Recurrent Neural Networks

RNNs

1. RNNs are a neural net architecture for modeling sequences—they perform sequential computation (rather than parallel computation)
2. They solve the following problem: What if we want our computations in the past to influence our computations in the future?
3. The recurrent feedback loops in an RNN are a kind of memory.

RNNs

1. RNN can potentially remember information over arbitrarily long time intervals.
2. In fact, RNNs are Turing complete: you can think of them as a little computer.
3. They can do anything a computer can do.

RNNs

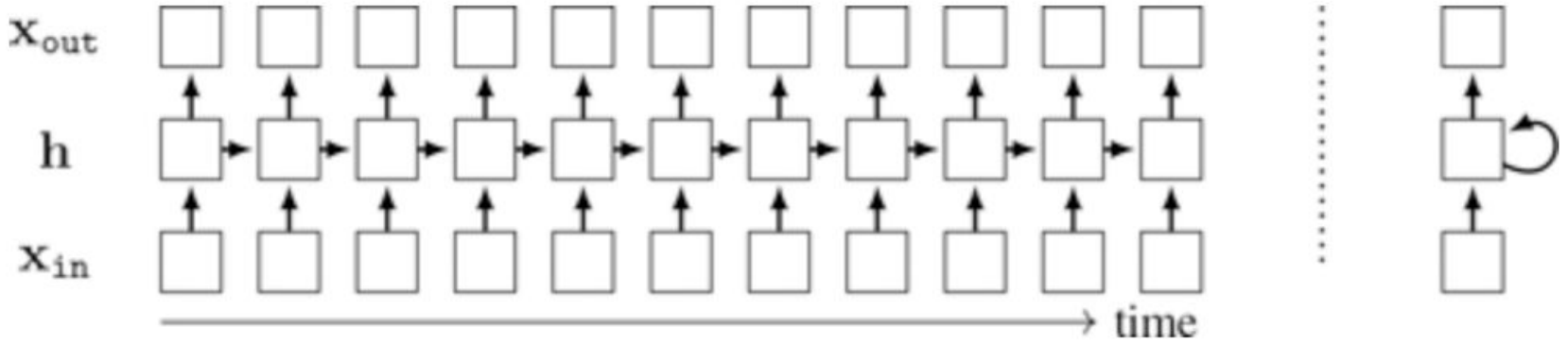


Figure 25.3: (left) RNN with multidimensional inputs, hidden states, and outputs. (right) the rolled-up version of this RNN, which can be unrolled for an unbounded number of steps.

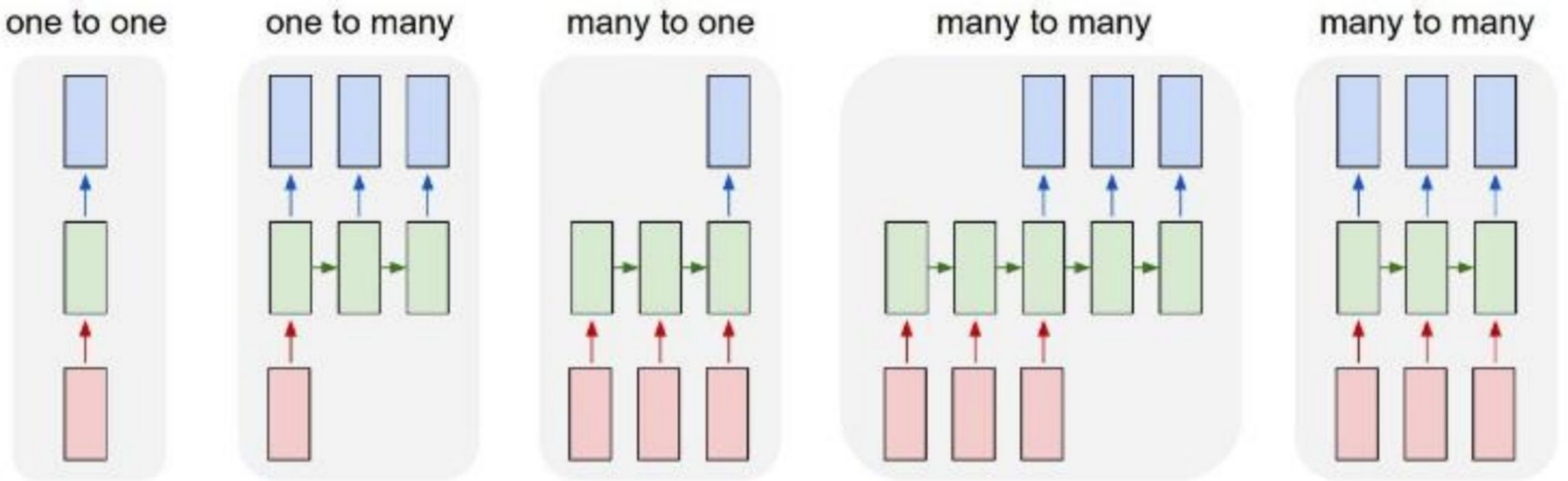
RNNs

1. Typically we will work with unrolled graphs because they are directed acyclic graphs (DAGs), and therefore all the tools we have developed for DAGs, including backpropagation, will extend naturally to them.
2. Backpropagation is only defined for DAGs and RNNs are not DAGs. To apply backpropagation to RNNs, we unroll the net for T timesteps
3. This is known as backpropagation through time (BPTT)

