Cross Validated

Recurrent vs Recursive Neural Networks: Which is better for NLP?

Ask Question

37

24

There are Recurrent Neural Networks and Recursive Neural Networks. Both are usually denoted by the same acronym: RNN. According to Wikipedia, Recurrent NN are in fact Recursive NN, but I don't really understand the explanation.

Moreover, I don't seem to find which is better (with examples or so) for Natural Language Processing. The fact is that, although Socher uses Recursive NN for NLP in his tutorial, I can't find a good implementation of recursive neural networks, and when I search in Google, most of the answers are about Recurrent NN.

Besides that, is there another DNN which applies better for NLP, or it depends on the NLP task? Deep Belief Nets or Stacked Autoencoders? (I don't seem to find any particular util for ConvNets in NLP, and most of the implementations are with machine vision in mind).

Finally, I would really prefer DNN implementations for C++ (better yet if it has GPU support) or Scala (better if it has Spark support) rather than Python or Matlab/Octave.

I've tried Deeplearning4j, but it's under constant development and the documentation is a little outdated and I can't seem to make it work. Too bad because it has the "black box" like way of doing things, very much like scikit-learn or Weka, which is what I really want.

machine-learning neural-networks deep-learning natural-language

edited Oct 22 '16 at 0:18
gung ♦
106k 34 258 520

asked May 22 '15 at 17:50
crscardellino
358 1 4 10

2 Answers

34

✓

Recurrent Neural networks are recurring over time. For example if you have a sequence

x = ['h', 'e', 'l', 'l']

This sequence is fed to a **single** neuron which has a single connection to itself.

At time step 0, the letter 'h' is given as input.At time step 1, 'e' is given as input. The network when unfolded over time will look like this

