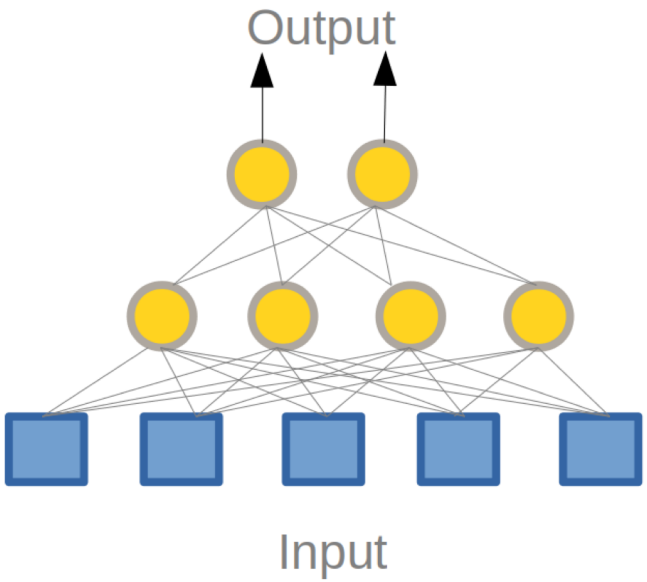
Recurrent Neural Networks

Heni Ben Amor, Ph.D.
Assistant Professor
Arizona State University

Ira A. Fulton Schools of Engineering
Arizona State University

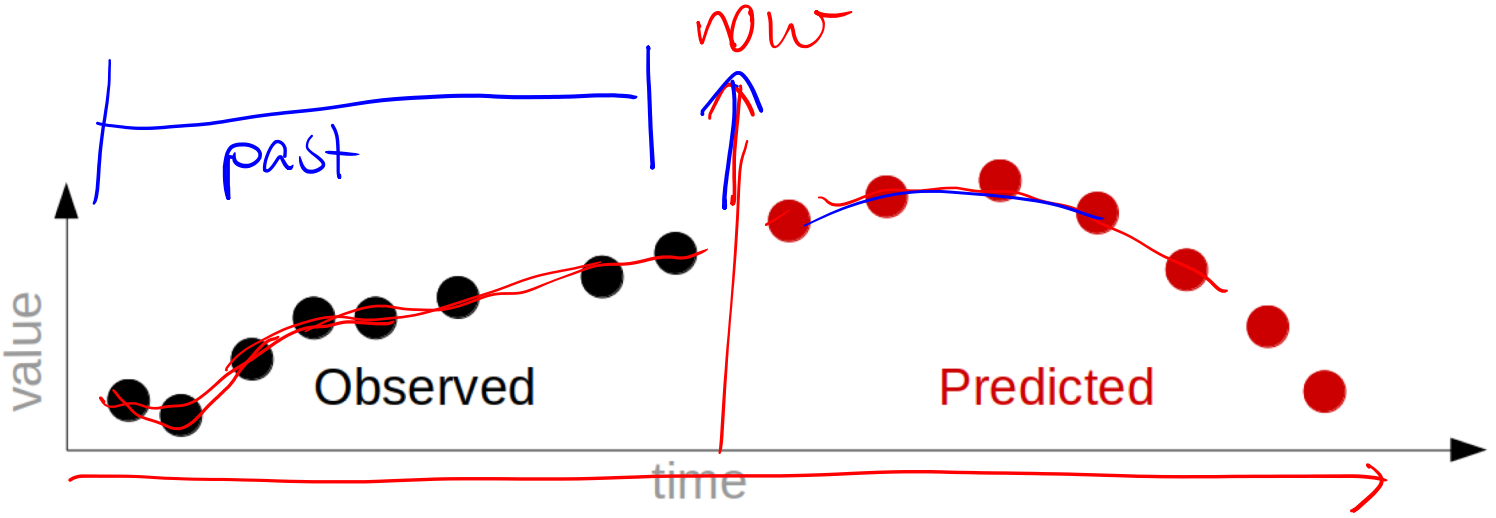
Feed-Forward Neural Networks

- Hierarchy of neurons
- Input layer, hidden layers, output layer
- Does not have a memory
- Independent of last decision



