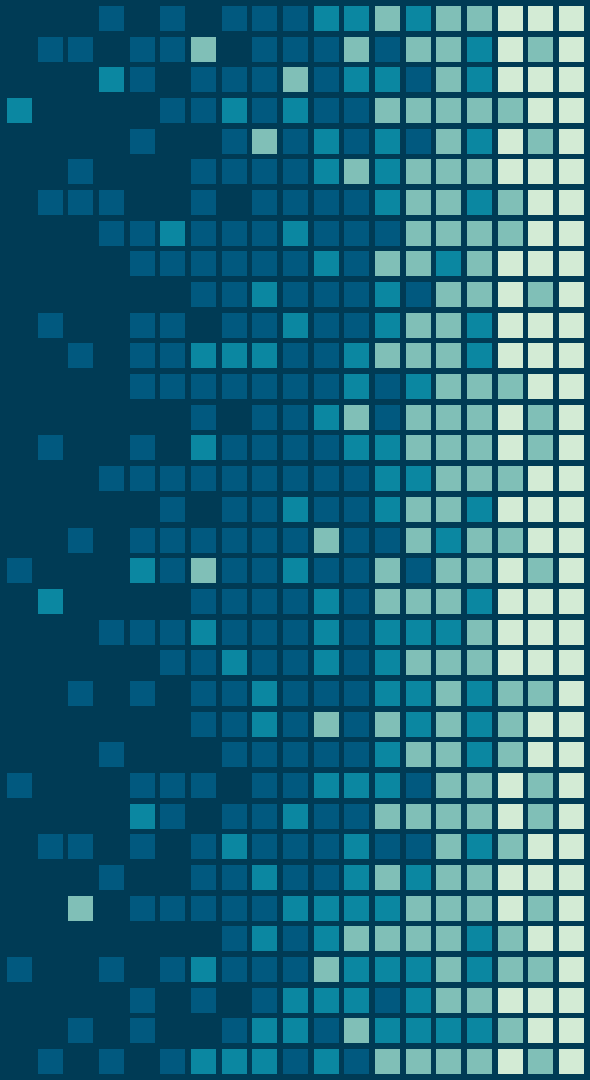
FastText: a library for efficient text classification



1.
Text Classification Task

Breve introduzione

La nascita del NLP

- Big Data: rendono possibile l'apprendimento
- Text classification fa parte di NLP
- Alcuni task specifici:
  - Sentence classification
  - Sentiment analysis
  - Rate Prediction

Text Classification Task

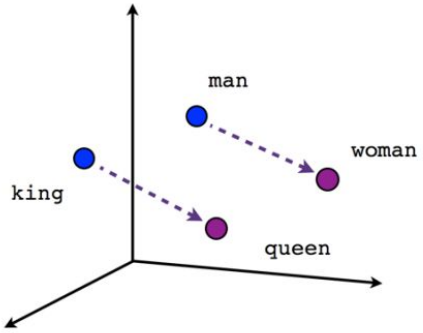
- Obiettivo: attribuire un'etichetta ad un nuovo documento mai osservato
- Dati: dataset di documenti etichettati con un numero finito di classi
- Algoritmo: dato un documento e la classe di appartenenza apprende quali sono le parole discriminanti per tale classe

2.
FastText Architecture

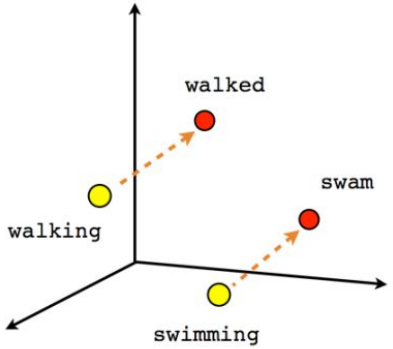
Word embeddings

- Le parole devono essere convertite in vettori per essere fornite in input ad un algoritmo
- I word embeddings sono vettori costruiti in modo ragionato per rappresentare le parole
- Parole simili sono rappresentate da vettori molto vicini tra loro nello spazio
- I word embeddings sono solitamente appresi in NLP

