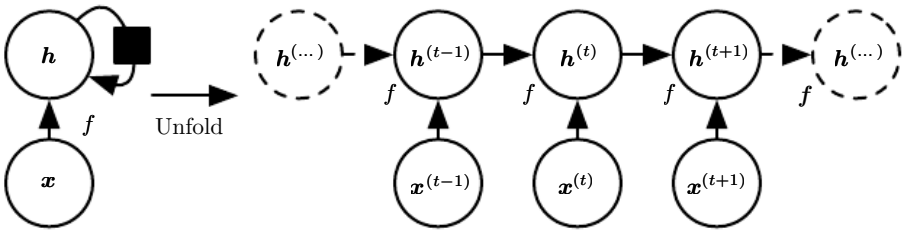
Recurrent Neural Network

CHEN Si

Contents

1 Definition



$$\boldsymbol{h}^{(t)} = f(\boldsymbol{h}^{(t-1)}, \boldsymbol{x}^{(t)}; \boldsymbol{\theta})$$

- Hidden units $\boldsymbol{h}$ represents

  1. the state of the dynamical system.

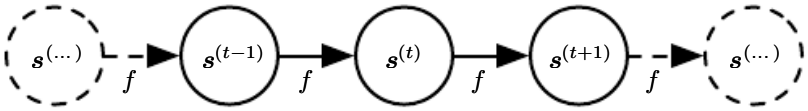
2. lossy summary of the task-relevant aspects of the past sequence of inputs up to t, when the recurrent network is trained to perform a task that requires predicting the future from the past.

- Black square indicates that an interaction takes place with a delay of a single time step.

1.1 Motivation

Dynamical System

Unfolded Computational Graph



where $\boldsymbol{s}^{(t)}$ is called the **state** of the system.

Recurrent Defintion

$$\boldsymbol{s}^{(t)} = f(\boldsymbol{s}^{(t-1)}; \boldsymbol{\theta})$$

Unfolded Definition (up to step 3)

$$\begin{aligned} \boldsymbol{s}^{(3)} &= f(\boldsymbol{s}^{(2)}; \boldsymbol{\theta}) \\ &= f(f(\boldsymbol{s}^{(1)}; \boldsymbol{\theta}); \boldsymbol{\theta}) \end{aligned}$$

Dynamical System with External Signal $\boldsymbol{x}^{(t)}$

$$\boldsymbol{s}^{(t)} = f(\boldsymbol{s}^{(t-1)}, \boldsymbol{x}^{(t)}; \boldsymbol{\theta})$$

1.2 Advantages

1. Learning just a single model f:
   - The learned model always has the same input size, regardless of the sequence length.
   - Parameter sharing: It is possible to use the same transition function f with the same parameters at every time step.
   - Allows generalization to sequence lengths that does not appear in the training set.
   - Requires much fewer training examples to estimate.

1.3 Disadvantages

1. Optimizing the parameters may be difficult, because of the reduced number of parameters.

1.4 How to determine the length τ

1. Add a special symbol corresponding to the end of a sequence.
2. Introduce an extra Bernoulli output to the model that represents the decision to either contnue generation or halt generation at each time step. (a more general way)
3. Add an extra output to the model that predicts the interger τ itself.

$$P(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(\tau)}) = P(\tau)P(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(\tau)} \mid \tau)$$

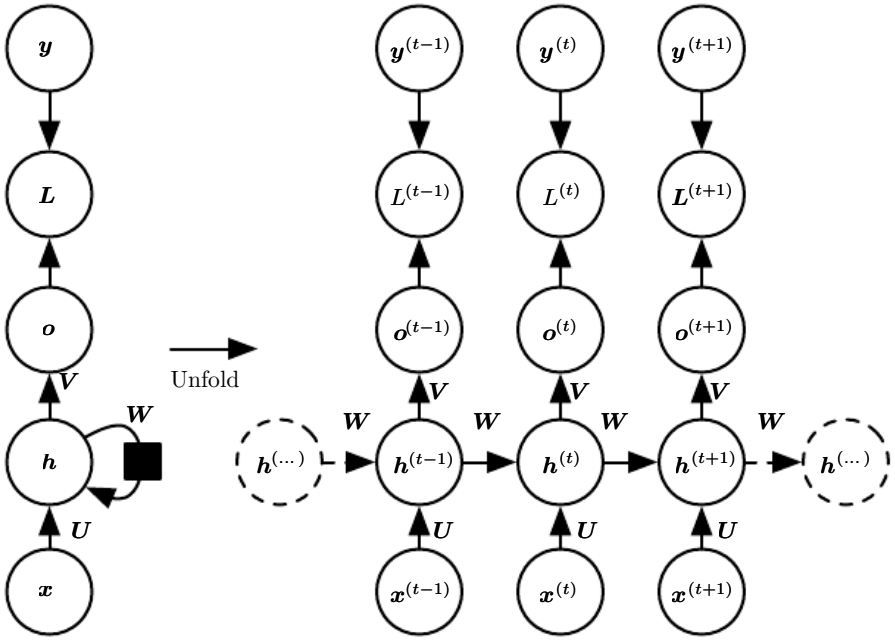
2 Taxonomy

2.1

Recurrent networks that produce an output at each time step and have recurrent connections between hidden units.

