Bag of Tricks for Efficient Text Classification

Armand Joulin Edouard Grave Piotr Bojanowski Tomas Mikolov
Facebook AI Research

EACL '17

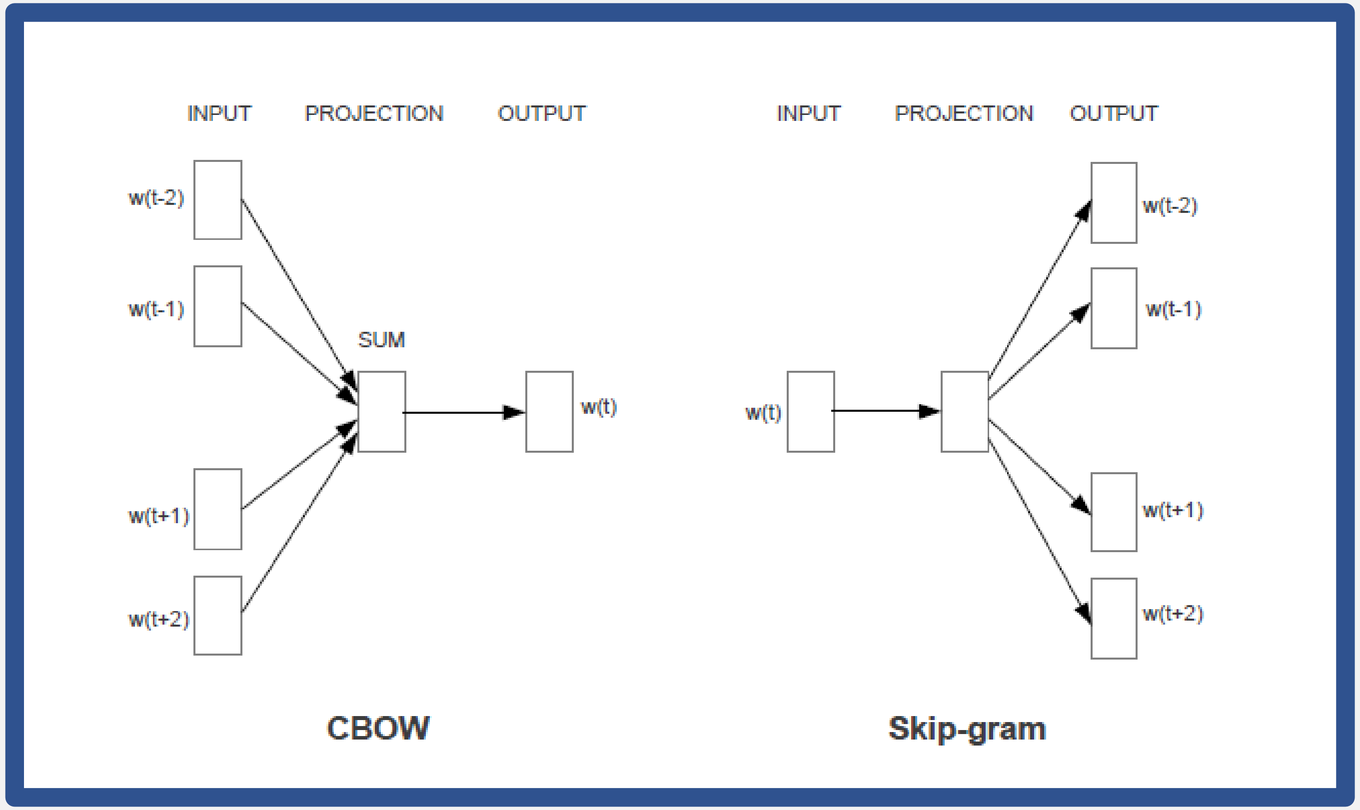
Noah Lee
noahlee357@korea.ac.kr
Data Intelligence Lab, Korea University
2020 12. 04

01 Problem Statement

How can we further efficiency on text classification?

