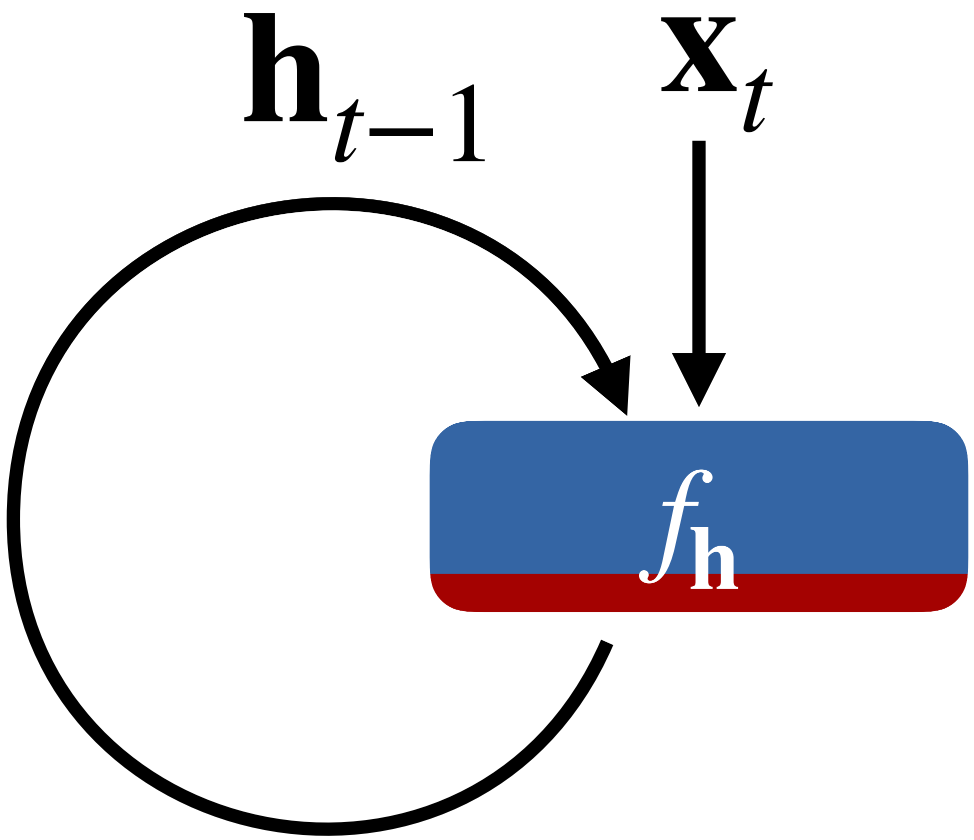
Training recurrent networks

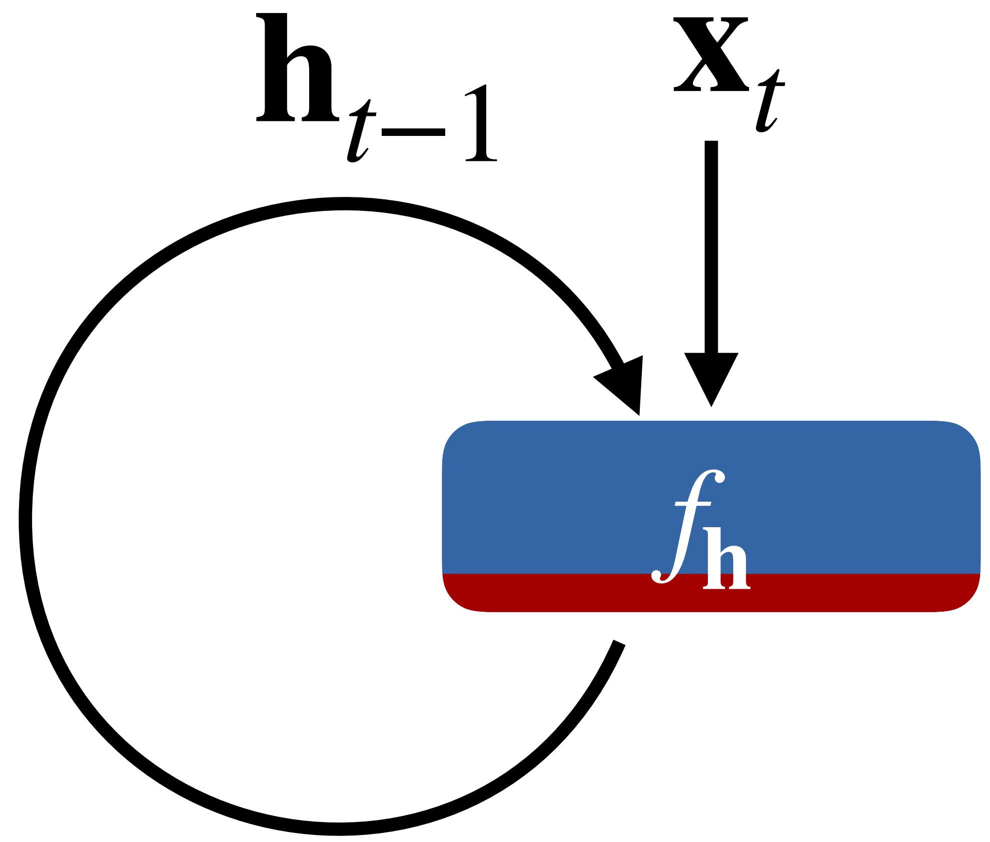
Recurrent Networks

