Recursive Neural Networks

Sargur Srihari
srihari@buffalo.edu

Topics

- Sequence Modeling: Recurrent and Recursive Nets

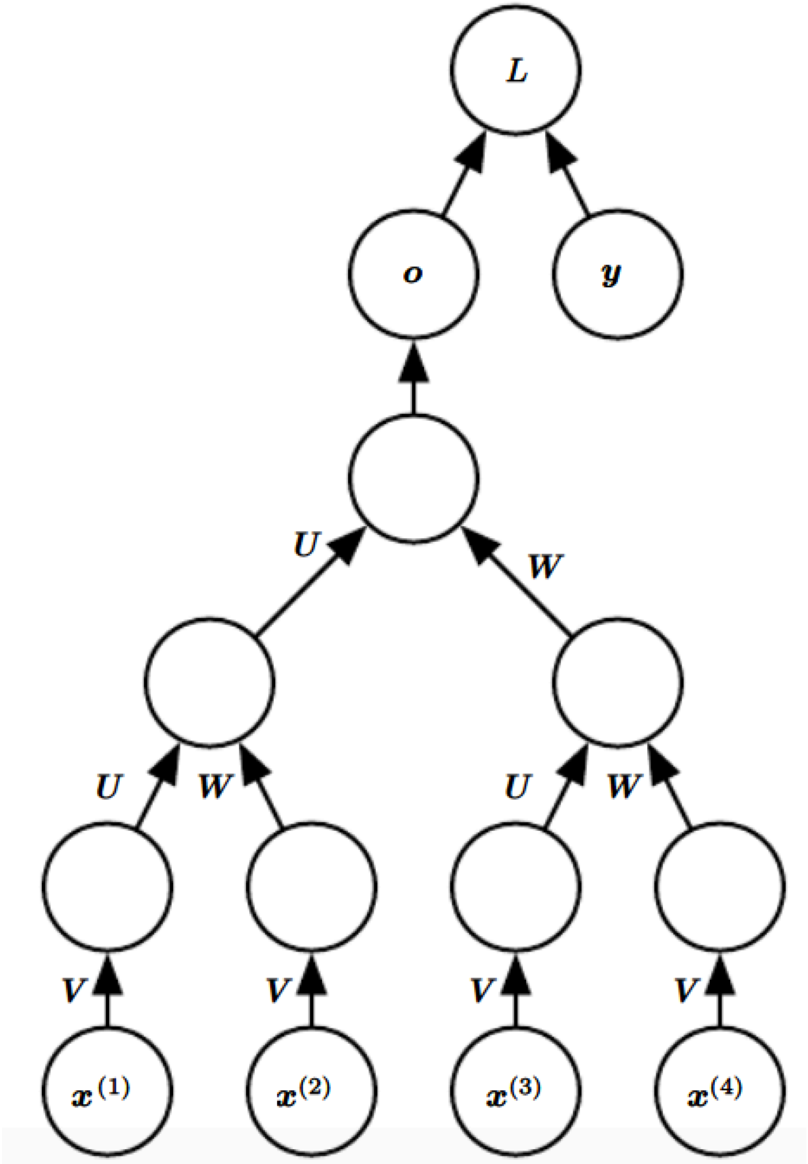
1. Unfolding Computational Graphs
2. Recurrent Neural Networks
3. Bidirectional RNNs
4. Encoder-Decoder Sequence-to-Sequence Architectures
5. Deep Recurrent Networks
6. Recursive Neural Networks
7. The Challenge of Long-Term Dependencies
8. Echo-State Networks
9. Leaky Units and Other Strategies for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

Recursive Neural Networks

- They are yet another generalization of recurrent networks with a different kind of computational graph
- It is structured as a deep tree, rather than the chain structure of RNNs
- The typical computational graph for a recursive network is shown next

Computational graph of a Recursive Network

- It generalizes a recurrent network from a chain to a tree
- A variable sequence $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, , \boldsymbol{x}^{(t)}$ can be mapped to a fixed size representation (the output $\boldsymbol{o}$), with a fixed set of parameters (the weight matrices U, V, W)
- Figure illustrates supervised learning case in which target $\boldsymbol{y}$ is provided that is associated with the whole sequence



Advantage of Recursive over Recurrent Nets

- For a sequence of the same length τ, the depth (measured as the no. of compositions of nonlinear operations) can be reduced from τ to $O(\log \boldsymbol{\tau})$, which might help deal with long-term dependencies
- An open question is how best to structure the tree

Need for Recursive nets in NLP

- Deep learning based methods learn low-dimensional, real-valued vectors for word tokens, mostly from a large data corpus, successfully capturing syntactic and semantic aspects of text
- For tasks where the inputs are larger text units, e.g., phrases, sentences or documents, a compositional model is first needed to aggregate tokens into a vector with fixed dimensionality that can be used for other NLP tasks
- Models for achieving this fall into two categories: recurrent models and recursive models

Recurrent Model for NLP

- Recurrent models deal successfully with time series data
- The recurrent models generally consider no linguistic structure aside from the word order
- They were applied early on to NLP by modeling a sentence as tokens processed sequentially and at each step combining the current token with previously built embeddings
- Recurrent models can be extended to bidirectional ones from both left to right and right to left
- These models consider no linguistic structure aside from word order

Recursive Models for NLP

- Recursive neural models (also referred to as tree models) by contrast are structured by syntactic parse trees
- Instead of considering tokens sequentially, recursive models combine neighbors based on the recursive structure of parse trees, starting from the leaves and proceeding recursively in a bottom-up fashion until the root of the parse tree is reached
  - Ex: for the phrase the food is delicious, following the operation sequence ((the food) (is delicious)) rather than the sequential order (((the food) is) delicious)

Advantage of Recursive Model for NLP

- They have the potential of capturing long-distance dependencies
- Two tokens may be structurally closer to each other even though they are far away in word sequence
- Ex: a verb and its corresponding direct object can be far away in terms of tokens if many adjectives lie inbetween, but they are adjacent in the parse tree
- However parsing is slow and domain dependent
- See performance comparison with LSTM on four NLP tasks at https://nlp.stanford.edu/pubemnlp2015_2_jiwei.pdf

Structure of the Tree

- One option is to have a tree structure that does not depend on the data, such as a balanced binary tree
- In some application domains, external methods can suggest the appropriate tree structure
  - Ex: when processing natural language sentences, the tree structure for the recursive network can be fixed to the structure of the parse tree of the sentence provided by a natural language parse
- Ideally, one would like the learner itself to discover and infer the tree structure that is appropriate for any given input

Variants of Recursive Net idea

- Associate data with a tree structure and associate inputs and targets with individual nodes of the tree
  - The computation performed for each node does not have to be the artificial neuron computation (affine transformation of all inputs followed by a monotone nonlinearity)
  - Can use a tensor operations of bilinear forms
    - Previously found useful to model linear relationships between concepts when the concepts are represented by continuous vectors