm{t}} \in \mathbb{R}^{n \times d}$, n batch size, d inputs
  - Output: $\mathbf{H}_{\mathrm{t}} \in \mathbb{R}^{n \times h}$, h hidden units
  - Parameter: $\mathbf{W}_{\mathrm{xh}} \in \mathbb{R}^{d \times h}$
  - Parameter: $\mathbf{W}_{\mathrm{hh}} \in \mathbb{R}^{h \times h}$
  - Parameter: $\mathbf{b}_{\mathrm{h}} \in \mathbb{R}^{1 \times h}$

$$\underset{n\times h}{\mathbf{H}_t} = \phi(\underset{n\times d}{\mathbf{X}_t}\underset{d\times h}{\mathbf{W}_{\mathrm{xh}}} + \underset{n\times h}{\mathbf{H}_{t-1}}\underset{h\times h}{\mathbf{W}_{\mathrm{hh}}} + \underset{n\times h}{\mathbf{b}_{\mathrm{h}}})$$

RNN with a Hidden Layer

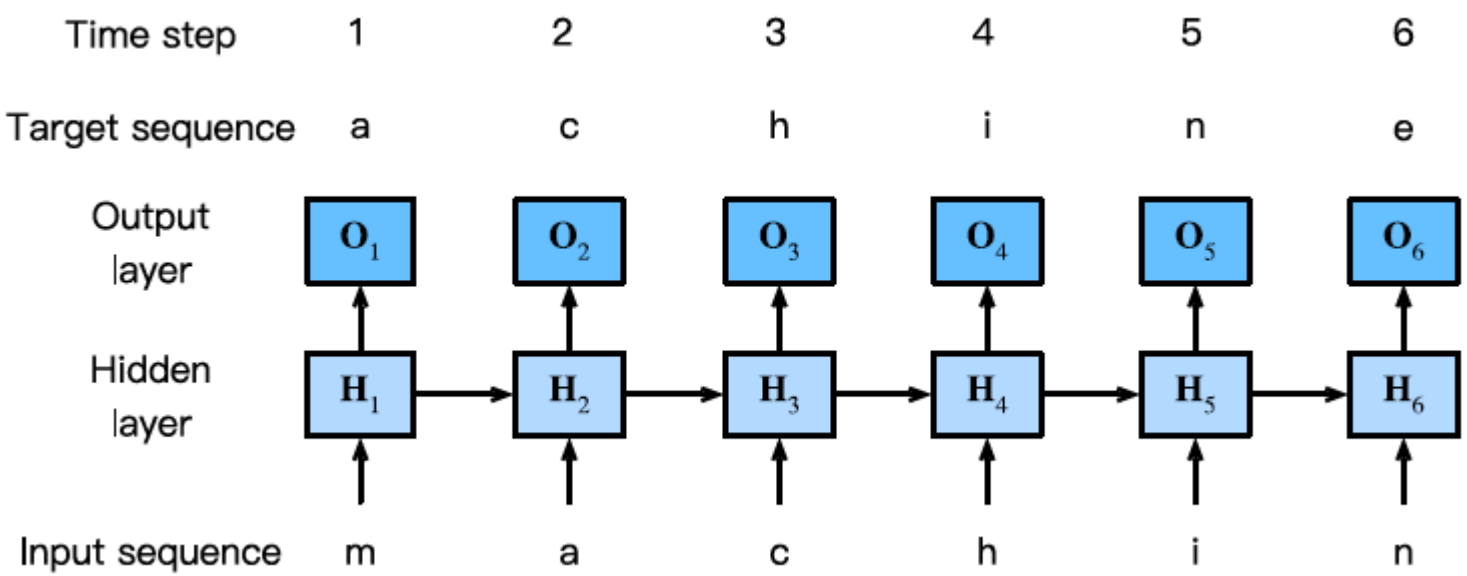


Can simplify by concatenation:

$$\text{concat}(\mathbf{X}_t, \mathbf{H}_t) \in \mathbb{R}^{n\times(d+h)}; \text{concat}(\mathbf{W}_{xh}, \mathbf{W}_{hh}) \in \mathbb{R}^{(d+h)\times h}$$