- Processes a sequence
- Feedback connections

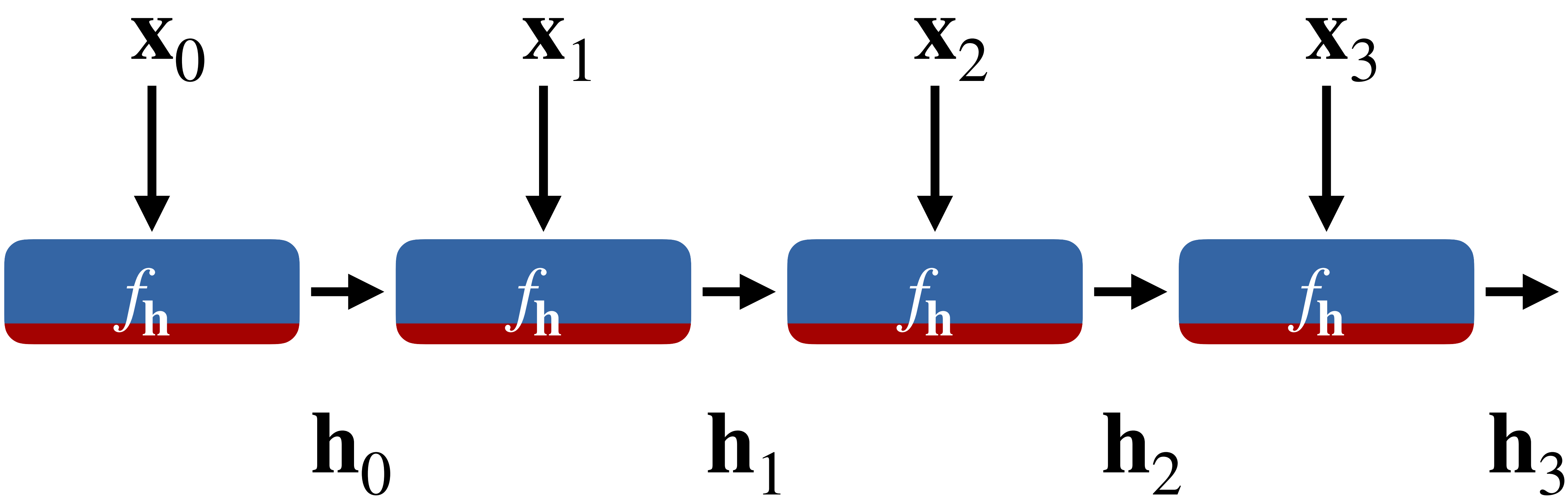


How do we train RNNs?