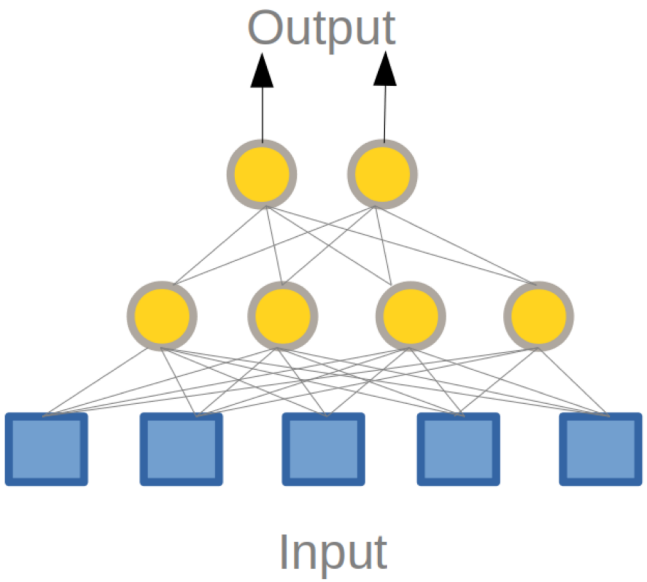
Tasks with Temporal Aspects

- Cause and effect
- Events can affect each other in time
- Time-series prediction such as:
  - Predicting a value of a stock
  - Predicting the likelihood of an earthquake
  - Weather forecasts



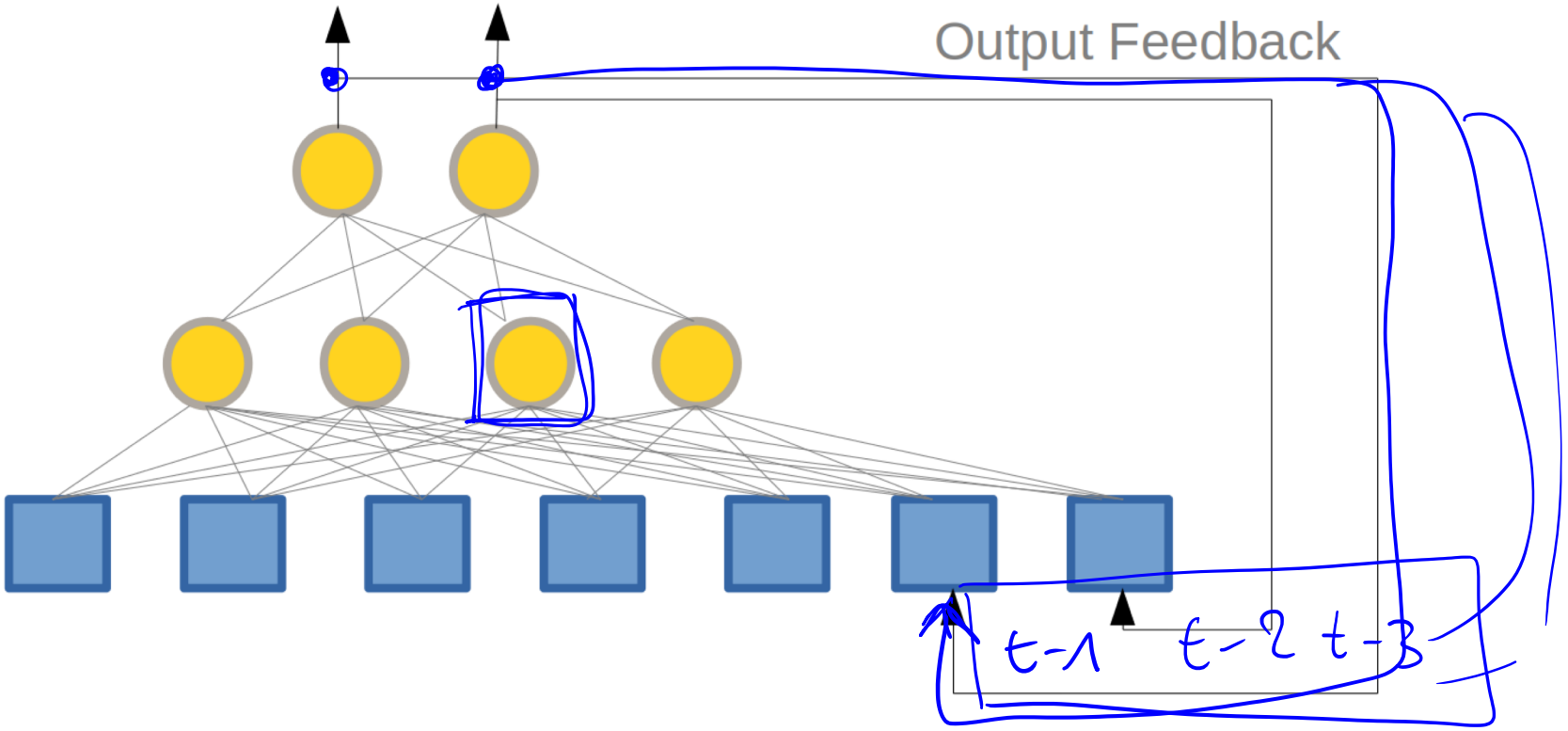
Feed-Forward Neural Networks

- Hierarchy of neurons
- Input layer, hidden layers, output layer
- Does not have a **memory**
- Independent of last decision



