Unfolding a recursive or recurrent computation into a computational graph

- results in the sharing of parameters across a deep network structure.
- consider the classical form of a dynamical system

$$s(t) = f(s(t-1); \vartheta) \quad \text{---(1)}$$

where *s*(*t*) is called the state of the system.

- Equation 1 is recurrent because the definition of *s* at time *t* refers back to the same definition at time $t - 1$.

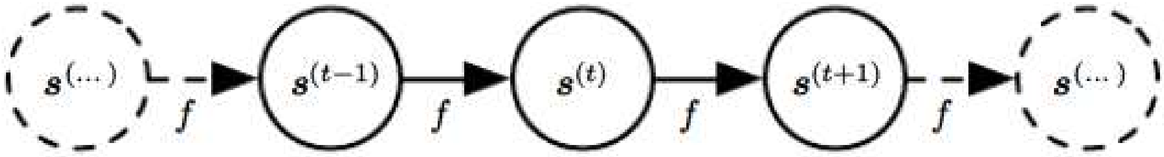
- For a finite number of time steps τ , the graph can be unfolded by applying the definition $\tau - 1$ times.
- For example, if we unfold equation 10.1 for $\tau = 3$ time

steps, we obtain

$$s(3) = f(s(2) ; \vartheta) \text{ ----(2)}$$

$$= f(f(s(1); \vartheta); \vartheta) \text{ -----(3)}$$

- Unfolding the equation by repeatedly applying the definition in this way has yielded an expression that does not involve recurrence.
- The unfolded computational graph of equation 10.1 and equation 10.3 is illustrated in the form of a traditional directed acyclic computational graph



- Each node represents the state at some time t and the function f maps the state at t to the state at $t + 1$. The same parameters (the same value of ϑ used to parametrize f) are used for all time steps.

- Example: a dynamical system driven by an external signal $x(t)$ will be represented as

$$s(t) = f(s(t-1)\ ,\ x(t);\ \vartheta)\ \text{---}(4)$$

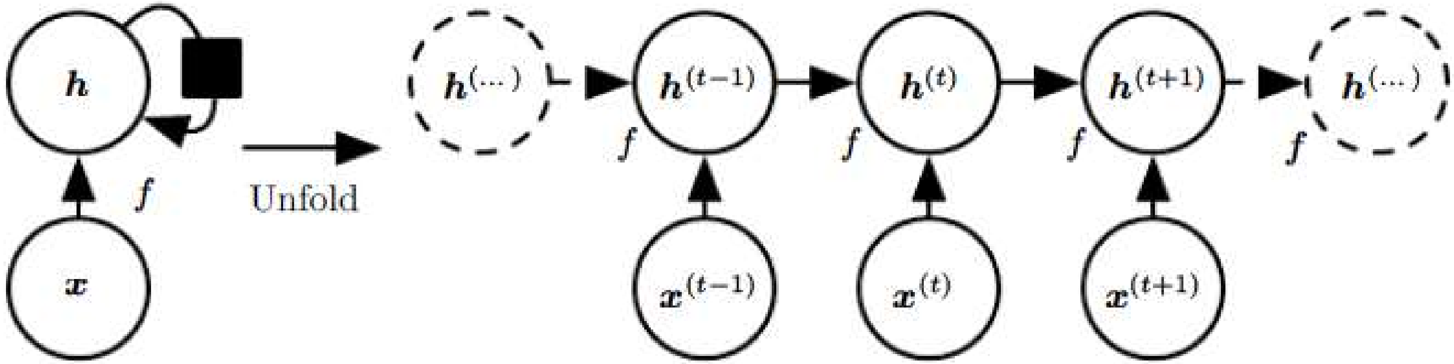
- Many recurrent neural networks use equation (5) or a similar equation to define the values of their hidden units

$$h(t) = f(h(t-1)\ ,\ x(t);\ \vartheta)\quad \text{-----}(5)$$

- typical RNNs will add extra architectural features such as output layers that read information out of the state *h* to make predictions

- $h(t)$ is a kind of lossy summary of the task-relevant aspects of the past sequence of inputs up to t.
- This summary is in general necessarily lossy, since it maps an arbitrary length sequence
- $(x(t), x(t-1) , x(t-2), . . . , x(2) , x(1))$ to a fixed length vector $h(t)$.
- Depending on the training criterion, this summary might selectively keep some aspects of the past sequence with more precision than other aspects.

A recurrent network with no outputs



- This recurrent network just processes information from the input *x* by incorporating it into the state *h* that is passed forward through time.
- *(Left)*Circuit diagram. The black square indicates a delay of a single time step.
- *(Right)*The same network seen as an unfolded computational graph, where each node is now associated with one particular time instance.