- No longer a simple forward and backward pass
    - Cycles

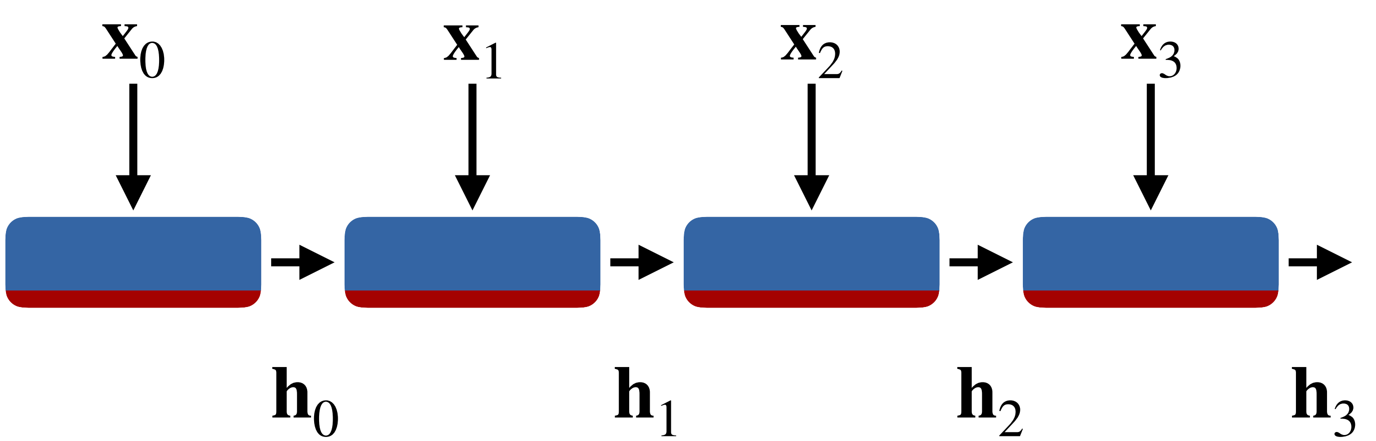


Solution: unrolling through time



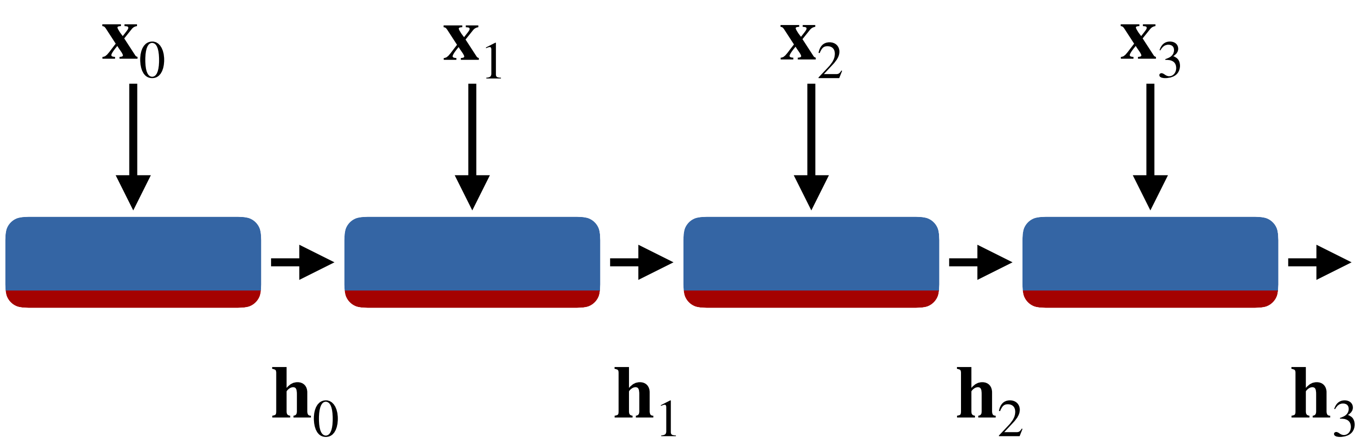
Unrolling through time

- Unrolled RNN (T steps)
  - Feed forward network
  - Shared parameters
- Trained with backprop