Character-level RNN

Character Level



- Input sequence: `machin`
- Output sequence: `achine`
- Each character is encoded as a d-dimensional vector , .e.g., one-hot encoding

d: dimension

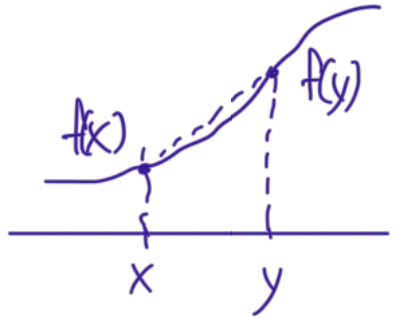
Gradient Clipping

- Training RNN uses backpropagation through time of "depth" $\mathcal{O}(T)$
- This can cause exploding or vanishing gradient problems
- When we update parameters with learning rate η: $\mathbf{x} \leftarrow \mathbf{x} - \eta \mathbf{g}$
- To limit the gradient within a ball of radius θ:

$$\mathbf{g} \leftarrow \min\left(1, \frac{\theta}{\|\mathbf{g}\|}\right)\mathbf{g}$$

- Formally, we say the objective/loss function f is *Lipschitz continuous* with constant L:

$$|f(\mathbf{x}) - f(\mathbf{y})| \leq L\|\mathbf{x} - \mathbf{y}\|$$
$$|f(\mathbf{x}) - f(\mathbf{x} - \eta\mathbf{g})| \leq L\eta\|\mathbf{g}\|$$



Notebook

- chapter_recurrent-neural-networks/rnn-scratch.ipynb
- chapter_recurrent-neural-networks/rnn-concise.ipynb

Backpropagation Through Time

- A RNN with one-hidden layer be represented at each time step t:

$$h_t = f(x_t, h_{t-1}, w_{\mathrm{h}}),$$

$$o_t = g(h_t, w_{\mathrm{o}}),$$

- Where f, g are transformation of the hidden and output layer, respectively
- Notice $h_t = f(x_t, h_{t-1}, w_h)$ depends on x_t, h_{t-1}, w_h, and recursively $h_{t-1} = f(x_{t-1}, h_{t-2}, w_h)$ depends on x_{t-1}, h_{t-2}, w_h
- Computing $\frac{\partial h_t}{\partial w_h}$ requires chaining through multiple $f(\dots)$

Backpropagation Through Time

$$h_t = f(x_t, h_{t-1}, w_h),$$
$$o_t = g(h_t, w_o),$$

- The discrepancy between output o_t and target y_t is

$$L(x_1, \ldots, x_T, y_1, \ldots, y_T, w_h, w_o) = \frac{1}{T}\sum_{t=1}^{T} l(y_t, o_t).$$

- Using the chain rule

$$\frac{\partial L}{\partial w_h} = \frac{1}{T}\sum_{t=1}^{T} \frac{\partial l(y_t, o_t)}{\partial w_h}$$
$$= \frac{1}{T}\sum_{t=1}^{T} \frac{\partial l(y_t, o_t)}{\partial o_t} \frac{\partial g(h_t, w_o)}{\partial h_t} \frac{\partial h_t}{\partial w_h}$$

Backpropagation Through Time

$$h_t = f(x_t, h_{t-1}, w_h),$$
$$o_t = g(h_t, w_o),$$

- Chain rule for partial of h_t wrt w_h

$$\frac{\partial h_t}{\partial w_h} = \frac{\partial f(x_t, h_{t-1}, w_h)}{\partial w_h} + \frac{\partial f(x_t, h_{t-1}, w_h)}{\partial h_{t-1}} \frac{\partial h_{t-1}}{\partial w_h}$$

