# Bag of Tricks for Efficient Text Classification

Armand Joulin, Edouard Grava, Piotr Bojanowski and Tomas Mikolov

Presented by: Ziyad Alsaeed

Jan 25, 2019

# Motivation

- Text classification has many different important applications (e.g. web search, information retrieval and ranking).

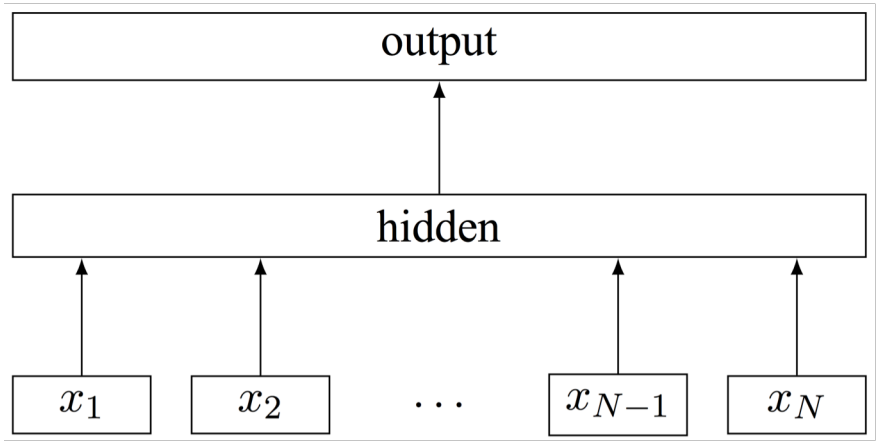
- Although neural network based models are increasingly popular and achieve very good performance (accuracy), they are not fast for light operations.

- Linear classifiers (e.g. logistic regression and SVM) often obtain similar performance as neural network if the right features are used. Moreover, linear classifier can scale to very large data sets.

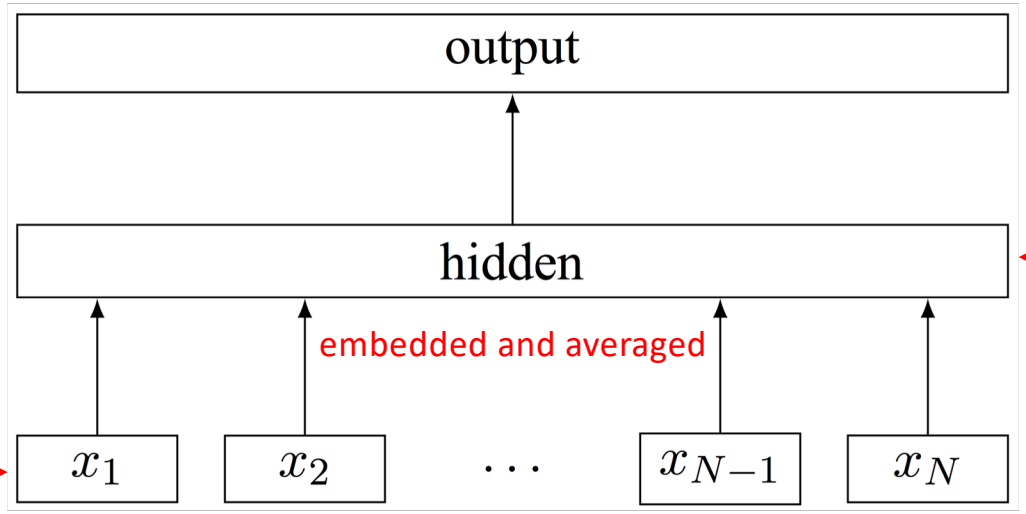
# Model Architecture

- Linear classifiers are possibly limited in generalization because they don't share parameters among features and classes.

- Such issue is commonly solved by factorizing the linear classifier into low rank matrices.