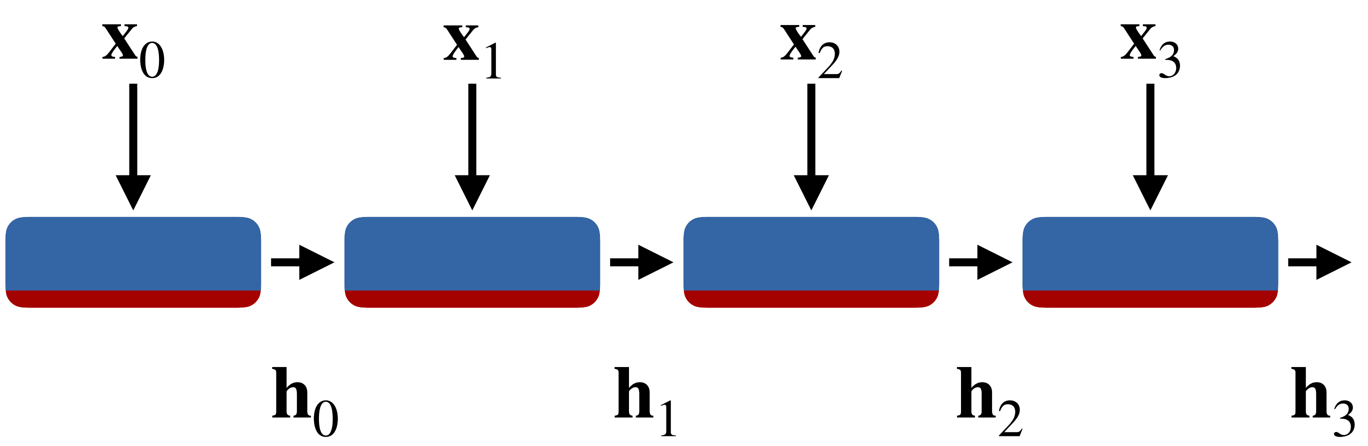
Unrolling through time - Issues

- Long unrolling
  - Computationally expensive
  - Vanishing or exploding gradients



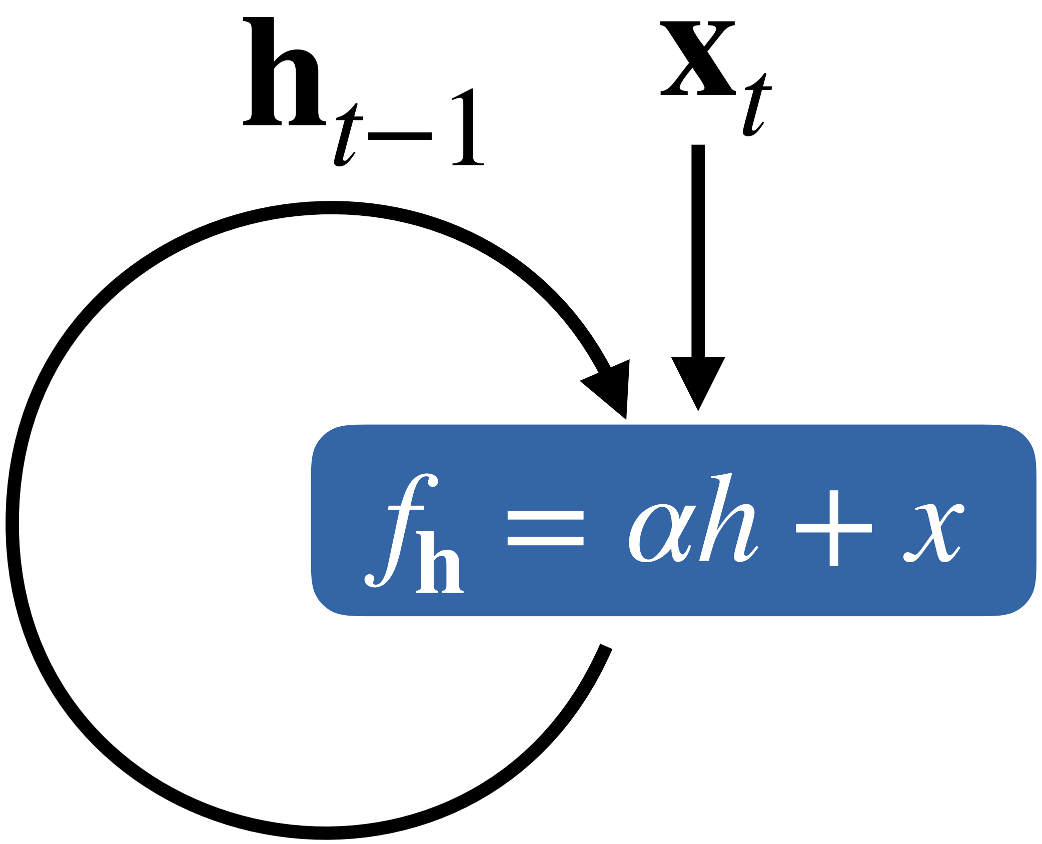
Computation

- Solution (hack)
  - Cut RNN after n steps
    - Set h=0
  - Works well in practice