- Notice the above recurrence relation has the form

$$a_t = b_t + c_t a_{t-1}$$

- Then $a_t = b_t + c_t(b_{t-1} + c_{t-1}a_{t-2}) = b_t + c_t b_{t-1} + c_t c_{t-1} a_{t-2}$
- And

$$a_t = b_t + \sum_{i=1}^{t-1} \left(\prod_{j=i+1}^{t} c_j \right) b_i$$

$a_t = b_t + c_t b_{t-1} + c_t c_{t-1} b_{t-2}$
$+ \cdots$

Backpropagation Through Time

- Then, substitute

$$a_t = \frac{\partial h_t}{\partial w_\mathrm{h}},$$

$$b_t = \frac{\partial f(x_t, h_{t-1}, w_\mathrm{h})}{\partial w_\mathrm{h}},$$

$$c_t = \frac{\partial f(x_t, h_{t-1}, w_\mathrm{h})}{\partial h_{t-1}},$$

into $a_t = b_t + c_t a_{t-1}.$

- We remove the recurrent computation:

$$\frac{\partial h_t}{\partial w_\mathrm{h}} = \frac{\partial f(x_t, h_{t-1}, w_\mathrm{h})}{\partial w_\mathrm{h}} + \sum_{i=1}^{t-1} \left(\prod_{j=i+1}^{t} \frac{\partial f(x_j, h_{j-1}, w_\mathrm{h})}{\partial h_{j-1}} \right) \frac{\partial f(x_i, h_{i-1}, w_\mathrm{h})}{\partial w_\mathrm{h}}.$$

Strategies for computing $\frac{\partial h_t}{\partial w_h}$

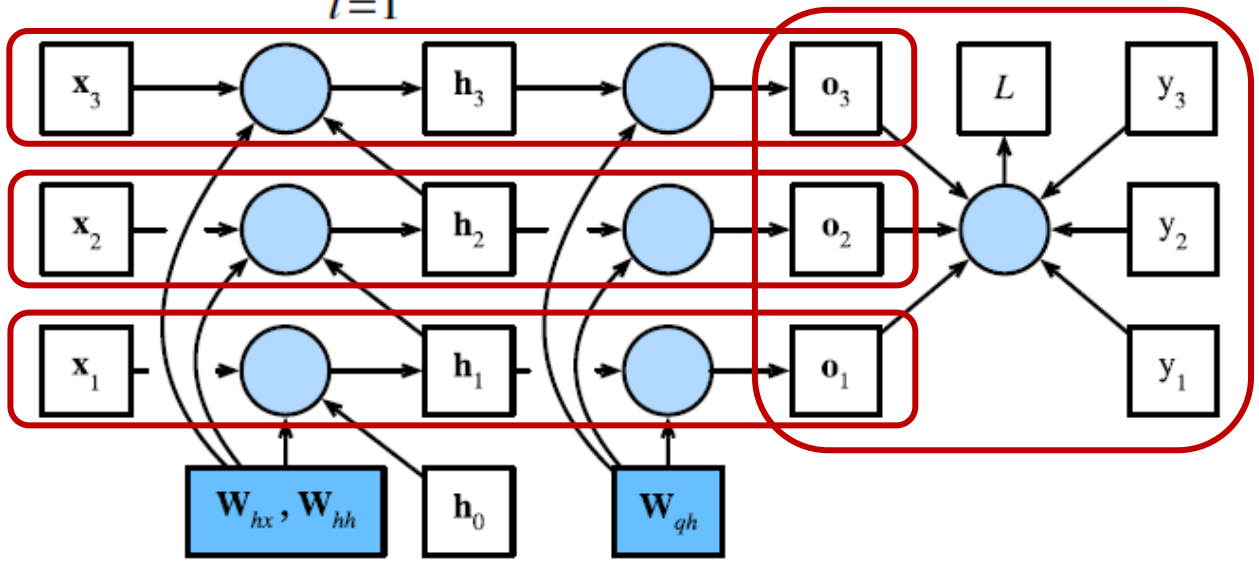
- Full Computation
  - Compute for all time steps $i = 1, \ldots, t-1$
  - Very slow, gradients can explode and sensitive to initial conditions
- Truncating time steps
  - Truncate after τ steps by terminating sum at $\partial h_{t-\tau}/\partial w_h$
  - Works well in practice. Biased toward more short-term influences.
- Random truncation
  - Randomly truncate the sequence
  - Empirically does not work much better than regular truncation