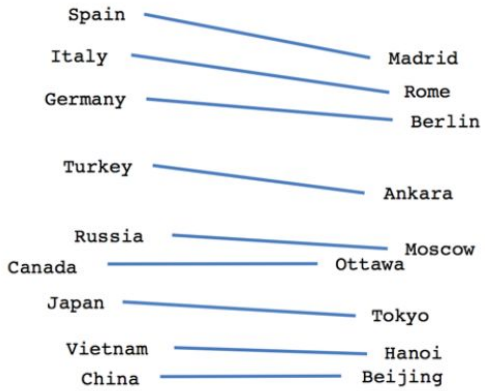
Relazioni tra vettori



Male-Female

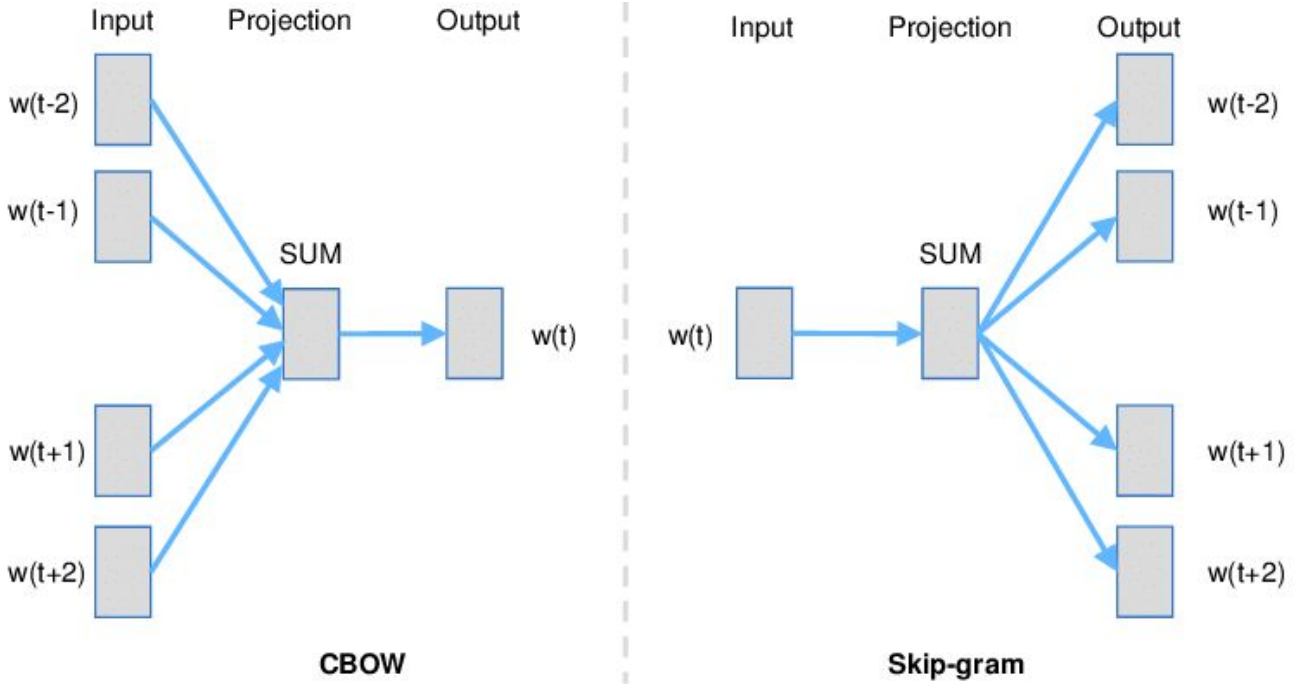


Verb tense



Country-Capital

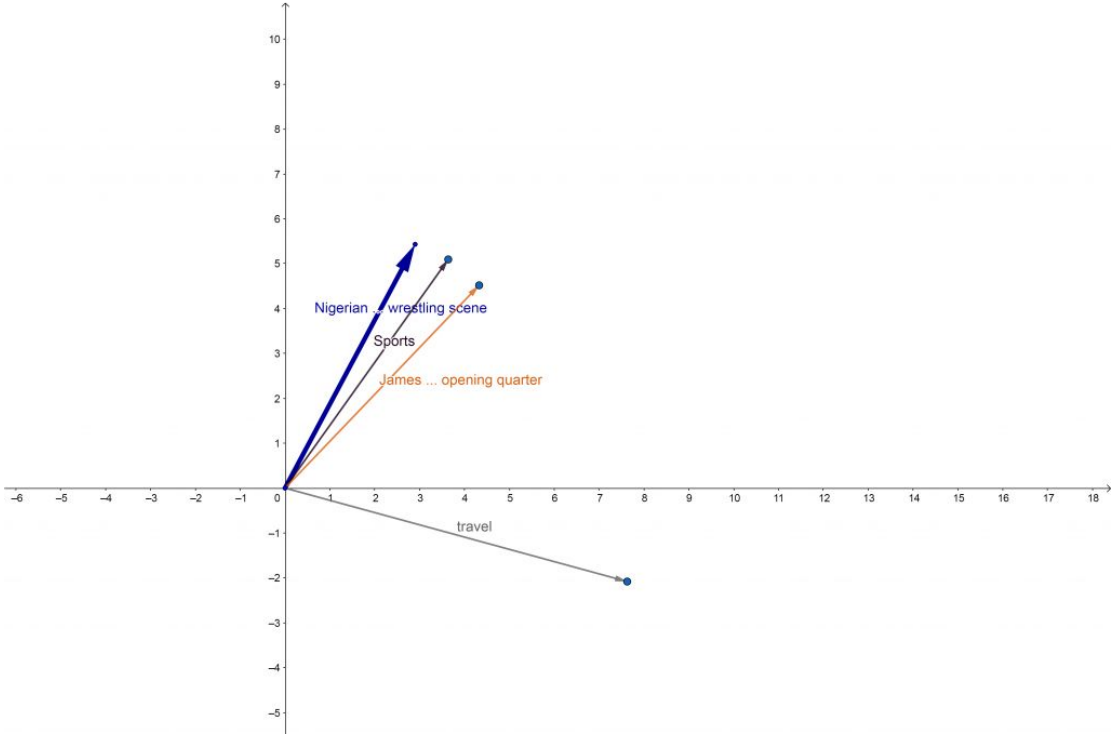
Apprendimento di word embeddings



FastText classifier: obiettivo

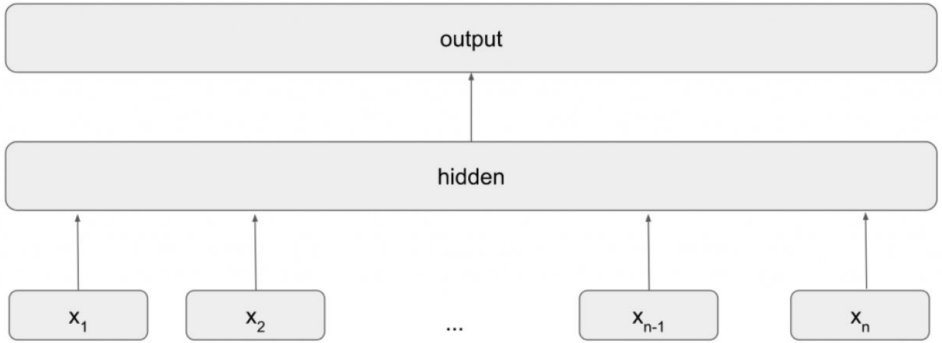
- Classificatore di testi: dato un documento predice una classe
- Interpretazione apprendimento: documenti e label sono vettori in un vector space di word embeddings

FastText classifier: interpretazione



FastText classifier: architettura

- Rete neurale con un solo hidden layer
- Input: parole di un documento
- Output: classe del documento



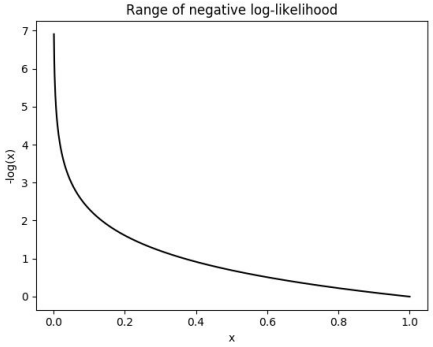
Fase forward (input - hidden)

- Costruzione della bag of words del documento in input