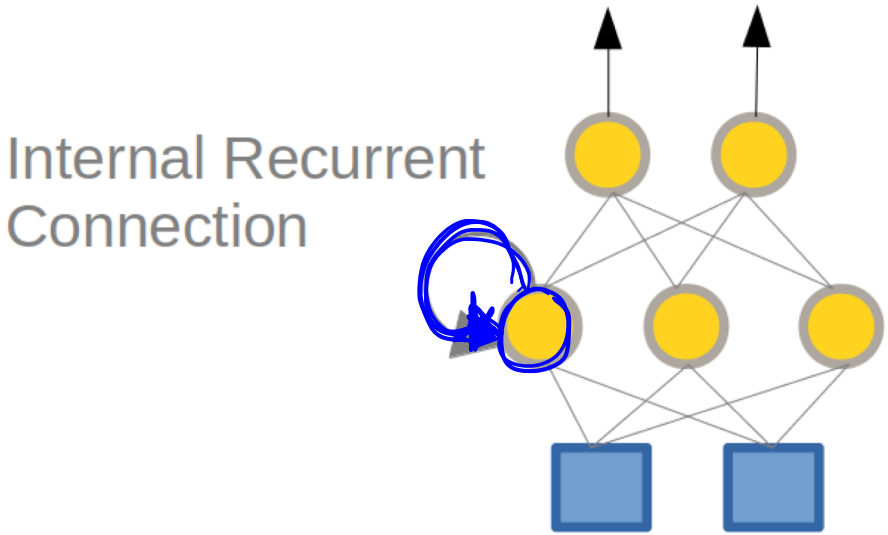
Potential Approach: Feedback Loop

- Keep track of N previous outputs
- Feed them as input to the network



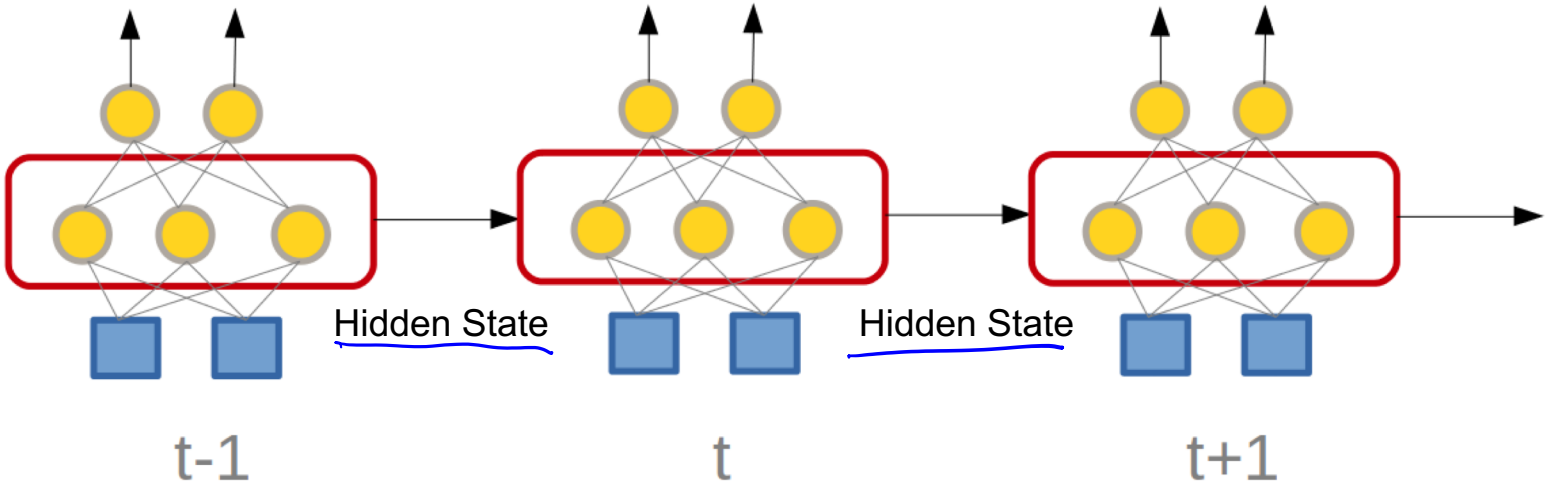
Recurrent Neural Networks

- Memory through recurrent connections
- Feedback information from last time step
- Internal loops within the network
- Propagate forward hidden state