02 Related Work

[Efficient Estimation of Word Representations in Vector Space] ‘13
Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean



02 Related Work

[Enriching Word Vectors with Subword Information] '16
Piotr Bojanowski, Edouard Grave, Armand Joulin and Tomas Mikolov

6.2 Character n-grams and morphemes

We want to qualitatively evaluate whether or not the most important n-grams in a word correspond to morphemes. To this end, we take a word vector that we construct as the sum of n-grams. As described in Sec. 3.2, each word w is represented as the sum of its n-grams: $u_w = \sum_{g \in \mathcal{G}_w} z_g$. For each n-gram g, we propose to compute the restricted representation $u_{w\backslash g}$ obtained by omitting g:

$$u_{w\backslash g} = \sum_{g' \in \mathcal{G} - \{g\}} z_{g'}.$$

- Word representation learning (morphology)
- Obtaining word vectors for out-of-vocabulary words

03 Motivation

Text Classification's application on

1) **Web Search**
2) **Information Retrieval**
3) **Ranking**
4) **Document Classification**

Neural Networks increasingly popular

(Kim, 2014; Zhang and LeCun, 2015; Conneau et al., 2016)

SLOW for large datasets

04

Contribution

fastText

1) Employs a simple yet efficient baseline for **text classification** for **large data sets.**

2) Utilizes **hierarchical softmax** & **n-gram features** to be on par with deep learning classifiers' accuracy yet orders faster.

3) Enables the training of more than **one billion** words in less than **ten minutes** (on standard multicore CPU).

05 Model : Baseline

Represent Sentences as Bag of Words(BoW)

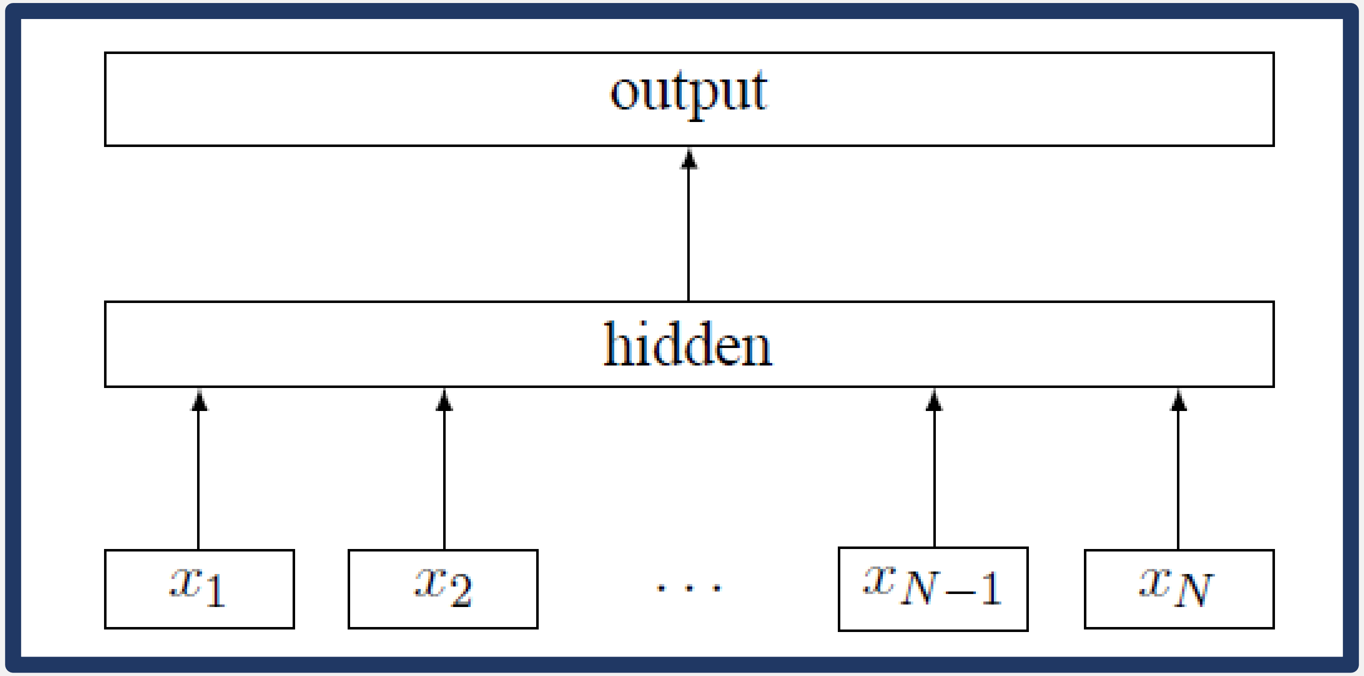
+

Train linear classifer (logistic regression / SVM)