$$\begin{aligned} \boldsymbol{a}^{(t)} &= \boldsymbol{b} + \boldsymbol{W}\boldsymbol{h}^{(t-1)} + \boldsymbol{U}\boldsymbol{x}^{(t)} \\ \boldsymbol{h}^{(t)} &= \tanh(\boldsymbol{a}^{(t)}) \\ \boldsymbol{o}^{(t)} &= \boldsymbol{c} + \boldsymbol{V}\boldsymbol{h}^{(t)} \\ \hat{\boldsymbol{y}}^{(t)} &= \text{softmax}(\boldsymbol{o}^{(t)}) \end{aligned} \tag{1}$$

with a initial state $\boldsymbol{h}^{(0)}$.



Characteristics

- Maps an input sequence to an output sequence of the same length.
- Loss:

$$\begin{aligned} & L\left(\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(\tau)}\}, \{\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(\tau)}\}\right) \\ =& \sum_t L^{(t)} \\ =& -\sum_t \log p_{\text{model}}\left(y^{(t)} \mid \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(t)}\}\right) \end{aligned} \tag{2}$$

  where $L^{(t)}$ is the negative log-likelihood of $y^{(t)}$ given $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(t)}$

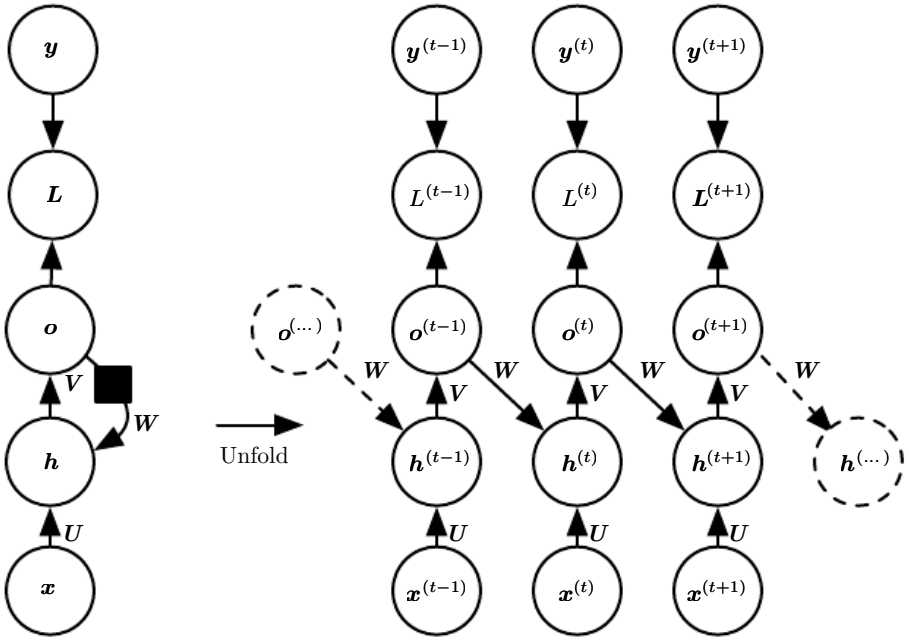
  - Conditional independence assumption: The conditional distribution

$$P(\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(\tau)} \mid \boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(\tau)}) = \prod_t P(\boldsymbol{y}^{(t)} \mid \boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(t)})$$

- Runtime is $O(\tau)$, memory cost is $O(\tau)$.
- Only able to represent distributins in which the $\boldsymbol{y}$ values are conditionally independent from each other given the $\boldsymbol{x}$ values.