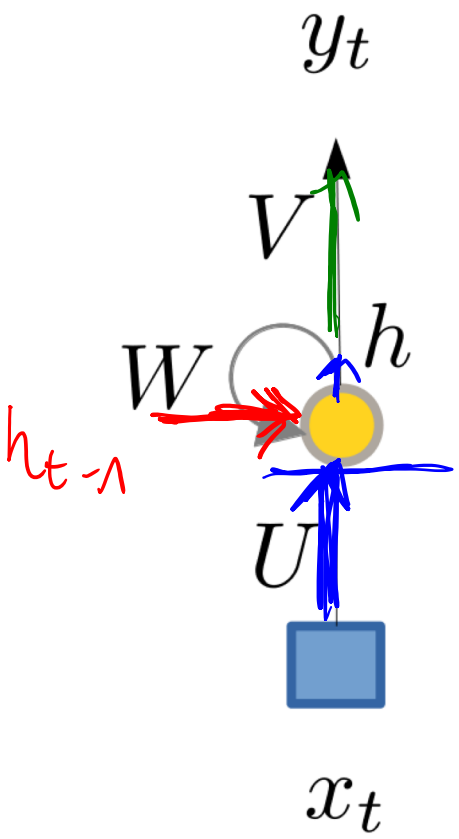
A Different Perspective

- RNNs can be seen as multiple ANNs communicating in time
- Internal states are shared in time
- Ideal for sequence learning
  - (text, music, video)
- **Time-series prediction**