Factorize linear classifier into low rank matrices

05 Model: Architecture

1) Input: Sentence with N **n-gram features**
2) **Look up** on first weight matrix
3) Word representations averaged into text representation (**Hidden variable**)
4) Fed to the **linear classifier**
5) Output: probability for **each class(or node)**



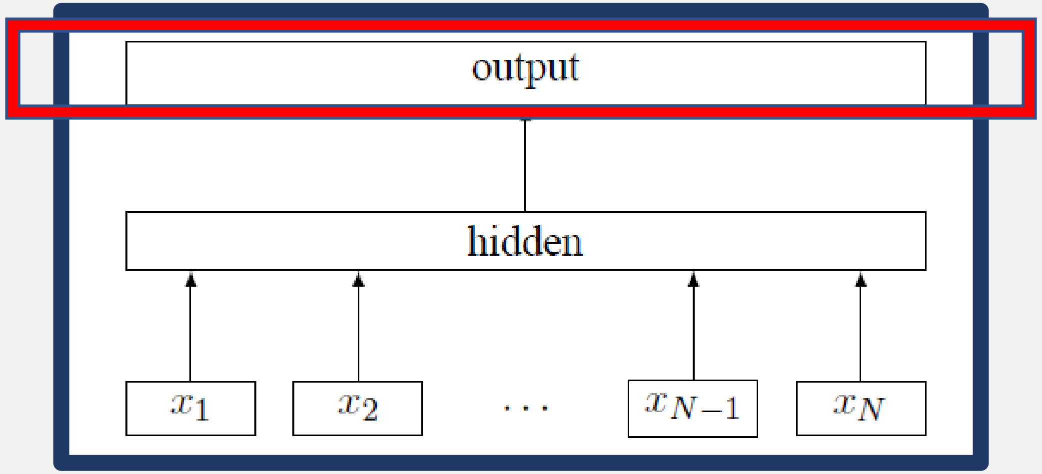
05 Model: Hierarchical Softmax*

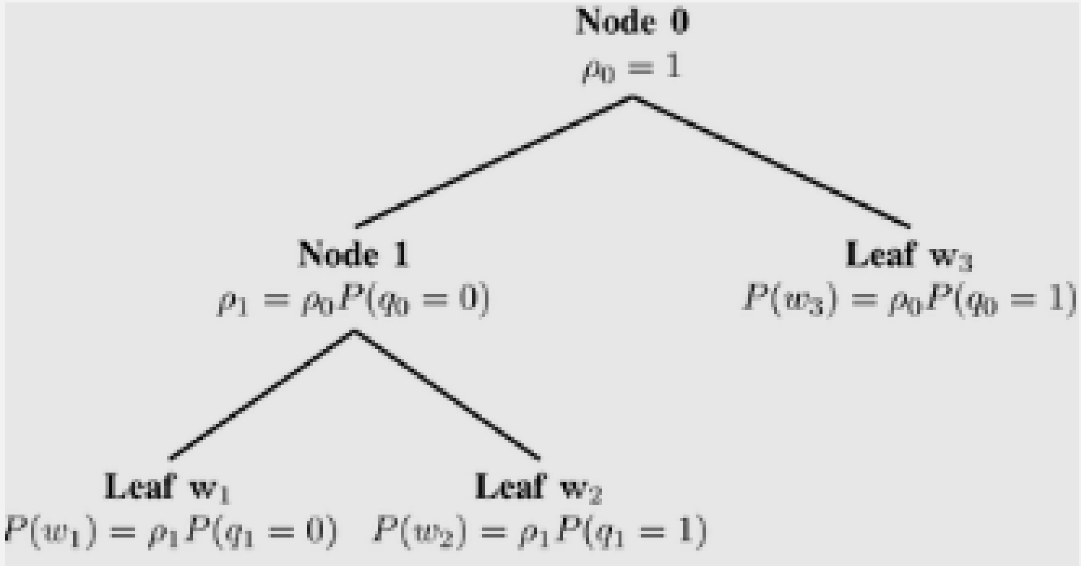
Computational Complexity : O(kh) -> $O(h\log_2 k)$

h = dimension of text representation, k = number of classes

Each node has a binary decision activation (e.g. sigmoid)