# Model Architecture (continue)



# Learning

- Use the softmax function to compute the probability distribution over predefined classes.
  - $N$: Number of documents.
  - $x_n$: normalized bag of features of the n-th document.
  - $y_n$: the label of the n-th document.
  - $A$ and $B$: are the weight matrices.

$$-\frac{1}{N} = \sum_{n=1}^{N} y_n \log(f(BAx_n))$$

# Hierarchical softmax

- The complexity if the linear classifier is $O(kh)$.
  - $k$: is the number of classes.
  - $h$: the dimension of the text representation.

- Use hierarchical softmax based on Huffman coding tree to improve the running time.

- Computational complexity during training becomes $O(h\ \log_2(k))$.

- This result on a tree where the probability of a node is always lower than the one of its parent, which is also advantages during test time.

# BoW vs. Bag of N-gram

- Bag of Words with order taken into account, is computationally expensive.

- Using N-gram capture at least partial information about the local word order. And performs well in practice.

- Use hashing trick to maintain fast and memory efficient mapping of the n-grams (10M bins for bigrams and 100M otherwise).

# Major Tricks

1. Introduce hidden variables to have a more general model.

2. Use Hierarchal softmax for efficiency.

3. Use n-gram features to maintain partial order and increase performance.

# Experiments

- Use sentiment analysis to compare fastText's performance compared to existing text classifier.

- Conduct a tag prediction experiment to evaluate fastText's scalability.

# Sentiment Analysis

- Use the same data set as from Zhang et al. (2015)
- Compare to six of the state of the art text classification techniques.
- Trained fastText using 10 hidden units with and without a bigram.

# Sentiment Analysis

| Model | AG | Sogou | DBP | Yelp P. | Yelp F. | Yah. A. | Amz. F. | Amz. P. |
|---|---|---|---|---|---|---|---|---|
| BoW (Zhang et al., 2015) | 88.8 | 92.9 | 96.6 | 92.2 | 58.0 | 68.9 | 54.6 | 90.4 |
| ngrams (Zhang et al., 2015) | 92.0 | 97.1 | 98.6 | 95.6 | 56.3 | 68.5 | 54.3 | 92.0 |
| ngrams TFIDF (Zhang et al., 2015) | 92.4 | 97.2 | 98.7 | 95.4 | 54.8 | 68.5 | 52.4 | 91.5 |
| char-CNN (Zhang and LeCun, 2015) | 87.2 | 95.1 | 98.3 | 94.7 | 62.0 | 71.2 | 59.5 | 94.5 |
| char-CRNN (Xiao and Cho, 2016) | 91.4 | 95.2 | 98.6 | 94.5 | 61.8 | 71.7 | 59.2 | 94.1 |
| VDCNN (Conneau et al., 2016) | 91.3 | 96.8 | 98.7 | 95.7 | 64.7 | 73.4 | 63.0 | 95.7 |
| `fastText`, $h = 10$ | 91.5 | 93.9 | 98.1 | 93.8 | 60.4 | 72.0 | 55.8 | 91.2 |
| `fastText`, $h = 10$, bigram | 92.5 | 96.8 | 98.6 | 95.7 | 63.9 | 72.3 | 60.2 | 94.6 |