Forward Pass

$$\mathbf{h}_t = \mathbf{W}_{hx}\mathbf{x}_t + \mathbf{W}_{hh}\mathbf{h}_{t-1},$$

$$\mathbf{o}_t = \mathbf{W}_{qh}\mathbf{h}_t,$$

$$L = \frac{1}{T}\sum_{t=1}^{T} l(\mathbf{o}_t, y_t) \qquad \mathbf{o}_t \in \mathbb{R}^q$$

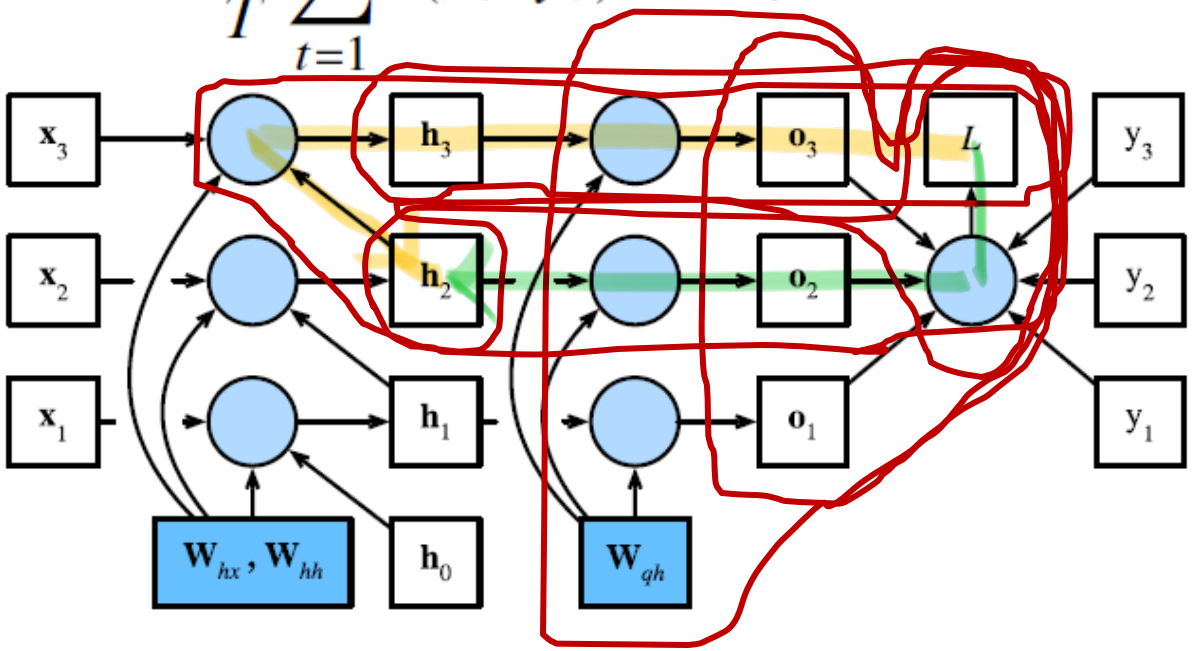


Backpropagation Through Time in Detail

$$\mathbf{h}_t = \mathbf{W}_{hx}\mathbf{x}_t + \mathbf{W}_{hh}\mathbf{h}_{t-1},$$

$$\mathbf{o}_t = \mathbf{W}_{qh}\mathbf{h}_t,$$

$$L = \frac{1}{T}\sum_{t=1}^{T} l(\mathbf{o}_t, y_t) \qquad \mathbf{o}_t \in \mathbb{R}^q$$



$$\frac{\partial L}{\partial \mathbf{o}_t} = \frac{\partial l(\mathbf{o}_t, y_t)}{T \cdot \partial \mathbf{o}_t} \in \mathbb{R}^q$$

$$\frac{\partial L}{\partial \mathbf{W}_{qh}} = \sum_{t=1}^{T} \text{prod}\left(\frac{\partial L}{\partial \mathbf{o}_t}, \frac{\partial \mathbf{o}_t}{\partial \mathbf{W}_{qh}}\right) = \sum_{t=1}^{T} \frac{\partial L}{\partial \mathbf{o}_t}\mathbf{h}_t^\top$$

$$\frac{\partial L}{\partial \mathbf{h}_T} = \text{prod}\left(\frac{\partial L}{\partial \mathbf{o}_T}, \frac{\partial \mathbf{o}_T}{\partial \mathbf{h}_T}\right) = \mathbf{W}_{qh}^\top \frac{\partial L}{\partial \mathbf{o}_T}$$