- La bag of words viene data in input alla rete
- Tramite una matrice di lookup tra input e hidden layer, la rete preleva i word embeddings per le parole passate
- L'hidden layer restituisce un vettore che rappresenta l'intero documento

Fase forward (hidden – output)

- Calcolo degli score per ogni etichetta
- Softmax: conversione degli score in probabilità
- Aggiornamento funzione errore
- Calcolo del gradiente della funzione errore
- Inizio fase backward

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

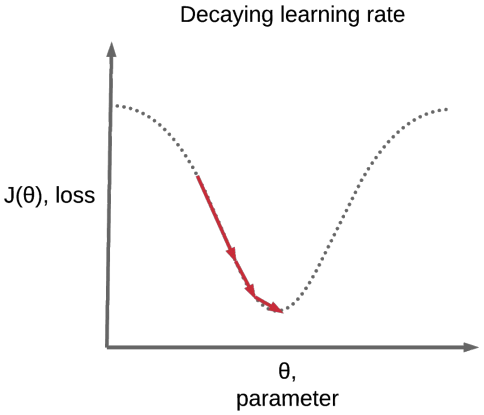


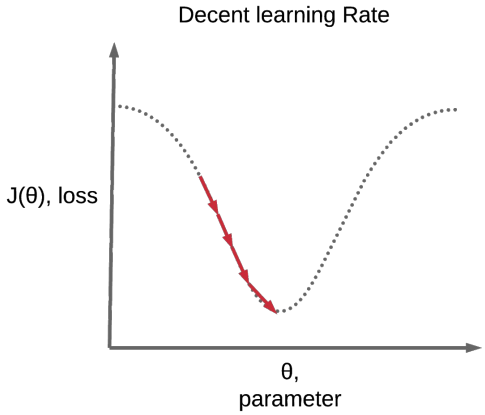
Fase backward

- Output - hidden: avvicinamento della classe corretta al documento
- Hidden - input: apprendimento dei word embeddings

