Bag of Tricks for Efficient Text Classification

Armand Joulin, Edouard Grave, Piotr Bojanowski, Tomas Mikolov
Facebook AI Research
Text classification
Natural language processing is fun.

Bag of Words (or n-grams)

Average low-dimensional!
Simple linear model

Text vector

Word vectors

output

hidden

x1

x2

... 

xn-1

xn

Softmax
Learning

\[-\frac{1}{N} \sum_{n=1}^{N} y_n \log(f(BA x_n))\]

- # documents
- label of n-th doc
- normalized bag of features of n-th doc
- weight matrices
- softmax
Hierarchical softmax

Probability of a node is always lower than one of its parent:

\[ P(n_{l+1}) = \prod_{i=1}^{l} P(n_i). \]
Hierarchical softmax

Probability of a node is always lower than one of its parent

$P(n_{l+1}) = \prod_{i=1}^{l} P(n_i)$. 

$O(h\log(k))$ vs $O(kh)$ training time!
## Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Yahoo</th>
<th>Accuracy</th>
<th>Time</th>
<th>Amazon full</th>
<th>Accuracy</th>
<th>Time</th>
<th>Amazon polarity</th>
<th>Accuracy</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>char-CNN</td>
<td></td>
<td>71.2</td>
<td>1 day</td>
<td>59.5</td>
<td></td>
<td>5 days</td>
<td></td>
<td>94.5</td>
<td>5 days</td>
</tr>
<tr>
<td>VDCNN</td>
<td></td>
<td>73.4</td>
<td>2h</td>
<td>63</td>
<td></td>
<td>7h</td>
<td></td>
<td>95.7</td>
<td>7h</td>
</tr>
<tr>
<td>fastText</td>
<td></td>
<td>72.3</td>
<td>5s</td>
<td>60.2</td>
<td></td>
<td>9s</td>
<td></td>
<td>94.6</td>
<td>10s</td>
</tr>
</tbody>
</table>

As good as NN!

Fast!
Summary

- fastText is often on par with deep learning classifiers
- fastText takes seconds, instead of days
- Can learn vector representations of words in different languages (with performance better than word2vec!)

Thanks!