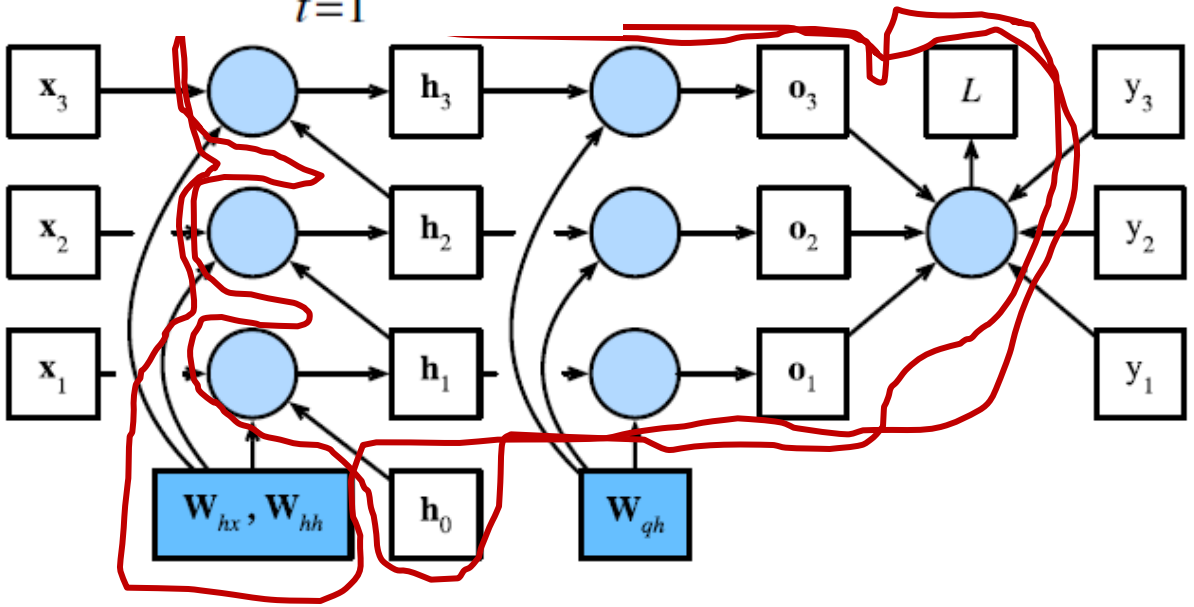
$$\frac{\partial L}{\partial \mathbf{h}_t} = \text{prod}\left(\frac{\partial L}{\partial \mathbf{h}_{t+1}}, \frac{\partial \mathbf{h}_{t+1}}{\partial \mathbf{h}_t}\right) + \text{prod}\left(\frac{\partial L}{\partial \mathbf{o}_t}, \frac{\partial \mathbf{o}_t}{\partial \mathbf{h}_t}\right)$$

$$= \mathbf{W}_{hh}^\top \frac{\partial L}{\partial \mathbf{h}_{t+1}} + \mathbf{W}_{qh}^\top \frac{\partial L}{\partial \mathbf{o}_t}$$

Backpropagation Through Time in Detail

$$\mathbf{h}_t = \mathbf{W}_{\text{hx}}\mathbf{x}_t + \mathbf{W}_{\text{hh}}\mathbf{h}_{t-1},$$
$$\mathbf{o}_t = \mathbf{W}_{\text{qh}}\mathbf{h}_t,$$

$$L = \frac{1}{T}\sum_{t=1}^{T} l(\mathbf{o}_t, y_t) \qquad \mathbf{o}_t \in \mathbb{R}^q$$