2.2

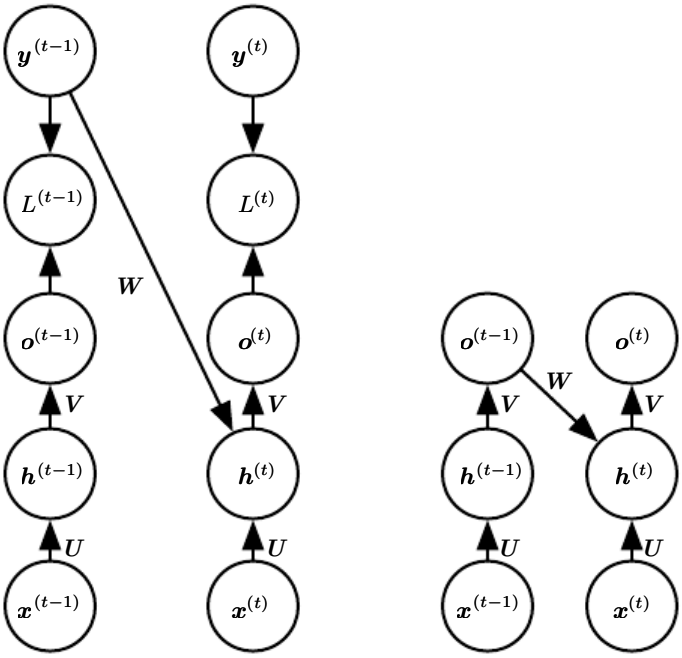
Recurrent networks that produce an output at each time step and have recurrent connections only from the output at one time step to the hidden units at the next time step.



Characteristics

- Strictly less powerful because it lacks hidden-to-hidden recurrent connections. (e.g. it connot simulate a universal Turing machine.)
- Training can be **parallelized** because all the time steps are decoupled (the training set provides the ideal value of previous output).
- Avoids back-propagation through time.
- Teacher forcing: For example, for a sequence with two time steps

$$\begin{aligned} &\log p\left(\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)} \mid \boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}\right) \\ =&\log p\left(\boldsymbol{y}^{(2)} \mid \boldsymbol{y}^{(1)}, \boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}\right)+\log p\left(\boldsymbol{y}^{(1)} \mid \boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}\right) \end{aligned}$$



Disadvantages

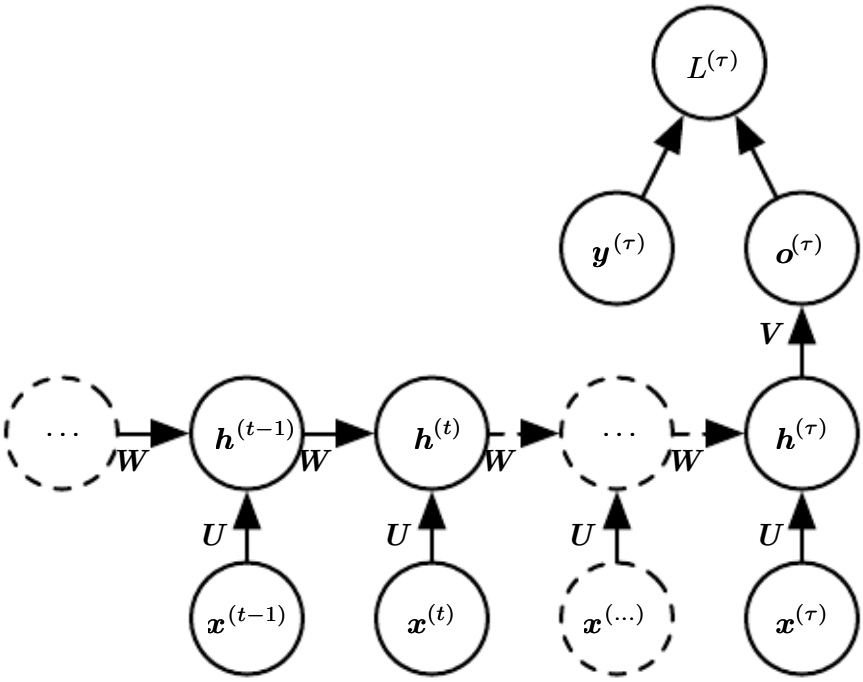
1. The disadvantages of strict teacher forcing (no BPTT) is that the inputs during training could be quite different from the inputs during testing (**closed-loop** mode).
   Solutions:

   (a) Train with both teacher-forced inputs and free-running inputs

   (b) Randomly chooses to use generated values or actual data values as input (curriculum learning strategy: gradially use more of the generated values as input).

2.3

Recurrent networks with recurrent connections between hidden units, that read an entire sequence and then produce a single output.



