RNNs as PGMs

Sargur Srihari
srihari@buffalo.edu

This is part of lecture slides on Deep Learning: http://www.cedar.buffalo.edu/~srihari/CSE676

Topics in Sequence Modeling

- Overview

1. Unfolding Computational Graphs
2. Recurrent Neural Networks
3. Bidirectional RNNs
4. Encoder-Decoder Sequence-to-Sequence Architectures
5. Deep Recurrent Networks
6. Recursive Neural Networks
7. The Challenge of Long-Term Dependencies
8. Echo-State Networks
9. Leaky Units and Other Strategies for Multiple Time Scales
10. LSTM and Other Gated RNNs
11. Optimization for Long-Term Dependencies
12. Explicit Memory

Topics in Recurrent Neural Networks

0. Overview
1. Teacher forcing for output-to-hidden RNNs
2. Computing the gradient in a RNN
3. RNNs as Directed Graphical Models
4. Modeling Sequences Conditioned on Context with RNNs

Topics

- RNNs as Directed Graphical Models
- Conditioning Predictions on Past Values
- Fully connected model versus RNN

Recurrent Networks as Directed PGMs

- With a predictive log-likelihood training objective, such as

$$L\left(\left\{\boldsymbol{x}^{(1)},..\boldsymbol{x}^{(t)}\right\},\left\{\boldsymbol{y}^{(1)},..\boldsymbol{y}^{(t)}\right\}\right)=\sum_t L^{(t)}=-\sum_t \log p_{model}\left(\boldsymbol{y}^{(t)}\mid\left\{\boldsymbol{x}^{(1)},..\boldsymbol{x}^{(t)}\right\}\right)$$

we train the RNN to estimate the conditional distribution of next sequence element $\boldsymbol{y}^{(t)}$ given past inputs. This may mean that we maximize log-likelihood

$$\log\ p(\boldsymbol{y}^{(t)}|\boldsymbol{x}^{(1)},..\boldsymbol{x}^{(t)})$$

or if the model includes connections from output at one time step to the next

$$\log\ p(\boldsymbol{y}^{(t)}|\boldsymbol{x}^{(1)},..\boldsymbol{x}^{(t)},\ \boldsymbol{y}^{(1)},..\boldsymbol{y}^{(t-1)})$$

Conditioning predictions on past $\boldsymbol{y}$ values

- Decomposing joint probability over sequence of $\boldsymbol{y}$ values as a series of one-step probabilistic predictions

  is one way to capture the full joint distribution across the whole sequence.

1. When we do not feed past $\boldsymbol{y}$ values as inputs that condition the next step prediction,

   directed PGM contains no edges from any $\boldsymbol{y}^{(i)}$ in the past to current $\boldsymbol{y}^{(t)}$.

   In this case outputs $\boldsymbol{y}^{(i)}$ are conditionally independent given the $\boldsymbol{x}$ values

2. When we do feed the actual $\boldsymbol{y}$ values (not their prediction, but the actual observed or generated values) back into the network

   the directed PGM contains edges from all $\boldsymbol{y}^{(i)}$ values in the past to the current $\boldsymbol{y}^{(t)}$ value.

A simple example

- RNN models a sequence of scalar random variables $Y=\{\boldsymbol{y}^{(1)},..\boldsymbol{y}^{(\tau)}\}$ with no additional inputs $\boldsymbol{x}$.
  - The input at time step t is simply the output at time step t-1
  - The RNN then defines a directed PGM over the $\boldsymbol{y}$ variables
- We parameterize the joint distribution of these observations using the chain rule for conditional probabilities:

$$P(Y) = P(\boldsymbol{y}^{(1)},..\boldsymbol{y}^{(\tau)}) = \prod_{t=1}^{\tau} P(\boldsymbol{y}^{(t)} \mid \boldsymbol{y}^{(t-1)}, \boldsymbol{y}^{(t-2)}, ..., \boldsymbol{y}^{(1)})$$

