P4: Feedforward Neural Networks

Spring 2021
Annoucements

- Feedback form: https://forms.gle/Bsgng7m21rXxWsTw5
- Reading materials
- Perusall use
- Pre-lecture questions
- Class structure (lecture + discussion)

- 584 readings will NOT be tested in midterm.
Deep Unordered Composition Rivals Syntactic Methods for Text Classification

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, Hal Daumé III

1University of Maryland, Department of Computer Science and UMIACS
2University of Colorado, Department of Computer Science

{miyyer, varunm, hal}@umiacs.umd.edu, Jordan.Boyd.Graber@colorado.edu
Key takeaways

- A very simple model called **Deep Averaging Networks (DAN)** achieves competent performance on **sentiment analysis** and **factoid question answering**.

DAN

\[
\begin{align*}
  h_2 &= f(W_2 \cdot h_1 + b_2) \\
  h_1 &= f(W_1 \cdot av + b_1) \\
  av &= \frac{1}{4} \sum_{i=1}^{4} c_i
\end{align*}
\]

How can we interpret these results?
A little bit of history

**EMNLP 2013**

Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank

Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang,
Christopher D. Manning, Andrew Y. Ng and Christopher Potts
Stanford University, Stanford, CA 94305, USA
richard@socher.org, {aperelyg, jcchuang, ang}@cs.stanford.edu
{jeaneis,manning,cpotts}@stanford.edu

**NIPS 2014**

Deep Recursive Neural Networks for Compositionality in Language

Ozan İrsoy
Department of Computer Science
Cornell University
Ithaca, NY 14853
oirsoy@cs.cornell.edu

Claire Cardie
Department of Computer Science
Cornell University
Ithaca, NY 14853
cardie@cs.cornell.edu
A little bit of history

Convolutional Neural Networks for Sentence Classification

Yoon Kim
New York University
yhk255@nyu.edu

Dropout: A Simple Way to Prevent Neural Networks from Overfitting

Nitish Srivastava
Geoffrey Hinton
Alex Krizhevsky
Ilya Sutskever
Ruslan Salakhutdinov
Department of Computer Science
University of Toronto
10 Kings College Road, Rm 3302
Toronto, Ontario, M5S 3G4, Canada.

NITISH@CS.TORONTO.EDU
HINTON@CS.TORONTO.EDU
KRIZ@CS.TORONTO.EDU
ILYA@CS.TORONTO.EDU
RSALAKH@CS.TORONTO.EDU

EMNLP 2014

JMLR 2014
Deep Averaging Networks (DAN)

The model doesn't model the word order and even n-gram information!

\[ av = \sum_{i=1}^{4} \frac{c_i}{4} \]

softmax: predict positive label

\[ z_2 = f(W_2 \cdot z_1) \]
\[ z_1 = f(W_1 \cdot av) \]
\[ av = \sum_{i=1}^{4} \frac{c_i}{4} \]

NBOB

DAN
Recursive Neural Networks (RNNs)

- This model relies on a parser to provide an input tree (error-prone)
- $W$ is shared in all the composition functions
Word dropout

• Drop entire word embeddings and take the average of remaining words

\[ r_w \sim \text{Bernoulli}(p) \]
\[ \hat{X} = \{ w | w \in X \text{ and } r_w > 0 \} \]
\[ z = g(w \in X) = \frac{\sum_{w \in \hat{X}} v_w}{|\hat{X}|} \]
Tasks & Datasets

Sentiment analysis

- RT: Rotten Tomatoes
  - 2-class, sentence-level classification
- Stanford sentiment treebank
  - 2-class or 5-class - ++, +, 0, -, —
  - sentence-level classification
- IMDB
  - 2-class, document-level classification

The phrase-level labels are only used for training!
Especially crucial for tree-based models

See more examples at https://nlp.stanford.edu/sentiment/treebank.html
Quizbowl QA task
- Input: 4-6 sentences describing an entity (authors, battles, or events)
- Output: entity

• 3,761 questions
• Augmented with 53,234 sentence/page-title pairs from Wikipedia

Example
• This creature has female counterparts named Penny and Gown.
• This creature appears dressed in Viking armor and carrying an ax when he is used as the mascot of PaX, a least privilege protection patch.
• This creature’s counterparts include Daemon on the Berkeley Software Distribution, or BSD.
• For ten points, name this mascot of the Linux operating system, a penguin whose name refers to formal male attire.
Experiments: sentiment analysis

<table>
<thead>
<tr>
<th>Model</th>
<th>RT</th>
<th>SST fine</th>
<th>SST bin</th>
<th>IMDB</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAN-ROOT</td>
<td></td>
<td>46.9</td>
<td>85.7</td>
<td>—</td>
<td>31</td>
</tr>
<tr>
<td>DAN-RAND</td>
<td>77.3</td>
<td>45.4</td>
<td>83.2</td>
<td>88.8</td>
<td>136</td>
</tr>
<tr>
<td>DAN</td>
<td>80.3</td>
<td>47.7</td>
<td>86.3</td>
<td>89.4</td>
<td>136</td>
</tr>
<tr>
<td>NBOW-RAND</td>
<td>76.2</td>
<td>42.3</td>
<td>81.4</td>
<td>88.9</td>
<td>91</td>
</tr>
<tr>
<td>NBOW</td>
<td>79.0</td>
<td>43.6</td>
<td>83.6</td>
<td>89.0</td>
<td>91</td>
</tr>
<tr>
<td>BiNB</td>
<td>—</td>
<td>41.9</td>
<td>83.1</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>NBSVM-bi</td>
<td>79.4</td>
<td>—</td>
<td>—</td>
<td>91.2</td>
<td></td>
</tr>
<tr>
<td>RecNN*</td>
<td>77.7</td>
<td>43.2</td>
<td>82.4</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>RecNTN*</td>
<td>—</td>
<td>45.7</td>
<td>85.4</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>DRecNN</td>
<td>—</td>
<td>49.8</td>
<td>86.6</td>
<td>—</td>
<td>431</td>
</tr>
<tr>
<td>TreeLSTM</td>
<td>—</td>
<td>50.6</td>
<td>86.9</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>DCNN*</td>
<td>—</td>
<td>48.5</td>
<td>86.9</td>
<td>89.4</td>
<td>—</td>
</tr>
<tr>
<td>PVEC*</td>
<td>—</td>
<td>48.7</td>
<td>87.8</td>
<td>92.6</td>
<td>—</td>
</tr>
<tr>
<td>CNN-MC</td>
<td>81.1</td>
<td>47.4</td>
<td><strong>88.1</strong></td>
<td>—</td>
<td>2,452</td>
</tr>
<tr>
<td>WRRBM*</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>89.2</td>
<td>—</td>
</tr>
</tbody>
</table>