HS





*Distributed Representations of Words and Phrases ‘13 - Mikolov

05

Model: Bag of N-grams

- Bag of Words **invariant to word order**

Ex> Jane likes monkey = Monkey likes Jane

- **Bag of n-grams** as additional features (storing partial info about **position)**

- Bigram important for classification where word order is important

- **Hashing Trick** to retain memory efficiency

Ex> Bigrams(two undividing tokens)

<Last donut at the night>

<Last, Last donut, donut at, at the, the night, night>

06 Datasets / Code

[Datasets]

AG's news corpus	http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html
Sogou news corpus	https://www.sogou.com/labs/resource/cs.php
DBPedia ontology dataset	https://wiki.dbpedia.org/datasets
Yelp reviews	https://www.yelp.com/dataset
Yahoo! Answers dataset	https://webscope.sandbox.yahoo.com/catalog.php?datatype=l
Amazon reviews.	https://snap.stanford.edu/data/

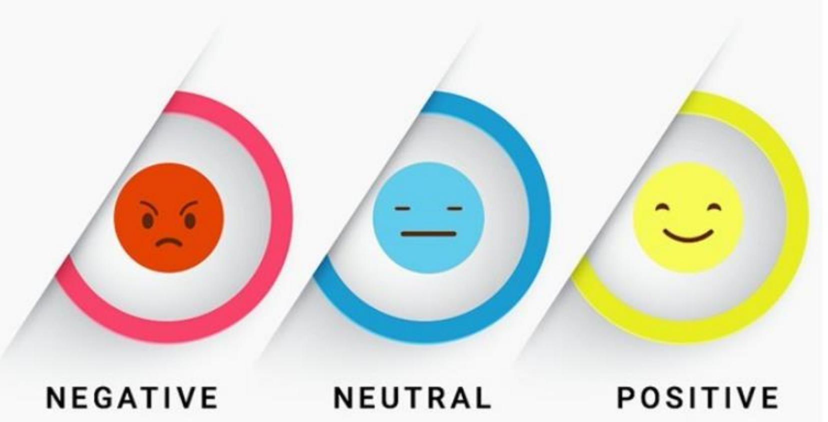
YFCC100M http://projects.dfki.uni-kl.de/yfcc100m/

[Code]

https://github.com/facebookresearch/fastText

07 Tasks

[Task 1 : Sentiment Analysis]



Input: Sentence/Document
Output: Sentiment Label

[Task 2 : Tag Prediction]

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

Input: Sentence/Document
Output: Tag

08 Settings & Basline/Sota

[Setting]

1) 5 Epochs
2) Decaying LR = [0.05, 0.1, 0.25, 0.5]
3) Stochastic gradient descent
4) Hidden units = 10 (Task1)
   50/200 (Task2)

[Baseline/Sota]

Task 1

1) char-CNN (Zhang and LeCun '15)
2) VDCNN (Conneau et al '16)

Task 2

1) Tagspace (Weston et al '14)

09

Task 1: 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

- Outperforms char-CRNN and baselines
- Adding bigram increases **1~4% accuracy** (trigram 97.1% on Sogou)

Task 1: 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

- VDCNN overall slightly higher => fastText **no pretrained word embedding**

09

Task 1: Sentiment Analysis

	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

- Methods using convolutions are **orders of magnitude slower**
- Max 15000x speed up with larger datasets

10 Task 2 : Tag Prediction

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

-fastText/fastText with bigrams outperforms Tagspace

10 Task 2 : Tag Prediction

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

- fastText performs upto 600 times faster at test time

11 Conclusion

fastText

1) Employs a simple yet efficient baseline for **text classification** for **large data sets.**

2) Utilizes **hierarchical softmax** & **N-gram features** to boost efficiency

3) On **par with accuracy** to Sotas on sentiment analysis & tagging but **exceedingly fast** in the train and test process

Q&A