A compressed sensing view of text embeddings and LSTMs

Sanjeev Arora, al. 2018

Salomé Do et Lucas Zanini
29/03/2019

ENSAE

Introduction

Introduction

L'article étudié a pour objectif de fournir :

- Un cadre d'analyse des ”text embeddings” basé sur le compressed sensing.
- Des résultats théoriques sur les embeddings fournis par des LSTM.

Table of contents

La représentation de texte comme problème de compressed sensing

Qu'est-ce qu'un text embedding?

Un text embedding est une représentation d'un texte sous la forme d'un vecteur dans $\mathbb{R}^d$, par exemple:

$$\text{'Le chien a mangé sa gamelle '} \rightarrow \begin{pmatrix} 0.45 \\ -0.76 \\ 0.94 \\ 0.67 \end{pmatrix}$$

Utile pour différentes tâches : classification de texte, calculs de similarités entre textes, clustering de textes.

Une première idée de text embedding

Supposons que l'on ait un vocabulaire $\mathcal{V} = \{v_1, ..., v_V\}$, de taille V, et un texte $x = w_1, ..., w_T$.

- **Bag of words** : on représente le texte par un vecteur qui compte les occurences dans le texte de chaque mot du vocabulaire :

$$x_{BoW} = \begin{pmatrix} 1 \\ 3 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} \sum_{t=1}^{T} 1_{\{w_t = v_1\}} \\ \sum_{t=1}^{T} 1_{\{w_t = v_2\}} \\ \sum_{t=1}^{T} 1_{\{w_t = v_3\}} \\ \vdots \\ \sum_{t=1}^{T} 1_{w_t = v_V} \end{pmatrix}$$

Problèmes : pas d'ordre des mots, sparse.

Une première idée de text embedding

- **Bag of-n-words** : On s'intéresse à des n-grammes :
  - Bigrammes de *'Le chien a mangé sa gamelle'* :
    {'Le chien', 'chien a', 'a mangé', 'mangé sa', 'sa gamelle'}
  - Trigrammes :
    {'Le chien a', 'chien a mangé', 'a mangé sa', 'mangé sa gamelle' }
- Pour $1 \leq k \leq n$, soit $\mathcal{V}_k = \{b_1, ..., b_{V_k}\}$ l'ensemble des bigrammes sur $\mathcal{V}$ (l'ordre des mots ne compte pas) :

$$B_k = \sum_{t=1}^{T-k+1} e_{\{w_t,\ldots,w_{t+k-1}\}} \quad \in \mathbb{R}^{V_k}$$

  La i-ème coordonnée de B_k correspond au nombre d'occurences du k-gramme b_i dans le texte.

Une première idée de text embedding

- Alors, une représentation du texte peut être :

$$x_{BonC} = [B_1, ..., B_n] \in \mathbb{R}^{\sum_{k=1}^{n} V_k}$$

- Approche plus complète que les simples bag-of-words.
- Cependant, toujours sparse, et de très haute dimension.
- Aucune information sémantique externe au texte (aucun 'prior knowledge' sur les mots).

Words embeddings et text embeddings

- Pour pallier à ces défauts, on adopte une autre approche : obtenir les text embeddings à partir de word embeddings.
- Algorithmes classiques : Word2Vec (2013), ELMo (2018), UMLfit (2018), BERT(2018)
- Représentations dans $\mathbb{R}^d$ des mots de $\mathcal{V}$ apprises sur un corpus de texte : intègre le 'prior knowledge'.
-

$$\text{'chien'} = \begin{pmatrix} 0.45 \\ -0.98 \\ \vdots \\ -0.56 \\ 0.2 \end{pmatrix} \in \mathbb{R}^d$$

Words embeddings et text embeddings

- Sens sémantique : on peut calculer une distance entre deux vecteurs, donc entre deux mots.

```
In [9]: model.wv.most_similar(['paracetamol'])
Out[9]:
[('codeine', 0.6557880640029907),
 ('tramadol', 0.5951346158981323),
 ('ibuprofene', 0.5070722103118896),
 ('doliprane', 0.5006594657897949),
 ('nefopam', 0.4992637038230896),
 ('propacetamol', 0.4889230728149414),
 ('dextropropoxyphene', 0.4812668859586487),
 ('tylenol', 0.4710041284561157),
 ('aspirine', 0.470241516828537),
 ('diclofenac', 0.4693360924720764)]
```

- Question : aggréger ces représentations de chaque mot à une représentation unique pour le texte?

DisC embeddings

- $\mathcal{V} = \{v_1, ..., v_V\}$, embeddings pour chacun des mots du vocabulaire : $x_{v_1}, ..., x_{v_V} \in \mathbb{R}^d$
- Soit $b = \{v_{i_1}, ..., v_{i_k}\} \in \mathcal{V}_k$ un k-gramme de $\mathcal{V}$. Alors :

$$x_b = \odot_{t=1}^{n} x_{v_{i_t}} \in \mathbb{R}^d$$

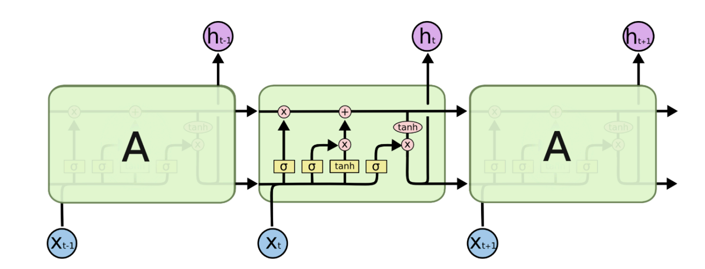
- Pour le texte complet :

$$z^{(n)} = \left[\sum_{t=1}^{T} x_{w_t}, ..., \sum_{t=1}^{T-n+1} x_{\{w_t, ..., w_t+n-1\}} \right] \in \mathbb{R}^{nd}$$