Modeling Recurrency in RNN

- Elements of a single recurrent unit
- Equations underlying the computations



now + past

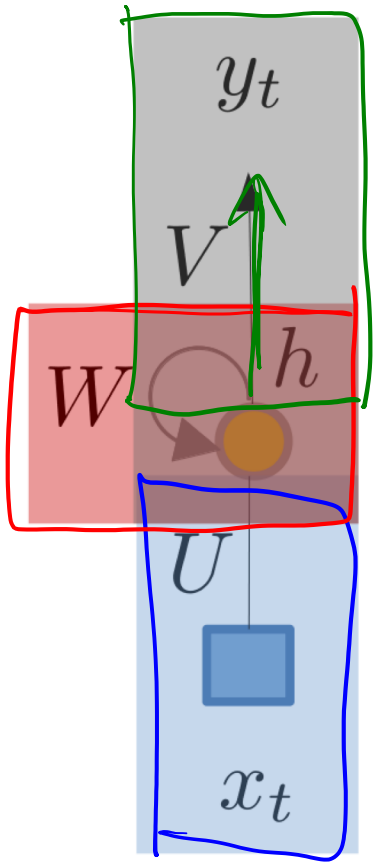
$$h_t = \sigma(Ux_t + Wh_{t-1})$$

weight

$$y_t = \phi(Vh_t)$$

Modeling Recurrency in RNN

- Elements of a single recurrent unit
- Equations underlying the computations



Generating Network Output

Incorporating Previous Hidden State

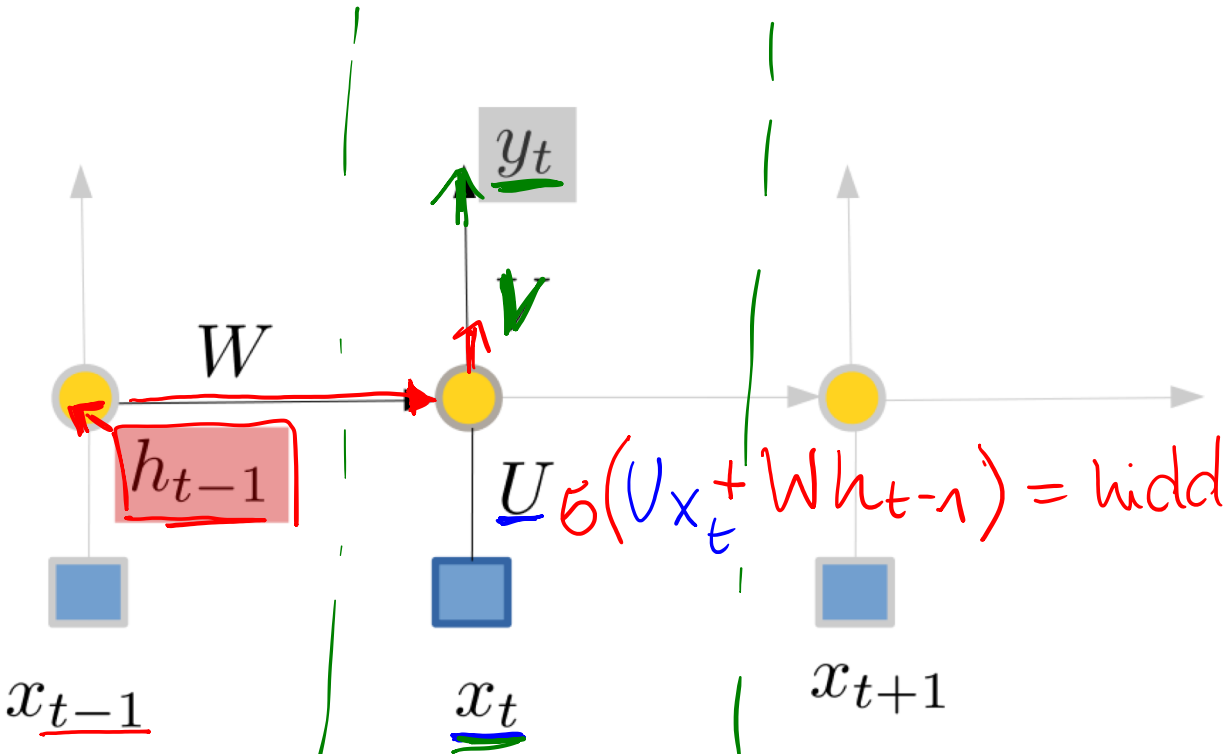
Processing the input

$$h_t = \sigma\left(Ux_t + Wh_{t-1}\right)$$

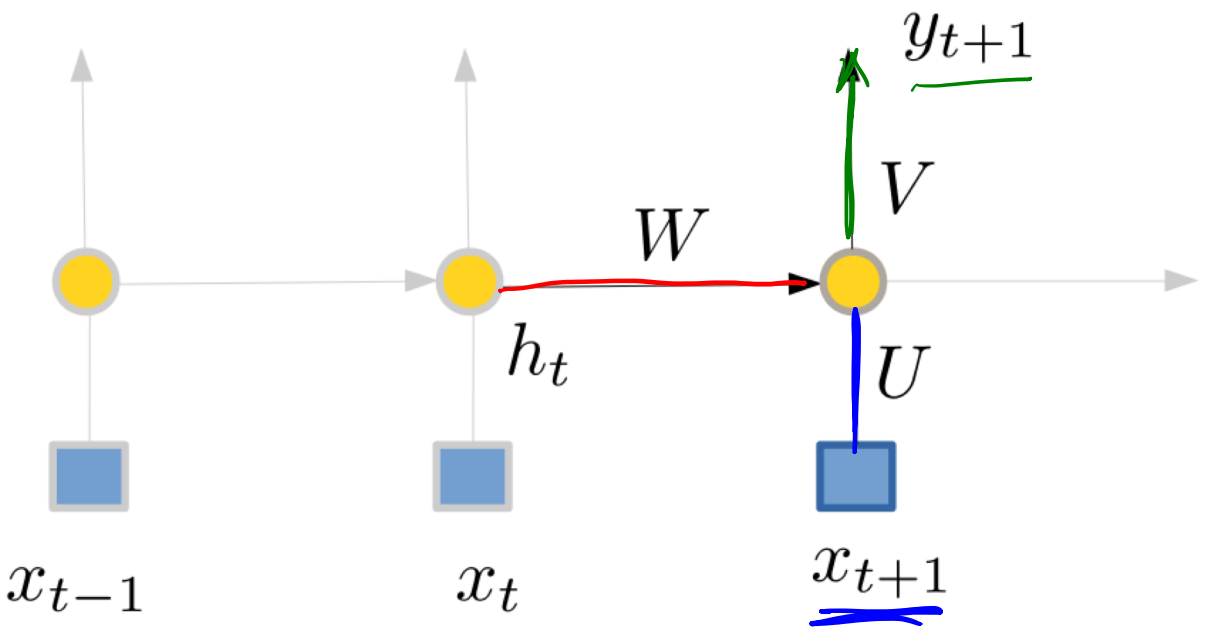
$$y_t = \text{softmax}\left(Vh_t\right)$$

softmax

Neurons in Recurrent Neural Networks



Neurons in Recurrent Neural Networks



Forward Propagation of Hidden State

- How does information travel through the network?
- Output $y_t = \text{softmax}(Vh_t)$
- What happens to the hidden state?