# Sentiment Analysis

| Model | AG | Sogou | DBP | Yelp P. | Yelp F. | Yah. A. | Amz. F. | Amz. P. |
|---|---|---|---|---|---|---|---|---|
| BoW (Zhang et al., 2015) | 88.8 | 92.9 | 96.6 | 92.2 | 58.0 | 68.9 | 54.6 | 90.4 |
| ngrams (Zhang et al., 2015) | 92.0 | 97.1 | 98.6 | 95.6 | 56.3 | 68.5 | 54.3 | 92.0 |
| ngrams TFIDF (Zhang et al., 2015) | 92.4 | 97.2 | 98.7 | 95.4 | 54.8 | 68.5 | 52.4 | 91.5 |
| char-CNN (Zhang and LeCun, 2015) | 87.2 | 95.1 | 98.3 | 94.7 | 62.0 | 71.2 | 59.5 | 94.5 |
| char-CRNN (Xiao and Cho, 2016) | 91.4 | 95.2 | 98.6 | 94.5 | 61.8 | 71.7 | 59.2 | 94.1 |
| VDCNN (Conneau et al., 2016) | 91.3 | 96.8 | 98.7 | 95.7 | 64.7 | 73.4 | 63.0 | 95.7 |
| `fastText`, $h = 10$ | 91.5 | 93.9 | 98.1 | 93.8 | 60.4 | 72.0 | 55.8 | 91.2 |
| `fastText`, $h = 10$, bigram | 92.5 | 96.8 | 98.6 | 95.7 | 63.9 | 72.3 | 60.2 | 94.6 |

# Sentiment Analysis

| Model | AG | Sogou | DBP | Yelp P. | Yelp F. | Yah. A. | Amz. F. | Amz. P. |
|---|---|---|---|---|---|---|---|---|
| BoW (Zhang et al., 2015) | 88.8 | 92.9 | 96.6 | 92.2 | 58.0 | 68.9 | 54.6 | 90.4 |
| ngrams (Zhang et al., 2015) | 92.0 | 97.1 | 98.6 | 95.6 | 56.3 | 68.5 | 54.3 | 92.0 |
| ngrams TFIDF (Zhang et al., 2015) | 92.4 | 97.2 | 98.7 | 95.4 | 54.8 | 68.5 | 52.4 | 91.5 |
| char-CNN (Zhang and LeCun, 2015) | 87.2 | 95.1 | 98.3 | 94.7 | 62.0 | 71.2 | 59.5 | 94.5 |
| char-CRNN (Xiao and Cho, 2016) | 91.4 | 95.2 | 98.6 | 94.5 | 61.8 | 71.7 | 59.2 | 94.1 |
| VDCNN (Conneau et al., 2016) | 91.3 | 96.8 | 98.7 | 95.7 | 64.7 | 73.4 | 63.0 | 95.7 |
| `fastText`, $h = 10$ | 91.5 | 93.9 | 98.1 | 93.8 | 60.4 | 72.0 | 55.8 | 91.2 |
| `fastText`, $h = 10$, bigram | 92.5 | 96.8 | 98.6 | 95.7 | 63.9 | 72.3 | 60.2 | 94.6 |

# Sentiment Analysis

How fastText would perform if we increase the n-gram or hidden unites?

| Model | AG | Sogou | DBP | Yelp P. | Yelp F. | Yah. A. | Amz. F. | Amz. P. |
|---|---|---|---|---|---|---|---|---|
| BoW (Zhang et al., 2015) | 88.8 | 92.9 | 96.6 | 92.2 | 58.0 | 68.9 | 54.6 | 90.4 |
| ngrams (Zhang et al., 2015) | 92.0 | 97.1 | 98.6 | 95.6 | 56.3 | 68.5 | 54.3 | 92.0 |
| ngrams TFIDF (Zhang et al., 2015) | 92.4 | 97.2 | 98.7 | 95.4 | 54.8 | 68.5 | 52.4 | 91.5 |
| char-CNN (Zhang and LeCun, 2015) | 87.2 | 95.1 | 98.3 | 94.7 | 62.0 | 71.2 | 59.5 | 94.5 |
| char-CRNN (Xiao and Cho, 2016) | 91.4 | 95.2 | 98.6 | 94.5 | 61.8 | 71.7 | 59.2 | 94.1 |
| VDCNN (Conneau et al., 2016) | 91.3 | 96.8 | 98.7 | 95.7 | 64.7 | 73.4 | 63.0 | 95.7 |
| `fastText`, $h = 10$ | 91.5 | 93.9 | 98.1 | 93.8 | 60.4 | 72.0 | 55.8 | 91.2 |
| `fastText`, $h = 10$, bigram | 92.5 | 96.8 | 98.6 | 95.7 | 63.9 | 72.3 | 60.2 | 94.6 |