where the right hand side of the bar | is empty for $t=1$

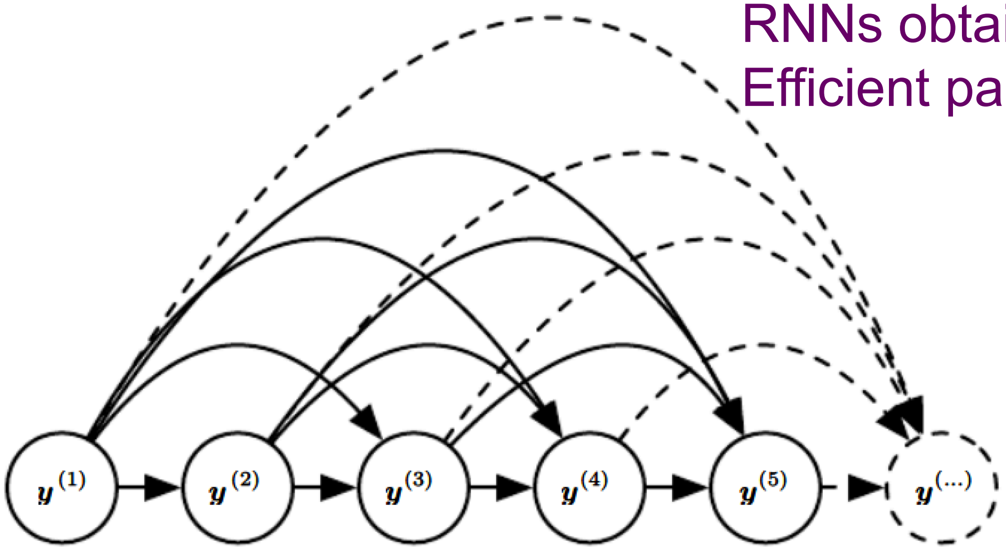
- Hence the negative log-likelihood of a set of values $\{\boldsymbol{y}^{(1)},..\boldsymbol{y}^{(\tau)}\}$ according to such a model is

$$L = \sum_t L^{(t)}$$

$$\text{where } L^{(t)} = -\sum_t \log P\left(\mathrm{y}^{(t)} = \boldsymbol{y}^{(t)} \mid y^{(t-1)}, y^{(t-2)}, ..., y^{(1)}\right)$$

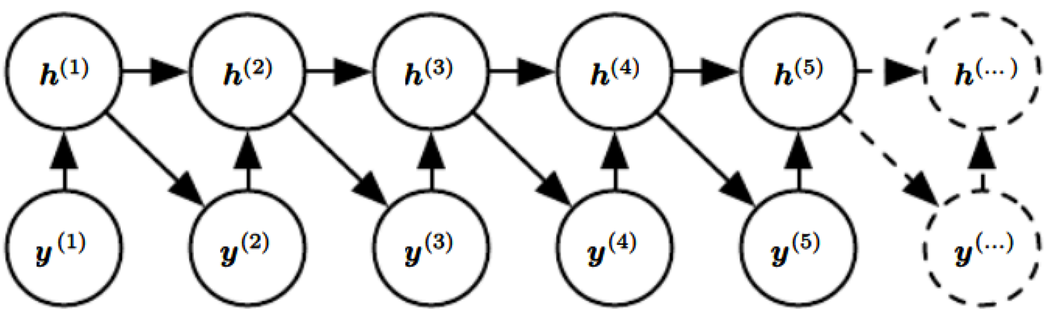
Fully connected graph model vs RNN

Every one of the past observations $\boldsymbol{y}^{(i)}$ may influence the conditional distribution of some $\boldsymbol{y}^{(t)}$. Parameterizing this way is inefficient. RNNs obtain the same connectivity but Efficient parameterization as shown below



Introducing the state variable in the PGM of RNN, even though it is a deterministic function of its inputs, helps see we get efficient parameterization

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



Every stage in the sequence for $\boldsymbol{h}^{(t)}$ and $\boldsymbol{y}^{(t)}$ involves the same structure and can share the parameters with other stages