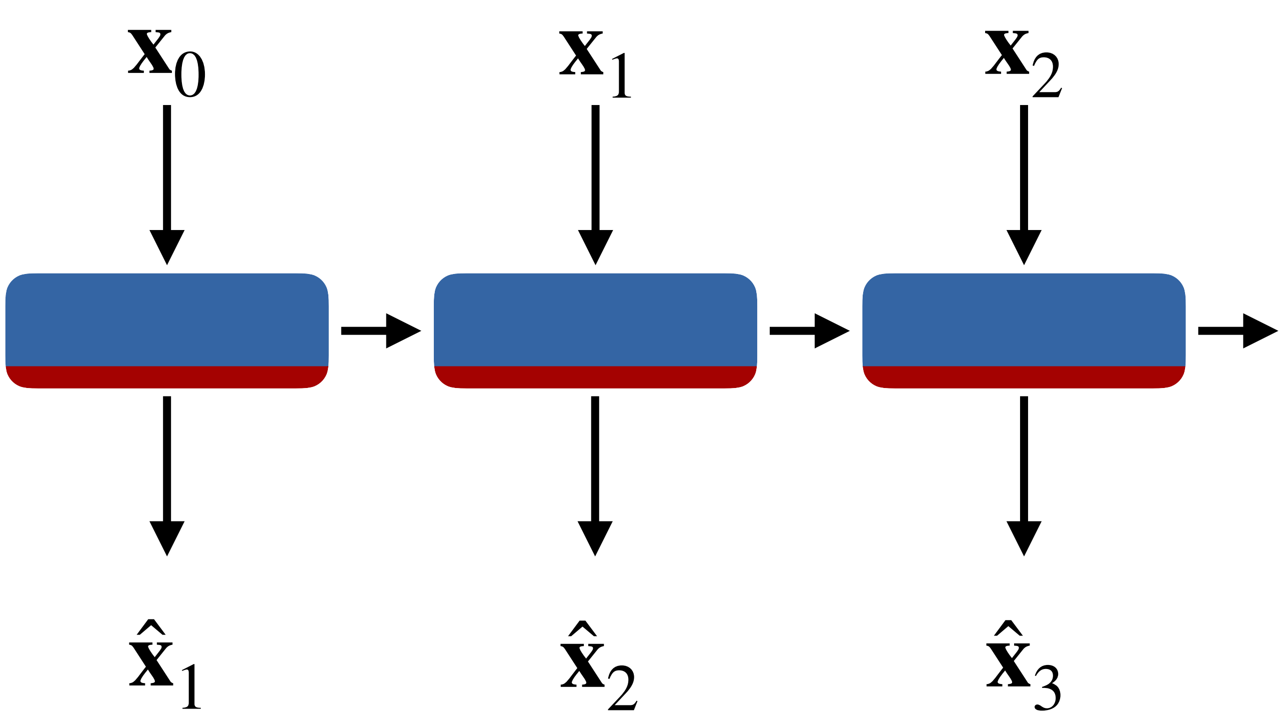
Vanishing and exploding gradients - Simple example

- Linear RNN
  - $\mathbf{h}_t = \alpha \mathbf{h}_{t-1} + \mathbf{x}_t$
- For large t
  - $a > 1 \quad \frac{\partial}{\partial \mathbf{x}_0} \mathbf{h}_t = \alpha^t \to \infty$
  - $a < 1 \quad \frac{\partial}{\partial \mathbf{x}_0} \mathbf{h}_t = \alpha^t \to 0$



Preventing vanishing and exploding gradients

- Generative models
  - Use ground truth inputs



Preventing vanishing and exploding gradients

- Exploding gradients
  - Gradient clipping

$$\nabla \ell' = \nabla \ell \min\left(1, \frac{\epsilon}{|\nabla \ell|}\right)$$

- Vanishing gradients
  - Different RNN structure