Hidden State at time step *t*: $\sigma(Ux_t + Wh_{t-1})$

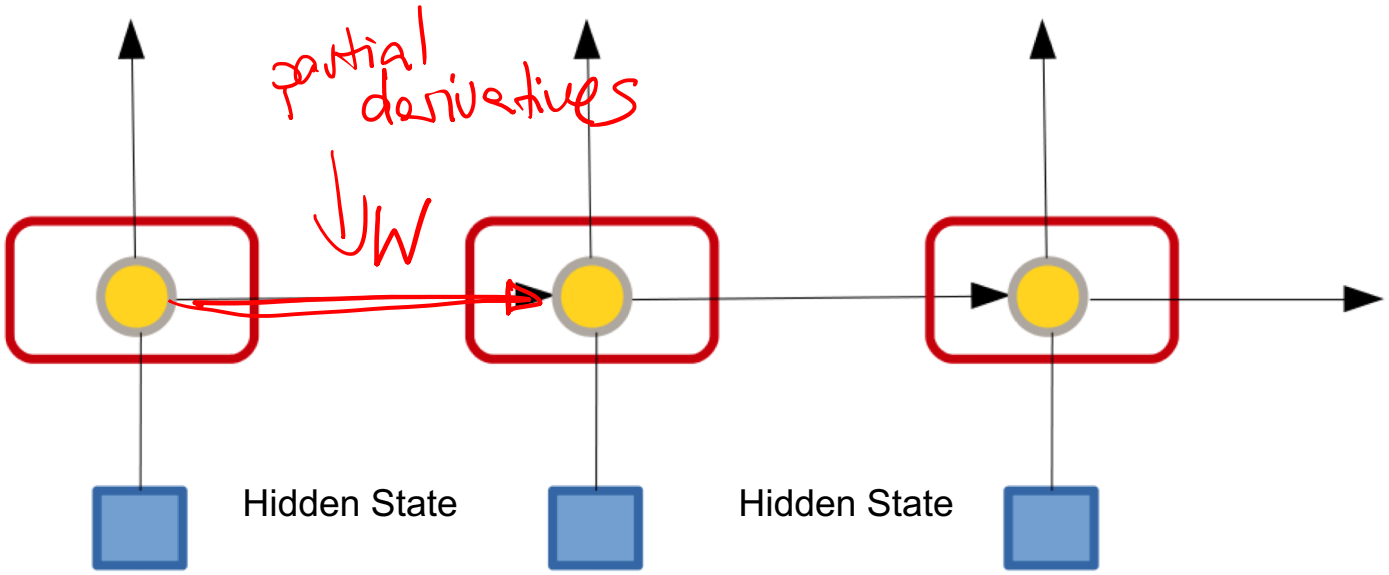
Hidden at *t+1*: $\sigma(Ux_{t+1} + W(\sigma(Ux_t + Wh_{t-1})))$

Hidden at *t+2*:

$$\sigma(Ux_{t+2} + W\sigma(Ux_{t+1} + W(\sigma(Ux_t + Wh_{t-1}))))$$

Back Propagation (BP) through Time

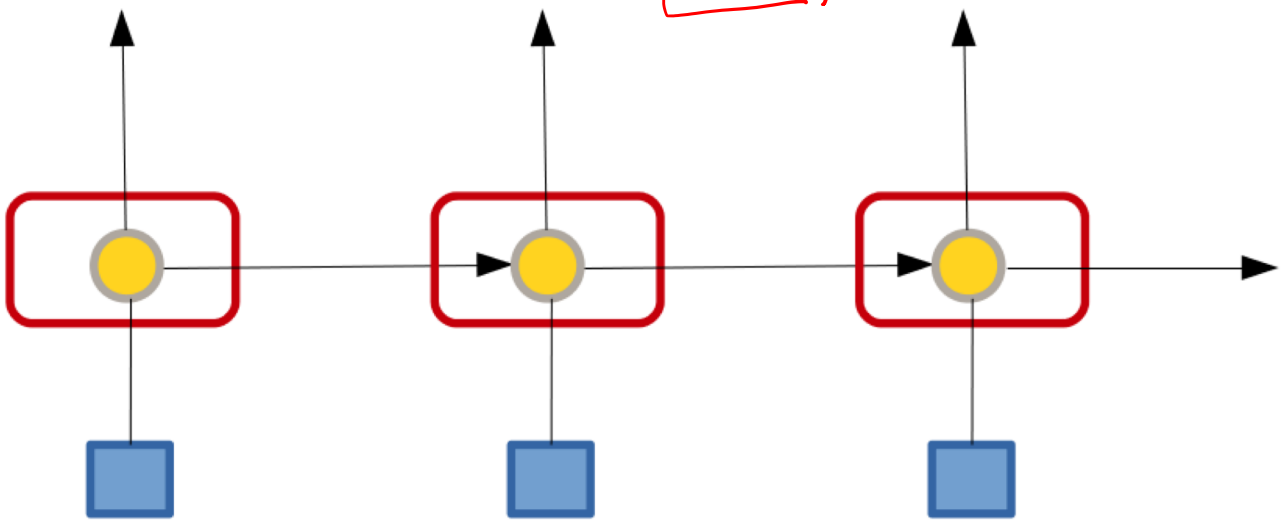
- The same as BP
- Use **unfolded** network