Caratteristiche algoritmo

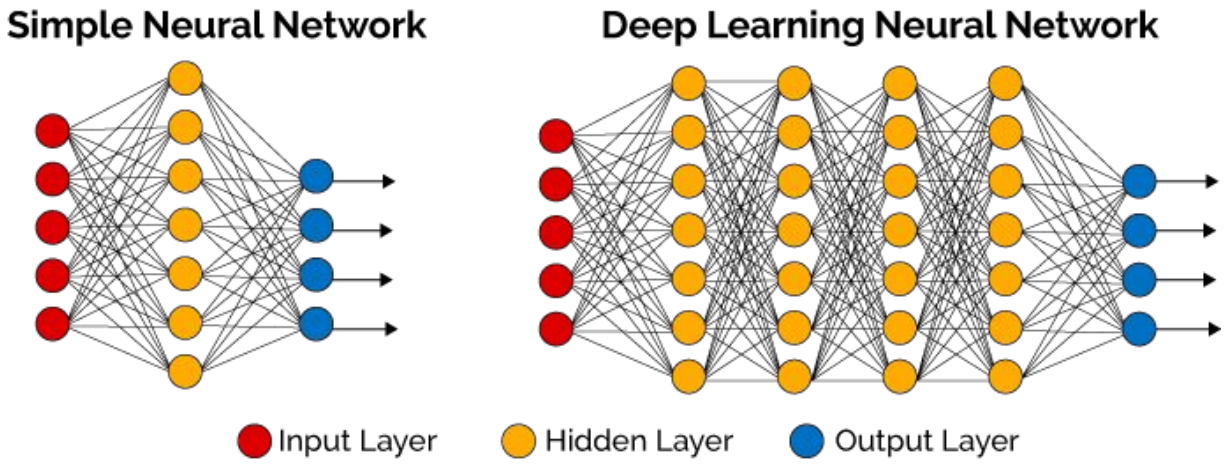
- Stochastic gradient descent
- Decaying learning rate





FastText VS deep learning

- Apprendimento più veloce
- Accuratezza in linea con i modelli deep learning



3.
Gli iperparametri di FastText

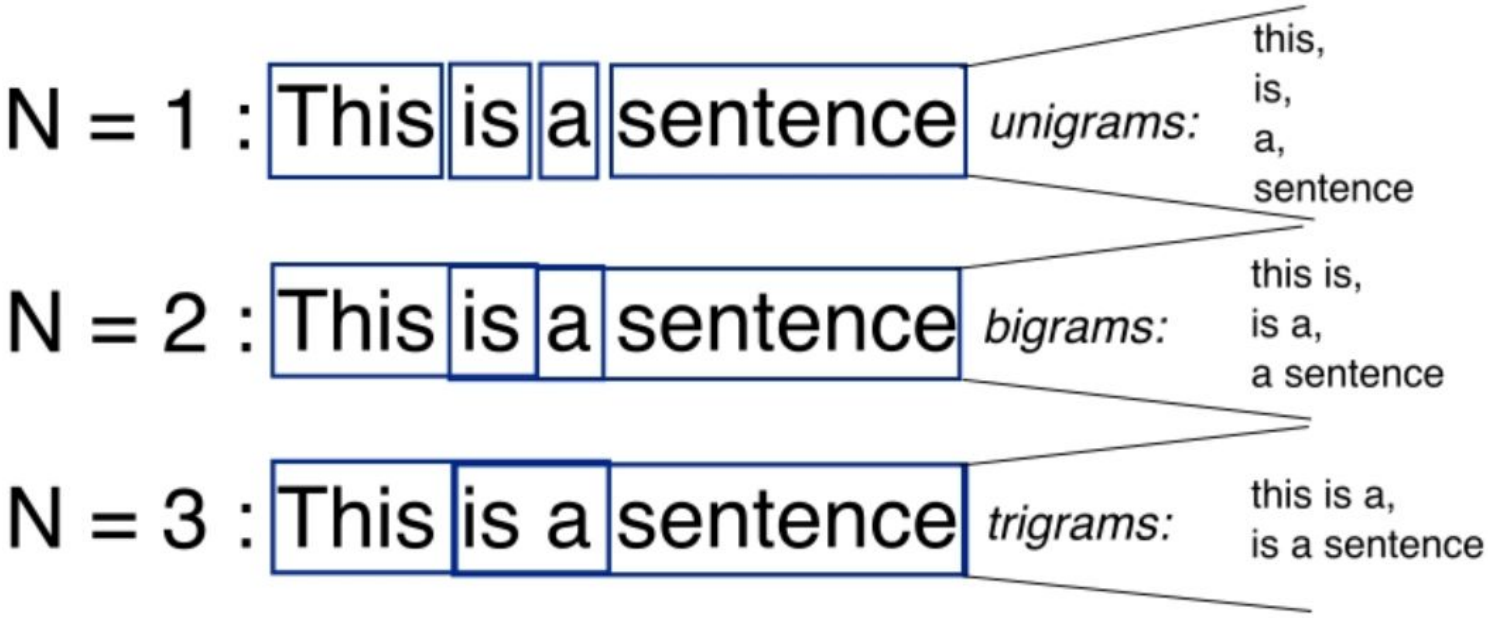
Numero di epoche

- Epoca: passaggio di tutti gli esempi di training durante l'apprendimento
- Può influire sulle performance del modello
- Può essere scelto in base alle performance su un validation set
- Deve essere scelto con attenzione