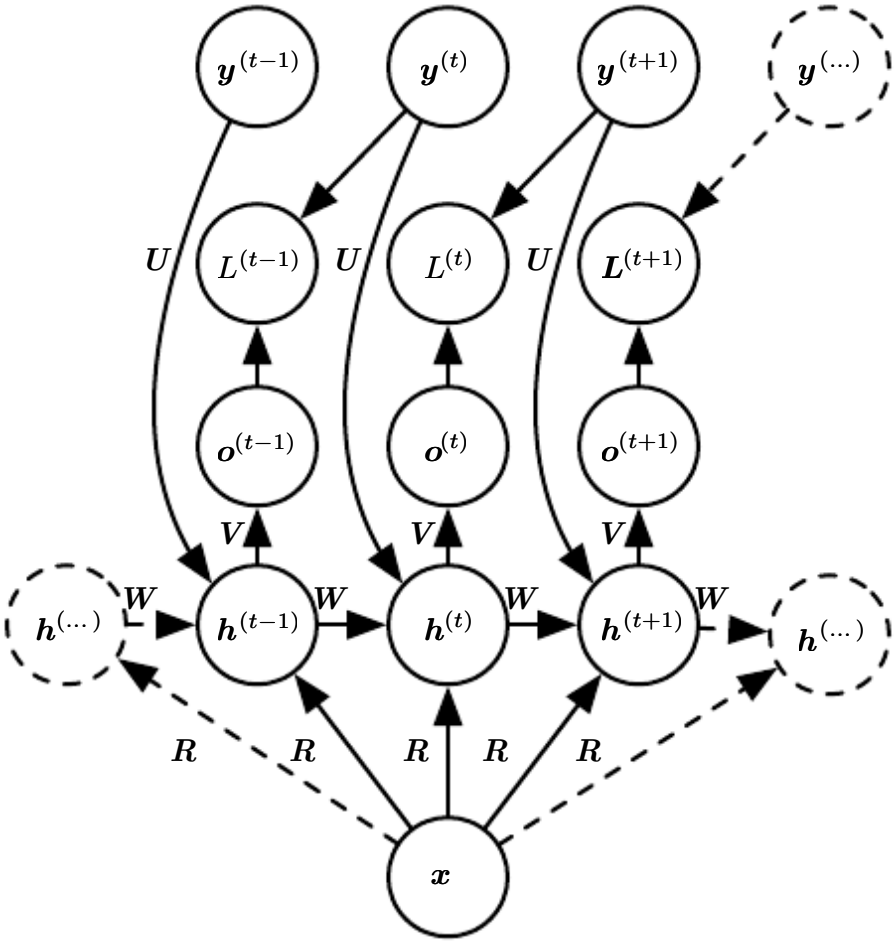
Characteristics

- There might be a target right at the end, or the gradient on the ouput $\boldsymbol{o}^{(t)}$ can be obtained by back-propagating from further downstream modules.

Applications

1. Summarize a sequence and produce a fixed-size representation used as input for further processing.

2.4



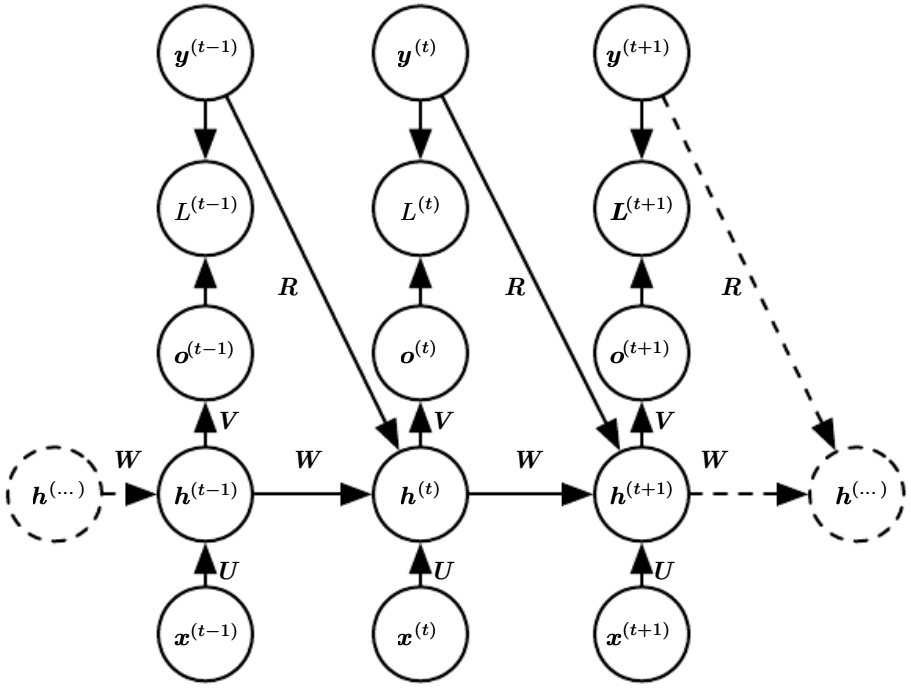
Characteristics

- $\boldsymbol{x}^\top \boldsymbol{R}$ is effectively a new bias parameter.
- Each element $\boldsymbol{y}^{(t)}$ of the observed output sequence serves both as input (for the current time step) and, during training, as target (for the previous time step).