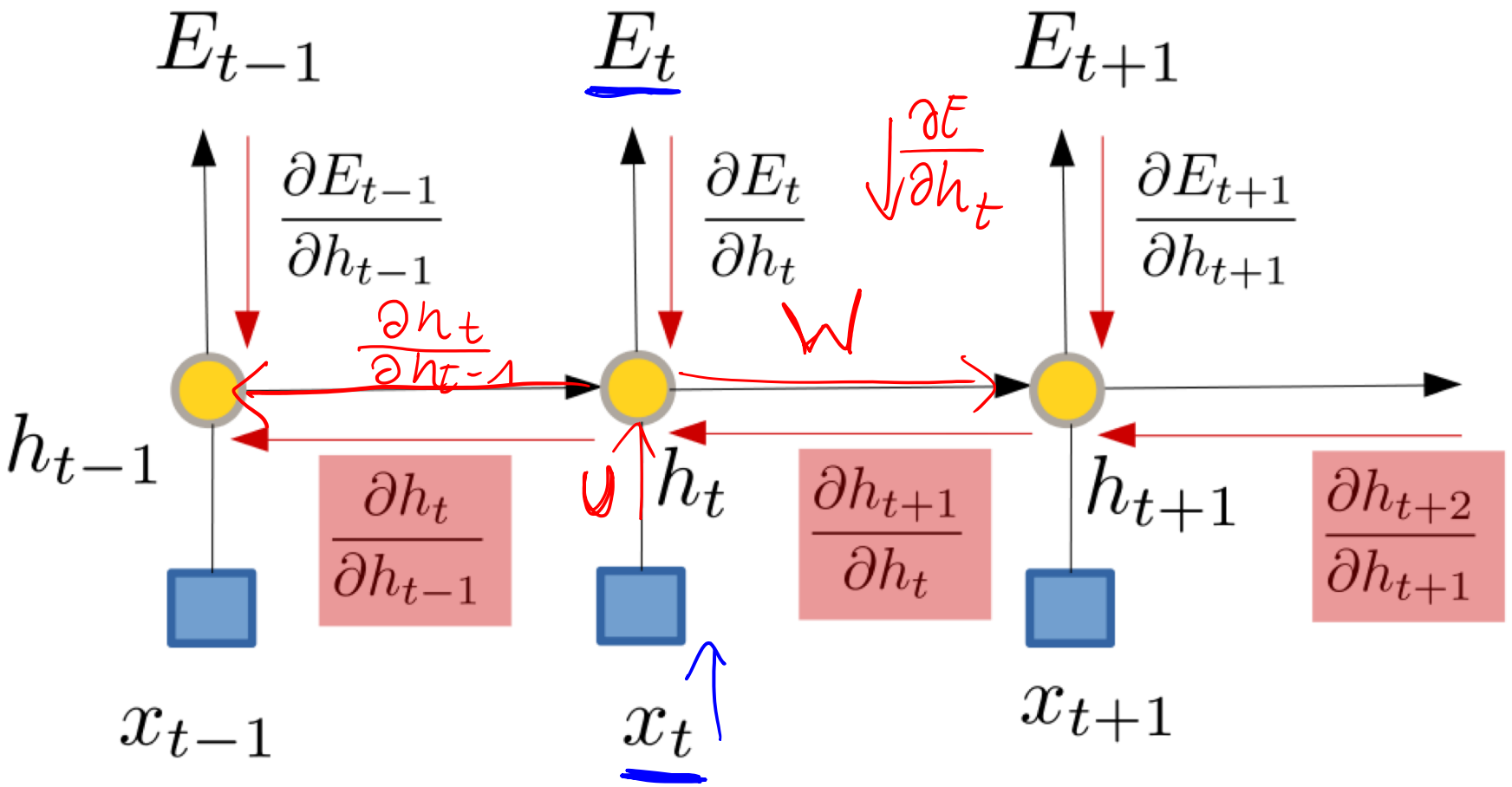
Loss Functions

- Each sample has multiple time steps
- Error function:

$$E = \frac{1}{2} \sum_{t=1}^{T} \sum_{i=1}^{N} ||y_t^{(i)} - g_t^{(i)}||^2$$

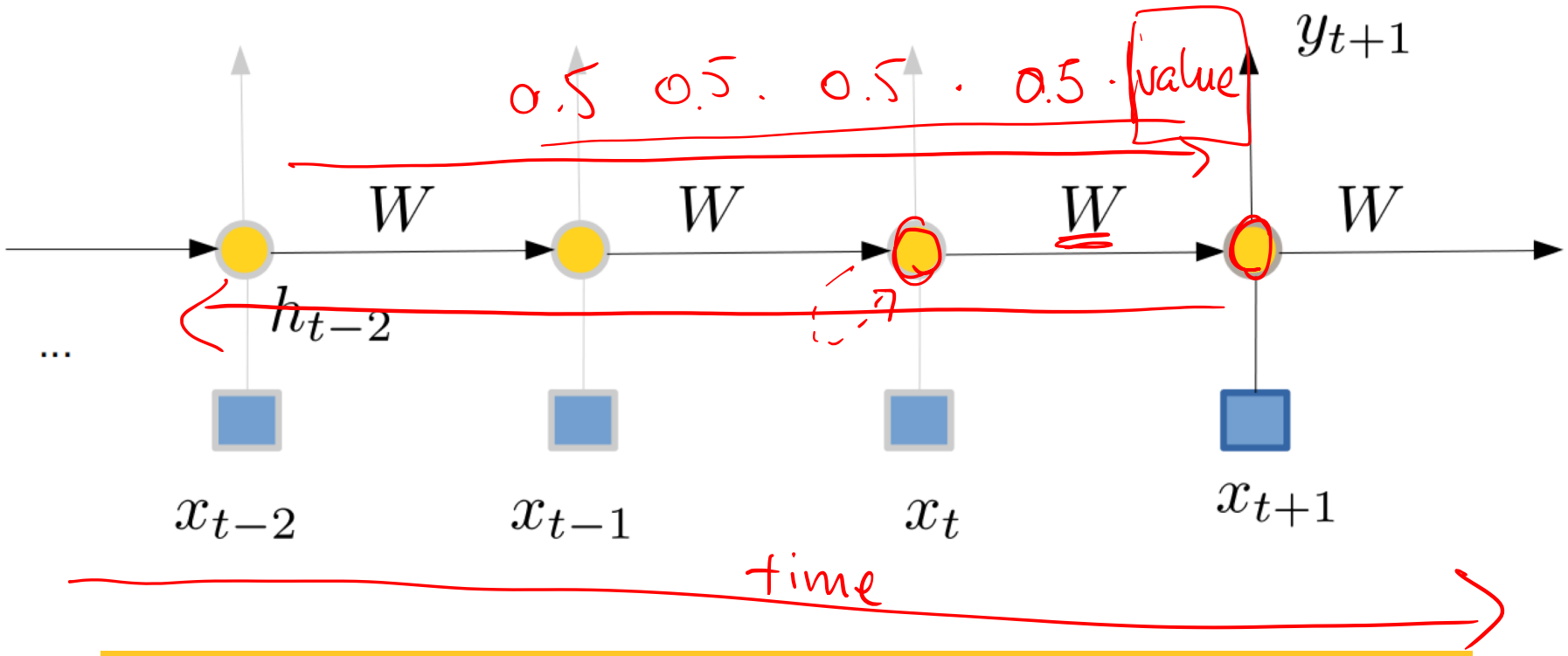


Back Propagation Through Time (BPTT)



Vanishing Gradients

- We focus on the temporal connections



- **What happens when W is smaller than 1?**

Vanishing and Exploding Gradients

- Challenge when training RNNs
- Gradients quickly shrink to negligible values
- Or, gradients may grow substantially and make learning unstable
- An immediate result of the **temporal connections**
- **Exponential growth in hidden state values**
- Effect: learning is slow and yields poor results