Learning rate e numero di n-grams

- Learning rate: determina la grandezza dell'aggiornamento ai pesi del modello durante il training
- N-grams: permette di settare il numero di n-grams di cui tener conto. Di default vengono usati gli uni-grams. Questo parametro influisce particolarmente sulle performance del modello

Cosa sono gli n-grams



Dimensione dei word embeddings

- Regola la dimensione dei word embeddings generati in fase di apprendimento
- Scelta della dimensione:
  - Indipendente dalla dimensione del vocabolario
  - Tradeoff tra dimensione e facilità di utilizzo da parte del modello
  - Scelta tramite validazione

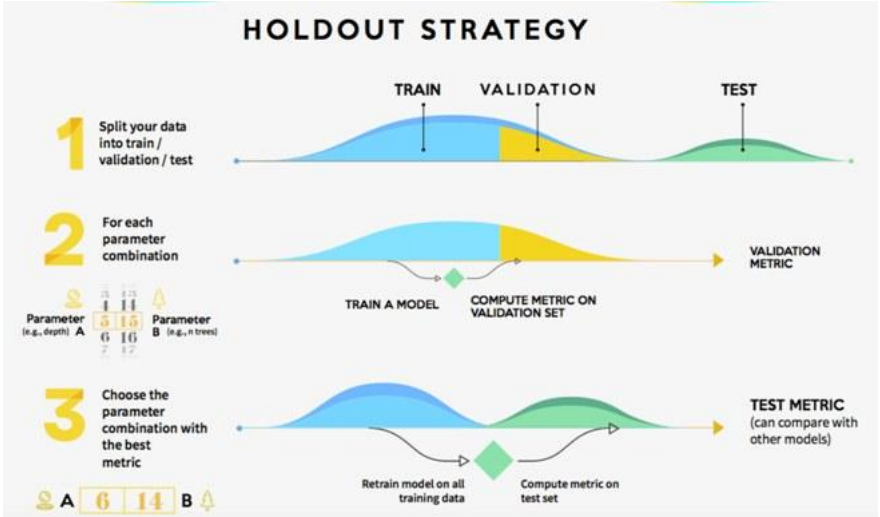
Tuning iperparametri

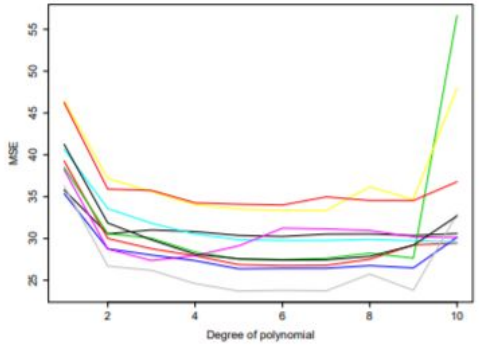
- Ricerca dei valori ottimali per gli iperparametri
- Avviene tramite applicazione della procedura grid search
- Scegliere con attenzione il numero di valori da testare per ogni parametro

Tuning iperparametri: hold-out

- Procedura usata nel paper
- Divisione training set in training set e validation set
  - Training set: usato per l'allenamento del modello
  - Validation set: usato esclusivamente per testare le performance del modello