# Sentiment Analysis – Other Methods

- Comparison to methods presented in Tang et al. (2015).
- Using a Bag of 5-grams leads to the best performance.

| Model | Yelp'13 | Yelp'14 | Yelp'15 | IMDB |
|---|---|---|---|---|
| SVM+TF | 59.8 | 61.8 | 62.4 | 40.5 |
| CNN | 59.7 | 61.0 | 61.5 | 37.5 |
| Conv-GRNN | 63.7 | 65.5 | 66.0 | 42.5 |
| LSTM-GRNN | 65.1 | 67.1 | 67.6 | 45.3 |
| `fastText` | 64.2 | 66.2 | 66.6 | 45.2 |

# Sentiment Analysis – Training Time

| | Zhang and LeCun (2015) | | Conneau et al. (2016) | | | fastText |
|---|---|---|---|---|---|---|
| | small char-CNN | big char-CNN | depth=9 | depth=17 | depth=29 | $h = 10$, bigram |
| AG | 1h | 3h | 24m | 37m | 51m | 1s |
| Sogou | - | - | 25m | 41m | 56m | 7s |
| DBpedia | 2h | 5h | 27m | 44m | 1h | 2s |
| Yelp P. | - | - | 28m | 43m | 1h09 | 3s |
| Yelp F. | - | - | 29m | 45m | 1h12 | 4s |
| Yah. A. | 8h | 1d | 1h | 1h33 | 2h | 5s |
| Amz. F. | 2d | 5d | 2h45 | 4h20 | 7h | 9s |
| Amz. P. | 2d | 5d | 2h45 | 4h25 | 7h | 10s |

- Table shows training time of a single epoch.
- Other methods are trained on a NVIDIA Tesla K40 GPU. fastText is trained on a CPU using 20 threads.

# Tag Prediction – Setup

- Use the YFCC100M dataset which consists of 100M images with captions, titles and tags.
- The goal is to use the captions and titles to predict the tags.
- Removed tags and words that occurred less than 100 times.
- Split the data as the following:
  - Train set of 91,188,648 examples.
  - Validation set of 930,497 examples.
  - Test set of 543,424 examples.

# Tag Prediction – Evaluation

| Model | prec@1 | Running time | |
|---|---|---|---|
| | | Train | Test |
| Freq. baseline | 2.2 | - | - |
| Tagspace, $h = 50$ | 30.1 | 3h8 | 6h |
| Tagspace, $h = 200$ | 35.6 | 5h32 | 15h |
| `fastText`, $h = 50$ | 31.2 | 6m40 | 48s |
| `fastText`, $h = 50$, bigram | 36.7 | 7m47 | 50s |
| `fastText`, $h = 200$ | 41.1 | 10m34 | 1m29 |
| `fastText`, $h = 200$, bigram | 46.1 | 13m38 | 1m37 |

# Tag Prediction – Examples

| Input | Prediction | Tags |
|---|---|---|
| taiyoucon 2011 digitals: individuals digital photos from the anime convention taiyoucon 2011 in mesa, arizona. if you know the model and/or the character, please comment. | #cosplay | #24mm #anime #animeconvention #arizona #canon #con #convention #cos **#cosplay** #costume #mesa #play #taiyou #taiyoucon |
| 2012 twin cities pride 2012 twin cities pride parade | #minneapolis | #2012twincitiesprideparade **#minneapolis** #mn #usa |
| beagle enjoys the snowfall | #snow | #2007 #beagle #hillsboro #january #maddison #maddy #oregon **#snow** |
| christmas | #christmas | #cameraphone #mobile |
| euclid avenue | #newyorkcity | #cleveland #euclidavenue |

# Comments