- Initialization with GloVe embeddings helps
- Phrase-level labels helps
- It seems to work better on sentence-level tasks than on document-level tasks
- DAN is fast and competitive on sentiment analysis
Experiments: quizbowl QA

<table>
<thead>
<tr>
<th>Model</th>
<th>Pos 1</th>
<th>Pos 2</th>
<th>Full</th>
<th>Time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoW-DT</td>
<td>35.4</td>
<td>57.7</td>
<td>60.2</td>
<td>—</td>
</tr>
<tr>
<td>IR</td>
<td>37.5</td>
<td>65.9</td>
<td>71.4</td>
<td>N/A</td>
</tr>
<tr>
<td>QANTA</td>
<td>47.1</td>
<td>72.1</td>
<td>73.7</td>
<td>314</td>
</tr>
<tr>
<td>DAN</td>
<td>46.4</td>
<td>70.8</td>
<td>71.8</td>
<td>18</td>
</tr>
<tr>
<td>IR-WIKI</td>
<td>53.7</td>
<td>76.6</td>
<td>77.5</td>
<td>N/A</td>
</tr>
<tr>
<td>QANTA-WIKI</td>
<td>46.5</td>
<td>72.8</td>
<td>73.9</td>
<td>1,648</td>
</tr>
<tr>
<td>DAN-WIKI</td>
<td><strong>54.8</strong></td>
<td>75.5</td>
<td>77.1</td>
<td>119</td>
</tr>
</tbody>
</table>

QANTA: recursive neural networks based on dependency tree

- The gap between DAN and QANTA increases when # of sentences increases
- DAN improves with noisy data
How do DANs work?

the film’s performances were awesome
DANs can’t handle negations well but tree-based models can’t either

- We collect 48 positive and 44 negative sentences from the SST that each contain at least one negation and one contrastive conjunction.

- When confronted with a negation, both the unordered **DAN** and syntactic **DRecNN** predict negative sentiment around 70% of the time.

- Accuracy on only the positive sentences in our subset is low: 37.5% for the **DAN** and 41.7% for the **DRecNN**

<table>
<thead>
<tr>
<th>Sentence</th>
<th>DAN</th>
<th>DRecNN</th>
<th>Ground Truth</th>
</tr>
</thead>
<tbody>
<tr>
<td>a <em>lousy</em> movie that’s <em>not merely unwatchable, but also unlistenable</em></td>
<td>negative</td>
<td>negative</td>
<td>negative</td>
</tr>
<tr>
<td>if you’re <em>not a prepubescent girl, you’ll be laughing at britney spears’</em> movie-starring debut whenever it doesn’t have you impatiently <em>squinting at your watch</em></td>
<td>negative</td>
<td>negative</td>
<td>negative</td>
</tr>
<tr>
<td><em>blessed with immense physical prowess he may well be, but sb is simply not an actor</em></td>
<td>positive</td>
<td>neutral</td>
<td>negative</td>
</tr>
<tr>
<td>who <em>knows</em> what exactly godard is on about in this film, but his words and images do <em>not</em> have to <em>add up to mesmerize you.</em></td>
<td>positive</td>
<td>positive</td>
<td>positive</td>
</tr>
<tr>
<td>it’s so <em>good</em> that its relentless, polished wit can withstand not <em>only inagen school productions, but even oliver parker’s</em> movie adaptation</td>
<td>negative</td>
<td>positive</td>
<td>positive</td>
</tr>
<tr>
<td><em>too bad, but thanks to some lovely comedic moments and several fine performances, it’s not a total loss</em></td>
<td>negative</td>
<td>negative</td>
<td>positive</td>
</tr>
<tr>
<td>this movie was <em>not good</em></td>
<td>negative</td>
<td>negative</td>
<td>negative</td>
</tr>
<tr>
<td>this movie was <em>good</em></td>
<td>positive</td>
<td>positive</td>
<td>positive</td>
</tr>
<tr>
<td>this movie was <em>bad</em></td>
<td>negative</td>
<td>negative</td>
<td>negative</td>
</tr>
<tr>
<td>the movie was <em>not bad</em></td>
<td>negative</td>
<td>negative</td>
<td>positive</td>
</tr>
</tbody>
</table>
Discussion

- DANs are fast and competitive on sentiment analysis and quizbowl QA tasks. Do you think these results generalize? What about other tasks?

- What are the limitations of DANs? How can we improve them?

- Do word order and compositionally matter?

- What does word dropout do?

- What do the non-linear layers do in DANs?