Hold-out validation

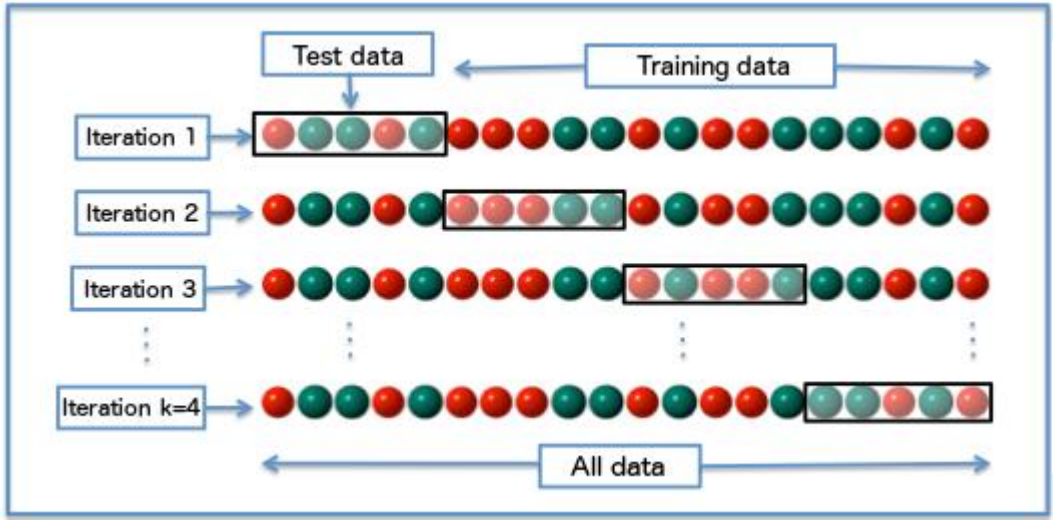


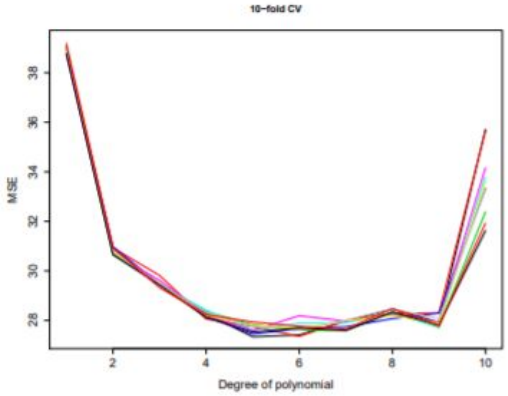


Tuning iperparametri: K-fold CV

- Strategia migliore
- Il training set viene suddiviso in K insiemi di dimensione uguale
- Si allenano K modelli
  - Per ogni modello, K-1 insiemi vengono utilizzati per il training, il restante insieme viene utilizzato come validation set

K-fold cross validation





4.
Analisi esperimento

Obiettivo esperimenti

- Sentiment analysis: confronto tra FastText e altri classificatori di testo esistenti
  - Tempi di apprendimento di vari ordini di grandezza migliori
  - Accuratezza in linea con gli altri modelli
- Tag prediction: analisi di scalabilità su un dataset di grandi dimensioni

Sentiment Analysis: tabella 1

- Allenamento di FastText su 8 dataset
  - 5 epoche
  - Word emebeddings di dimensione 10
  - Learning rate scelti su un validation set
- Visualizzazione delle performance con o senza l'utilizzo di 2-grams
- Le prestazioni risultano in linea con quelle dei classificatori confrontati

Sentiment Analysis: tabella 1

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

Tabella 1: critiche

- **Non** viene fornito il validation set per la procedura hold-out
- **Non** si conoscono le percentuali di split in training set e validation set
- **Non** si conosce il motivo della scelta di:
  - Dimensione embeddings (10)
  - Numero di epoche (5)
- Vengono forniti i valori dei learning rate ottimali per riprodurre l'esperimento ma la procedura utilizzata per identificarli è ignota

Risultati ottenuti: parametri forniti

	AG	Sogou	DBP	Yelp p.	Yelp f.	Yahoo A.	Amazon f.	Amazon p.
No bigram	91.5 / 91.4	93.9 / 93.8	98.1 / 98.3	93.8 / 93.8	60.4 / 60.4	72.0 / 72.0	55.8 / 55.6	91.2 / 91.1
Bigram	92.5 / 92.1	96.8 / 96.8	98.6 / 98.6	95.7 / 95.6	63.9 / 63.9	72.3 / 72.4	60.2 / 60.3	94.6 / 94.6

- I risultati sono in linea con quelli della tabella visualizzata nel paper
- La variabilità dei risultati è data da:
  - Variabilità dell'inserimento randomico degli esempi in training e test set
  - Funzionamento dell'algoritmo di ottimizzazione SGD

Tentativo di riproduzione dei risultati

- Utilizzo di hold-out validation per identificare i valori ottimali del learning rate
- Split 70/30 training/validation set
- Allenamento di 4 modelli differenti con i learning rate proposti (0.05, 0.1, 0.25, 1.0) sul training set
- Valutazione delle performance dei 4 modelli sul validation set

Risultati ottenuti: riproduzione

	AG	Sogou	DBP	Yelp p.	Yelp f.	Yahoo A.	Amazon f.	Amazon p.
No bigram	91.5 / 91.3	93.9 / 93.8	98.1 / 98.3	93.8 / 93.8	60.4 / 60.5	72.0 / 72.0	55.8 / 55.7	91.2 / 91.1
Bigram	92.5 / 92.1	96.8 / 96.7	98.6 / 98.5	95.7 / 95.6	63.9 / 63.9	72.3 / 71.8	60.2 / 59.8	94.6 / 94.6

- Learning rate che si discostano da quelli ottimali identificati dai ricercatori
- Risultati differenti in termini di accuratezza sui dataset (Yahoo A. e Amazon f.)
- Risultati comunque in linea con quelli proposti
- **Impossibilità** di applicare K-fold CV

Utilizzo di 3-grams

	AG	Sogou	DBP	Yelp p.	Yelp f.	Yahoo A.	Amazon f.	Amazon p.
Bigram	91.5	93.9	98.1	93.8	60.4	72.0	55.8	91.2
Trigram	92.5	97.1	98.6	95.9	64.3	72.4	60.7	95.0

- I ricercatori affermano che utilizzando i 3-grams le performance dovrebbe aumentare
- In alcuni dataset l'ordine delle parole è irrilevante nel determinare la classe, infatti non vi sono degli aumenti di accuratezza eccessivi

Sentiment Analysis: tabella 2

- I ricercatori dimostrano che i tempi di apprendimento di FastText sono di vari ordini di grandezza inferiori rispetto a quelli dei modelli confrontati
- Vengono riportati i tempi di apprendimento di **un'epoca** sugli 8 dataset
- I tempi sono ottenuti eseguendo il training su una CPU con 20 threads