Applications

1. Image captioning, where a single image is used as input to a model that then produces a sequence of words describing the image.

2.5



3 Back Propagation Through Time (BPTT)

Here we talk about the BPTT of equation 1 as forward propagation and equation 2 as loss function. (Note: here $\hat{\boldsymbol{y}}^{(t)} = \text{softmax}(\boldsymbol{o}^{(t)})$)

1. $\dfrac{\partial L}{\partial L^{(t)}} = 1$

2. $\left(\nabla_{\boldsymbol{o}^{(t)}} L\right)_i = \dfrac{\partial L}{\partial o_i^{(t)}} = \dfrac{\partial L}{\partial L^{(t)}} \dfrac{\partial L^{(t)}}{\partial o_i^{(t)}} = \hat{y}_i^{(t)} - \mathbb{1}_{i=y^{(t)}}$

3. For $t = \tau$

$$\nabla_{\boldsymbol{h}^{(\tau)}} L = \boldsymbol{V}^\top \nabla_{\boldsymbol{o}^{(\tau)}} L$$

$\forall t = 1, \dots, \tau - 1$

$$\begin{aligned}\nabla_{\boldsymbol{h}^{(t)}} L &= \left(\frac{\partial \boldsymbol{h}^{(t+1)}}{\partial \boldsymbol{h}^{(t)}}\right)^\top \left(\nabla_{\boldsymbol{h}^{(t+1)}} L\right) + \left(\frac{\partial \boldsymbol{o}^{(t)}}{\partial \boldsymbol{h}^{(t)}}\right)^\top \left(\nabla_{\boldsymbol{o}^{(t)}} L\right) \\ &= \boldsymbol{W}^\top \text{diag}\left(1 - \left(\boldsymbol{h}^{(t+1)}\right)^2\right) \left(\nabla_{\boldsymbol{h}^{(t+1)}} L\right) + \boldsymbol{V}^\top \left(\nabla_{\boldsymbol{o}^{(t)}} L\right)\end{aligned}$$

4.

$$
\begin{aligned}
\nabla_{\boldsymbol{c}} L &= \sum_t \left(\frac{\partial \boldsymbol{o}^{(t)}}{\partial \boldsymbol{c}^{(t)}}\right)^\top \nabla_{\boldsymbol{o}^{(t)}} L = \sum_t \nabla_{\boldsymbol{o}^{(t)}} L \\
\nabla_{\boldsymbol{b}} L &= \sum_t \left(\frac{\partial \boldsymbol{h}^{(t)}}{\partial \boldsymbol{b}^{(t)}}\right)^\top \nabla_{\boldsymbol{h}^{(t)}} L = \sum_t \operatorname{diag}\left(1 - \left(\boldsymbol{h}^{(t)}\right)^2\right) \nabla_{\boldsymbol{h}^{(t)}} L \\
\nabla_{\boldsymbol{V}} L &= \sum_t \sum_i \frac{\partial L}{\partial o_i^{(t)}} \nabla_{\boldsymbol{V}^{(t)}} o_i^{(t)} = \sum_t (\nabla_{\boldsymbol{o}^{(t)}} L) \boldsymbol{h}^{(t)\top} \\
\nabla_{\boldsymbol{W}} L &= \sum_t \sum_i \frac{\partial L}{\partial h_i^{(t)}} \nabla_{\boldsymbol{W}^{(t)}} h_i^{(t)} \\
&= \sum_t \operatorname{diag}\left(1 - \left(\boldsymbol{h}^{(t)}\right)^2\right) (\nabla_{\boldsymbol{h}^{(t)}} L) \boldsymbol{h}^{(t-1)\top} \\
\nabla_{\boldsymbol{U}} L &= \sum_t \sum_i \frac{\partial L}{\partial h_i^{(t)}} \nabla_{\boldsymbol{U}^{(t)}} h_i^{(t)} \\
&= \sum_t \operatorname{diag}\left(1 - \left(\boldsymbol{h}^{(t)}\right)^2\right) (\nabla_{\boldsymbol{h}^{(t)}} L) \boldsymbol{x}^{(t)\top}
\end{aligned}
$$