Rdoc2vec

Jake Clark, Austin Cooke, Steven Rolph, Stephen Sherrard
CS4624: Multimedia, Hypertext, and Information Access
Final Presentation
Dr. Edward A. Fox
Virginia Tech Blacksburg VA 24061
Surge 109     5/2/2017
Parsing and Neural Network

- **Added Functionality**
  - Allows for custom stop words
  - Can trim vocabulary for words below a certain frequency

- **Reduces the Size of the Problem Space**
  - Less weights to keep track off
    - Reduces the size of the neural network [1]
  - Less weights to train
    - Reduces the time spent training [1]
Building our own Neural Network

- The basic structure of a neural network
- Hidden = Weight1 * Input1 + Weight2 * Input2 + ...
Building our own Neural Network

- These operations can be imagined as matrix operations
- “One-hot” vector
Building our own Neural Network

- Back Propagation is the training phase
- Incomplete and untested as of yet
### Sample Documents

#### Table 2 -- Initial sample document sets collected

<table>
<thead>
<tr>
<th>Documents</th>
<th>Number of Documents</th>
<th>Size Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>US Congressional Districts</td>
<td>442</td>
<td>3-40kb</td>
</tr>
<tr>
<td>4-Octave Singers</td>
<td>42</td>
<td>3-140kb</td>
</tr>
<tr>
<td>Universities</td>
<td>24</td>
<td>2-35kb</td>
</tr>
</tbody>
</table>

#### Table 3 -- Revised sample document sets

<table>
<thead>
<tr>
<th>Documents</th>
<th>Quantity</th>
<th>Size Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virginia</td>
<td>6</td>
<td>9-19kb</td>
</tr>
<tr>
<td>Virginia2</td>
<td>6</td>
<td>1-2kb</td>
</tr>
<tr>
<td>Florida</td>
<td>27</td>
<td>1-2kb</td>
</tr>
<tr>
<td>Singer/VA</td>
<td>8</td>
<td>1-3kb</td>
</tr>
</tbody>
</table>
Testing Results (Virginia)
Testing Results (Virginia2)
Saving Results

- Results saved to .csv file
  - Can save new file or append to file and return full data set
- Built using `readr` library for increased performance
  - `readr` write_csv is twice as fast as R base `write.csv`[5]

Future Work:

- Extend to handle other file types
Plotting Results

● t-SNE Algorithm & Reducing Dimensions
  ○ Allows visualization of high dimensional data in 2D and 3D
  ○ ‘Rtsne’ package uses the Barnes-Hut-SNE algorithm
  ○ Barnes-Hut: $O(n \log n)$ [6]
  ○ Baseline t-SNE: $O(n^2)$ [6]

● Future Work:
  ○ Implement using ‘rtsne’ R package
t-SNE Example

2,000 most common English words (300 dimensions to 2) [6]
Lessons Learned

● Timeline / Schedule
  ○ Attempting DBOW and Distributed Memory

● Research
  ○ Finding a balance between research and decision making

● Better defining goals
  ○ Be more realistic about scope
Demo

Running the script
1. First we build a shared vocabulary
2. Then we create a document vector for each document
3. Finally we have a list containing cosine similarity between two documents.
References


Acknowledgements

Dr. Edward Fox
CS 4624 Professor
fox@vt.edu

Don Sanderson
Service Manager, Marketing Solutions
Eastman Chemical Company
donsanderson@eastman.com

Adam Spannbauer
R Programmer and Data Scientist
Eastman Chemical Company
adamspannbauer@eastman.com