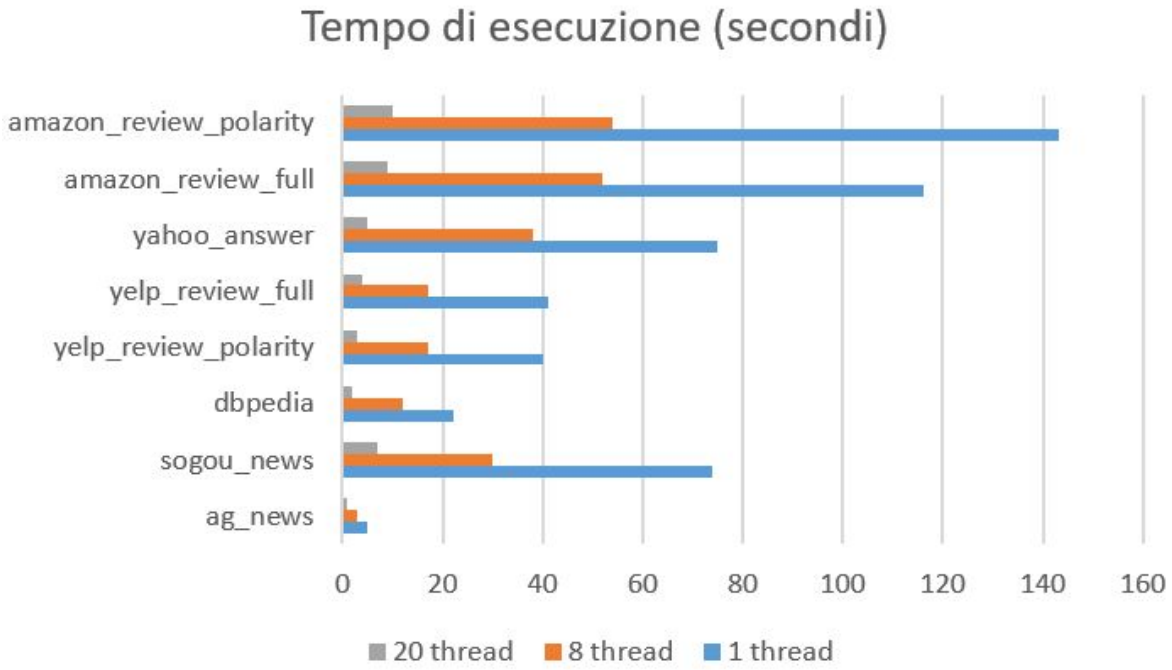
Sentiment Analysis: tabella 2

	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

Tentativo di riproduzione dei risultati

- Non è stato fornito il codice per riprodurre i risultati
- Non si è a disposizione di una CPU con 20 threads
- Si è verificato che l'andamento dei tempi di apprendimento su una CPU con 8 thread fosse in linea con i tempi proposti

Risultati ottenuti



Sentiment Analysis: tabella 3

- I ricercatori dimostrano che i risultati ottenuti con FastText sui dataset del paper *Tang et al. (2015)* sono in linea con i risultati ottenuti dai modelli proposti nello stesso

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

Critiche e tentativo di riproduzione

- Non sono stati forniti i dataset per riprodurre l'esperimento
- Sono stati cercati i dataset nel web
- I dataset non sono stati trovati in quanto vengono aggiornati di anno in anno
- La tabella risulta **irriproducibile**

Tag prediction: tabella 4

- Test di scalabilità di FastText su un dataset di grandi dimensioni
- I ricercatori dimostrano che FastText raggiunge accuratezze migliori con tempi di apprendimento di gran lunga inferiori rispetto ai modelli confrontati
- Training su CPU con 20 threads
- Testing su CPU con un solo thread

Tag prediction: tabella 4

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

Tentativo di riproduzione dei risultati

- I ricercatori hanno fornito il dataset preprocessato e diviso in training validation e test set
- A causa della vastità del dataset non è stato possibile riprodurre la tabella
- Il tempo di apprendimento per una sola combinazione di iperparametri è di 30 minuti su una CPU a 8 thread

5.
Efficacia del paper proposto

Punti dolenti del paper

- Architettura
  - I ricercatori fanno pensare che il modello sia un semplice modello lineare ed invece è una rete neurale
  - Nella descrizione dell'architettura vengono date per scontate troppe nozioni
  - L'analisi dell'architettura ha richiesto uno studio di circa **15 ore** del modello

Punti dolenti del paper (2)

- Ci si aspetta che nella pagina web ci sia una descrizione approfondita dell'architettura e invece non è così
  - Documentazione scarsa e poco chiara
- Nel raccontare gli esperimenti sono stati commessi diversi errori, i quali sono stati descritti nelle slide precedenti