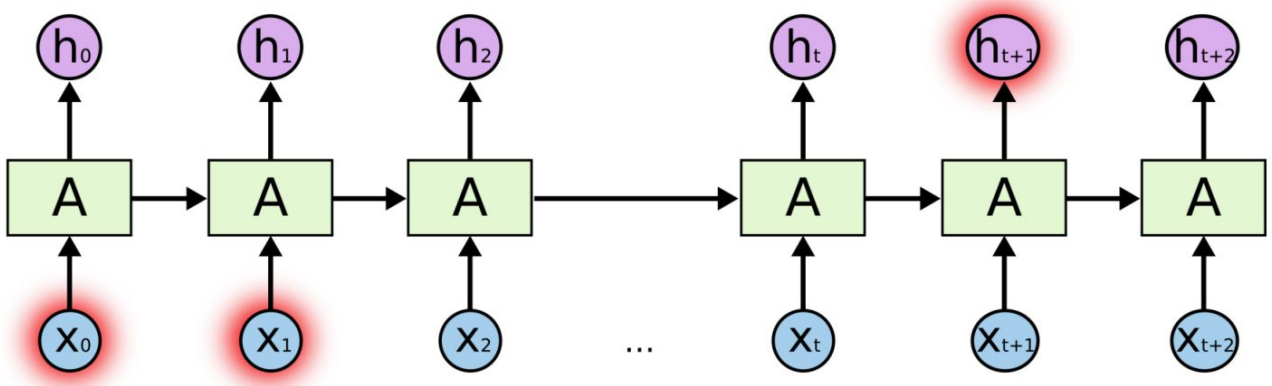
RNNs

1. Notice that the we can get very large computation graphs when we run an RNN for many timesteps
2. If values in W are large, the gradient will explode. If they are small, the gradient will vanish
3. Basically, this system is not numerically well-behaved.

Recurrent Neural Networks: Process Sequences



One to one RNN: Difficult to learn long-term associations



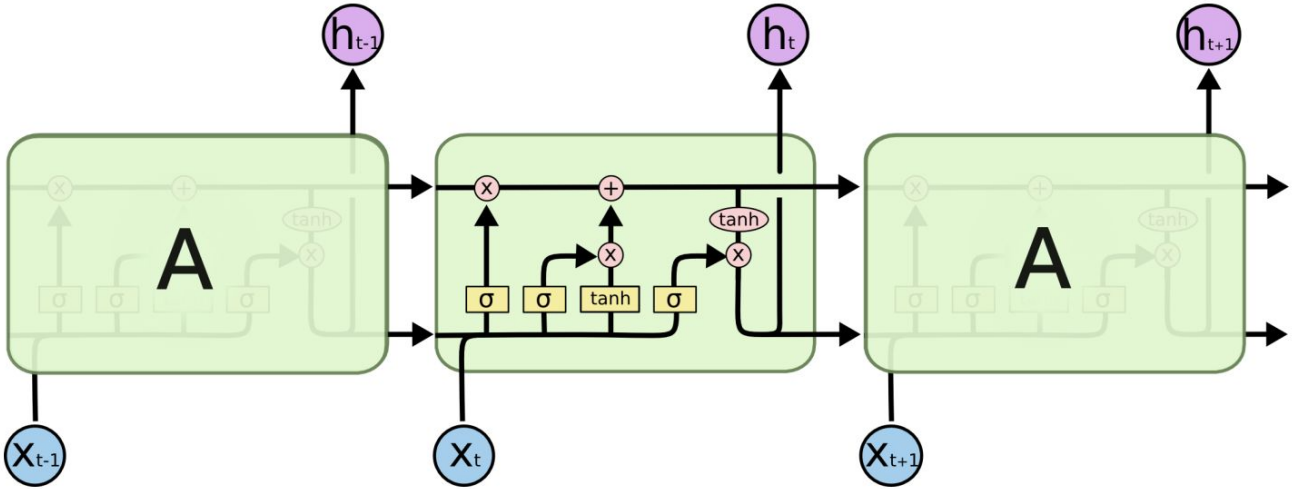
Long Short-Term Memory

1. Long Short-Term Memory layers (LSTMs) are a special kind of recurrent module, that is designed to avoid the vanishing and exploding gradients problem
2. An LSTM has a cell state, c, which is a vector it can read from and write to.
3. The cell state can record memories and retrieve them as needed.
4. The cell state is built to store persistent memories that can last a long time (hence the name long short-term memory)

Long Short-Term Memory

1. The cell state is updated based on the incoming signals, h_{t-1} and $x_{in}[t]$.
2. Then the cell state is used to determine the hidden state h_t to output.
3. Different subcomponents of the LSTM layer decide **what to delete** from the cell state, **what to write** to the cell state, and **what to read** off the cell state.

LSTM(Long Short-Term Memory



References

1. Foundations of Computer Vision - Chapter 25