Home

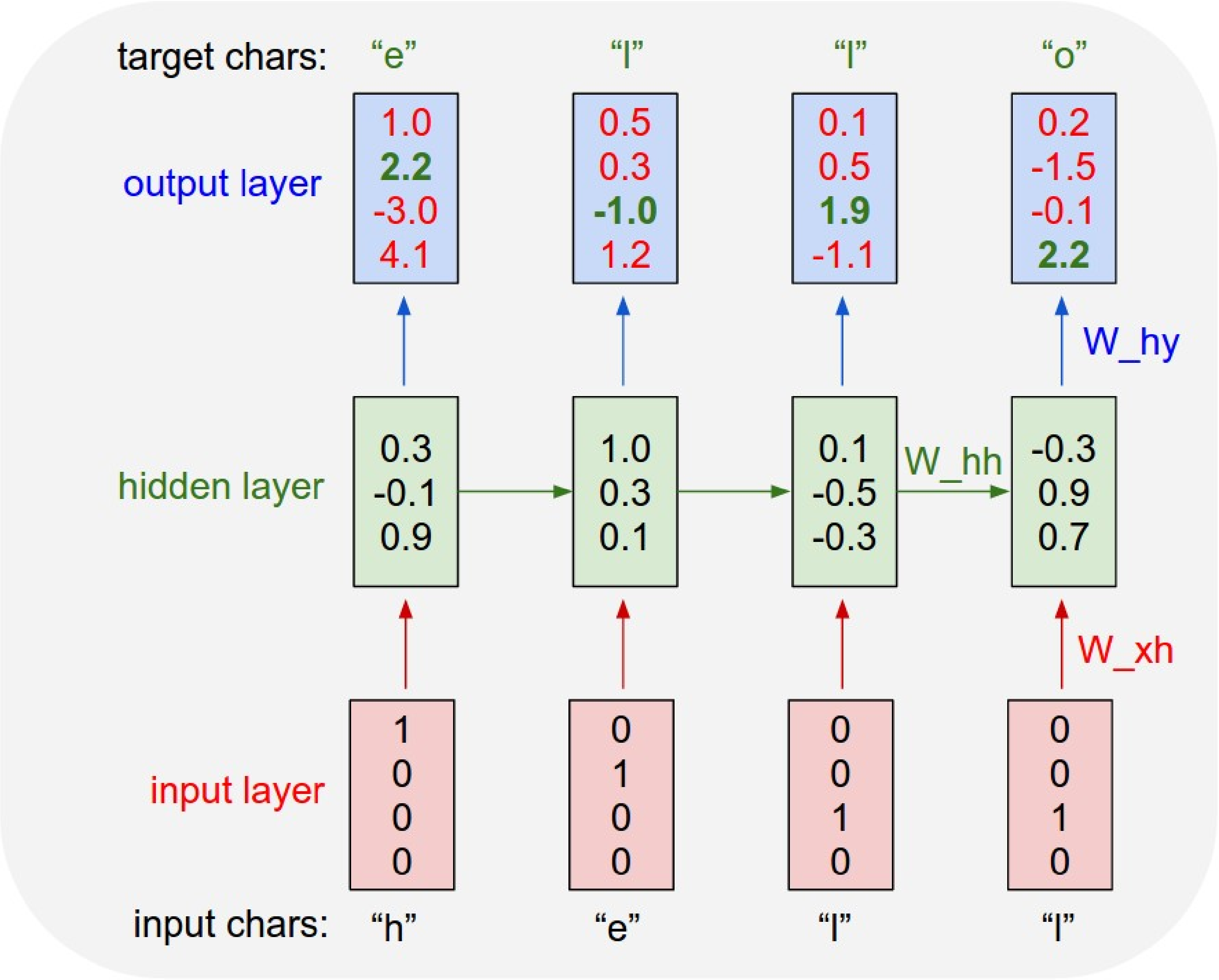
Questions

Tags

Users

Unanswered

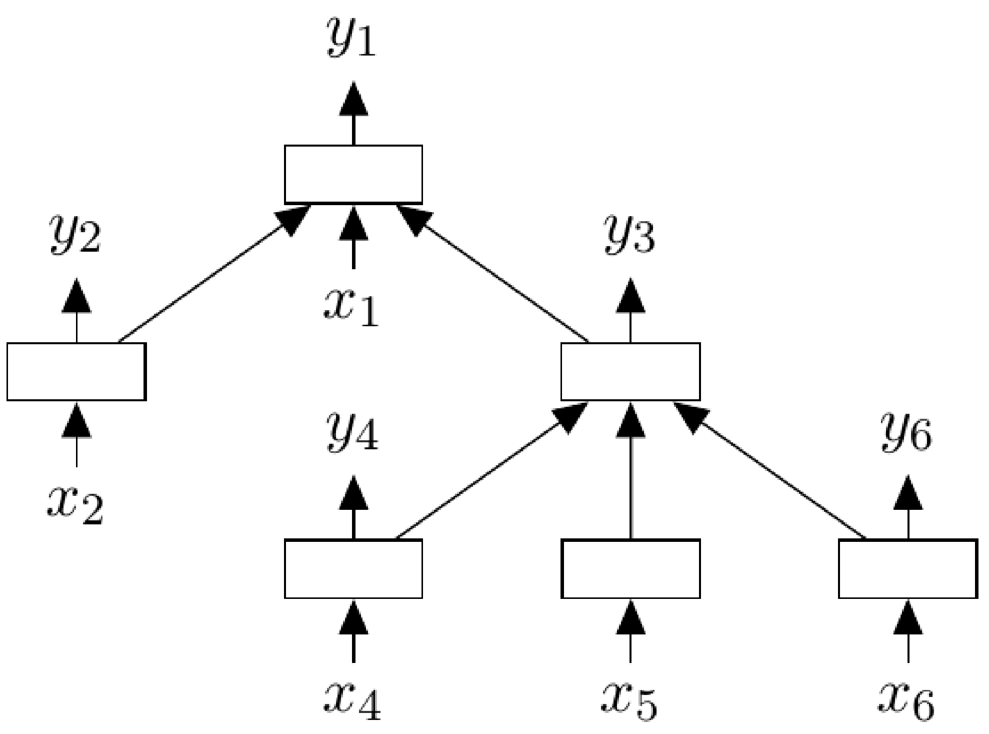
unfolded over time will look like this:



A recursive network is just a generalization of a recurrent network. In a recurrent network the weights are shared (and dimensionality remains constant) along the length of the sequence because how would you deal with position-dependent weights when you encounter a sequence at test-time of different length to any you saw at train-time. In a recursive network the weights are shared (and dimensionality remains constant) at every node for the same reason.

This means that all the W_xh weights will be equal(shared) and so will be the W_hh weight. This is simply because it is a single neuron which has been unfolded in time.

This is what a **Recursive Neural Network** looks like.



It is quite simple to see why it is called a Recursive Neural Network. Each parent node's children are simply a node similar to that node.

The Neural network you want to use depends on your usage. In Karpathy's blog, he is generating characters one at a time so a recurrent neural network is good.

But if you want to generate a parse tree, then using a Recursive Neural Network is better because it helps to create better hierarchical representations.

If you want to do deep learning in c++, then use CUDA. It has a nice user-base, and is fast. I do not know more about that so cannot comment more.

In python, Theano is the best option because it provides automatic differentiation, which means that when you are forming big, awkward NNs, you don't have to find gradients by hand. Theano does it automatically for you. This feature is lacked by Torch7.

Theano is very fast as it provides C wrappers to python code and can be implemented on GPUs. It also has an awesome user base, which is very important while learning something new.

answered Jun 27 '15 at 23:15

Azrael
596 3 7

11

Large Recurrent Neural Networks are considered maybe the most powerful model for NLP. A great article written by A. Karpathy on Recurrent Neural Networks and character level modeling is available at http://karpathy.github.io/2015/05/21/rnn-effectiveness/

at http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Having tried a large number of libraries for deep learning (theano, caffe etc.). I would strongly suggest the use Torch7 which is considered the state-of-the-art tool for NNs and it supported by NYU, Facebook AI and Google DeepMind. Torch7 is based on lua and there are so many examples that you can easily familiarize with. A lot of code can be found on github, a good start would be https://github.com/wojzaremba/lstm.

Finally, the beauty of lua is that LuaJIT can be injected very easily in Java, Python, Matlab etc.

answered May 23 '15 at 20:08
Yannis Assael
2,080 12 22

2 Suggest reading Karpathy's blog. Really heapful in understanding RNN. – SolessChong Jul 25 '15 at 5:34