Miglioramenti proposti

- Inserimento nel paper il link al sito web ufficiale
- Evitare di spiegare male l'architettura nel paper
- Inserire la spiegazione dell'architettura del sito web
- Spiegare in maniera più approfondita l'architettura
- Consigliare all'utente la lettura di diversi temi di NLP prima di addentrarsi nell'analisi dell'architettura
- Aggiungere una descrizione di come si sono ottenuti i risultati
  - Esplicitare la procedura utilizzata per il tuning degli iperparametri
  - Esplicitare lo split tra training e validation set

6.
Esempio completo su un dataset nuovo

Yelp dataset 2019

- Obiettivo: predirre il punteggio di una recensione dato il testo della recensione (Sentiment Analysis)
- Procedura:
  - Preprocessing dei dati
  - Split in training e test set (70/30)
  - Split in training e validation set (70/30)
  - Tuning iperparametri con procedura grid search
    - Dim: 10, 20
    - Epoche: 5, 10
    - Learning rate: 0.1, 0.25, 0.5, 1.0
    - N-grams: 1, 2
  - Test performance su test set

Preprocessing in FastText

- Rimozione delle stop words
- Conversione in minuscolo delle parole
- Rimozione di punteggiatura
- Mescolamento dei dati
- Formattazione del dataset per essere fornito in input al classificatore

Importanza del preprocessing

- Step preliminare all'allenamento di un modello
- Consiste in:
  - Formattazione dei dati per essere dati input all'algoritmo
  - Rimozione valori fuori range
  - Rimozione del rumore: preprocessing nel text classification
  - Imputazione di valori mancanti
- Step più importante del machine learning

Risultati procedura grid search

Parametri (lr e w dim)	Precisione	Parametri (lr e w dim)	Precisione	Parametri (lr e w dim)	Precisione	Parametri (lr e w dim)	Precisione
(0.1 5 1 10)	68.4	(0.25 5 1 10)	68.5	(0.5 5 1 10)	68.5	(1.0 5 1 10)	68.5
(0.1 5 1 20)	68.4	(0.25 5 1 20)	68.5	(0.5 5 1 20)	68.5	(1.0 5 1 20)	68.5
(0.1 5 2 10)	**71.4**	(0.25 5 2 10)	71.2	(0.5 5 2 10)	71.0	(1.0 5 2 10)	70.8
(0.1 5 2 20)	71.4	(0.25 5 2 20)	71.2	(0.5 5 2 20)	71.0	(1.0 5 2 20)	70.8
(0.1 10 1 10)	68.5	(0.25 10 1 10)	68.5	(0.5 10 1 10)	68.5	(1.0 10 1 10)	68.5
(0.1 10 1 20)	68.5	(0.25 10 1 20)	68.5	(0.5 10 1 20)	68.5	(1.0 10 1 20)	68.5
(0.1 10 2 10)	70.4	(0.25 10 2 10)	69.8	(0.5 10 2 10)	69.4	(1.0 10 2 10)	69.0
(0.1 10 2 20)	70.4	(0.25 10 2 20)	69.7	(0.5 10 2 20)	69.4	(1.0 10 2 20)	68.9

Performance sul test set

- La combinazione di iperaparametri che ha portato alle migliori performance sul validation set è:
  - Dimensione embeddings: 10
  - Numero di epoche: 5
  - Learning rate: 0.1
  - N-grams: 2
- Il modello con tali parametri è stato allenato sull'intero training set
- Precisione sul test set: 71.5%
  - Risultato in linea con i risultati ottenuti dai ricercatori nella tabella 3

Conclusioni

- FastText si è rivelato essere un classificatore veloce e preciso
  - La sua velocità rende più accurata la procedura di tuning degli iperparametri
  - La semplicità della sua architettura permette di effettuare training di grandi dataset anche su macchine normali
  - Grazie agli n-grams riesce ad ottenere accuratezze migliori della maggior parte dei modelli più complessi

Fonti

- Sito web ufficiale - https://fasttext.cc/
- Paper ufficiale - https://research.fb.com/wp-content/uploads/2016/07/eacl2017.pdf?
- FastText Tutorial - https://www.youtube.com/watch?v=4l_At3oalzk
- FastText by Piotr B. - https://www.youtube.com/watch?v=CHcExDsDeHU
- CBOW - https://iksinc.online/tag/continuous-bag-of-words-cbow/
- Average word vectors - http://yaronvazana.com/2018/09/20/average-word-vectors-generate-document-paragraph-sentence-embeddings/
- Learning rate settings - https://machinelearningmastery.com/learning-rate-for-deep-learning-neural-networks/
- Softmax function - https://ljvmiranda921.github.io/notebook/2017/08/13/softmax-and-the-negative-log-likelihood/
- Word2vec - https://en.wikipedia.org/wiki/Word2vec
- Embeddings dimension - https://www.quora.com/How-are-the-dimensions-i-e-300-400-picked-in-a-Word2vec-or-a-GloVE-algorithm-Im-just-confused-on-how-this-dimensionality-comes-into-play-for-each-of-the-10-000-words-in-corpus
- Word embeddings - https://www.youtube.com/watch?v=5PL0TmQhItY
- Libro su FastText - https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781789130997