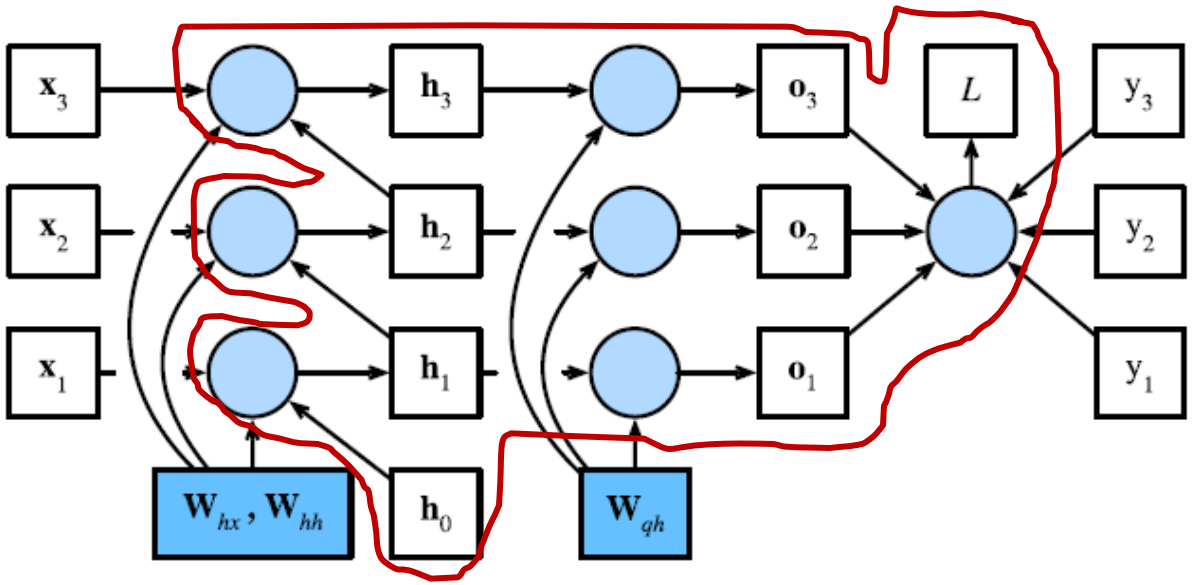
$$\frac{\partial L}{\partial \mathbf{W}_{\text{hx}}} = \sum_{t=1}^{T} \text{prod}\left(\frac{\partial L}{\partial \mathbf{h}_t}, \frac{\partial \mathbf{h}_t}{\partial \mathbf{W}_{\text{hx}}}\right) = \sum_{t=1}^{T} \frac{\partial L}{\partial \mathbf{h}_t}\mathbf{x}_t^\top,$$

$$\frac{\partial L}{\partial \mathbf{W}_{\text{hh}}} = \sum_{t=1}^{T} \text{prod}\left(\frac{\partial L}{\partial \mathbf{h}_t}, \frac{\partial \mathbf{h}_t}{\partial \mathbf{W}_{\text{hh}}}\right) = \sum_{t=1}^{T} \frac{\partial L}{\partial \mathbf{h}_t}\mathbf{h}_{t-1}^\top,$$

$$\frac{\partial L}{\partial \mathbf{h}_t} = \sum_{i=t}^{T} (\mathbf{W}_{\text{hh}}^\top)^{T-i}\mathbf{W}_{\text{qh}}^\top \frac{\partial L}{\partial \mathbf{o}_{T+t-i}}$$

Backpropagation Through Time in Detail

$$\frac{\partial L}{\partial \mathbf{h}_t} = \sum_{i=t}^{T} \left(\mathbf{W}_{\text{hh}}^{\top}\right)^{T-i} \mathbf{W}_{\text{qh}}^{\top} \frac{\partial L}{\partial \mathbf{o}_{T+t-i}}$$



- For long sequences, the matrix $\mathbf{W}_{hh}^{T}$ is raised to large powers
- Eigenvalues less than 1, it becomes very small
- Eigenvalues greater than 1, it becomes very large
- Matrix becomes ill conditioned, contributing to vanishing and exploding gradients