- Dimension plus faible que celle des BonC, représentation dense.
- Les DisC embeddings peuvent s'écrire comme une version compressée des Bag-of-n-grams:

$$z^{(n)} = Ax_{BonC}$$

LSTMs embeddings



- Réseaux de neurones récurrents.
- Permettent une représentation du texte qui garde en mémoire une partie du texte.
- L'état de sortie est utilisé comme Embedding du texte:

$$z_{LSTM} = h_T$$

Questions

Questions à ce stade :

- Comment évaluer ces différents embeddings?
- Dans l'écriture qui lie Bag-of-n-grams et DisC,

$$z^{(n)} = Ax_{BonC}$$

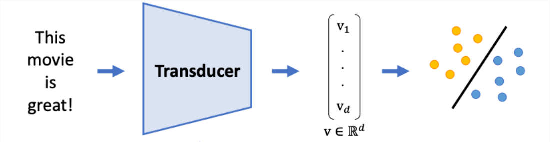
  Quelles sont les propriétés de la matrice de compression A?

Evaluation des embeddings

On peut penser à deux manières d'évaluer les text embeddings :

- **En compressed sensing** : Reconstrution du signal des Bag-of-n-grams à partir du signal compressé DisC
- **En apprentissage statistique** : performances du signal compressé comparativement au signal original sur une tâche de classification (linéaire ici)

Dans cette partie, on s'intéresse à l'apprentissage dans le domaine compressé avec le schéma suivant :



Apprentissage dans le domaine compressé

Théorème du compressed learning

Hypothèses :

- Ensemble d'apprentissage $S = \{(x_i, y_i)_{i=1,\dots,m}\} \subset \mathcal{X} \times \{-1, 1\}$, $\mathcal{X} \subset \mathbb{R}^N$, $0 \in \mathcal{X}$. $(x_i, y_i) \overset{i.i.d}{\sim} \mathcal{D}$ et $||x_i|| \leq R$ pour tout $i = 1, \dots, m$
- Ensemble d'apprentissage compressé $S_A = \{(Ax_i, y_i)_{i=1,\dots,m}\} \in \mathbb{R}^d \times -1, 1$
- Fonction de perte l, λ-Lipschitz, convexe. $l_{\mathcal{D}}$ notation du risque théorique, l_S notation du risque empirique sur S. $L(w) = l(w) + \frac{1}{2C}||w||_2^2$ régularisation l_2 de l.
- $w_0 \in \mathbb{R}^N$ classifieur linéaire qui minimise le risque théorique $l_{\mathcal{D}}$.
- $\hat{w}_A \in \mathbb{R}^d$ minimiseur du risque empirique régularisé sur l'échantillon compressé L_{S_A}.
- $A \in \mathbb{R}^{d \times N}$.

Théorème du compressed learning

Condition RIP généralisée : $A \in \mathbb{R}^{d \times N}$ est $(\mathcal{X}, \varepsilon)$-RIP si pour tout $x \in \mathcal{X}$,

$$(1-\varepsilon)||x||_2 \leq ||Ax||_2 \leq (1+\varepsilon)||x||_2$$

Théorème du compressed learning

Théorème :
Si A matrice de compression satisfait RIP($\Delta\mathcal{X}, \varepsilon$), avec $\Delta\mathcal{X} = \{x - x' : x, x' \in \mathcal{X}\}$, alors avec probabilité $1 - 2\delta$:

$$l_{\mathcal{D}}(\hat{w}_A) \leq l_{\mathcal{D}}(w_0) + \mathcal{O}\left(\lambda R ||w_0||_2 \sqrt{\varepsilon + \frac{1}{m}\log\frac{1}{\delta}}\right) \tag{1}$$

LSTM Embeddings

LSTM Embeddings

- Objectif : prouver un résultat similaire au théorème précédent pour les LSTM embeddings.

LSTM Embeddings

Deux difficultés:

- Les LSTM Embeddings ne peuvent **pas** (en general) s'exprimer sous la forme

$$z^{LSTM} = A^{LSTM} X_{BonC}$$

- L'embedding dépend des word embeddings utilisés en entrée du LSTM.

LSTM Embeddings

Solutions:

- montrer qu'il existe un LSTM vérifiant

$$z^{LSTM} = z^{DisC}$$

- Utiliser des word embeddings particuliers:

$$x_{v_i} \overset{iid}{\sim} \mathcal{U}(\{-1, 1\}^d)$$

LSTM Embeddings

Hypothèses :

- $(x_i, y_i) \overset{iid}{\sim} \mathcal{D}$ sur l'ensemble des BonG de longueur inférieure à T.
- Word embeddings iid selon une loi $\mathcal{U}(\{\frac{-1}{\sqrt{d}}, \frac{1}{\sqrt{d}}\})$

Theorem

Pour $d = \tilde{\Omega}(\frac{T}{\varepsilon} \log \frac{nV_n^{max}}{\gamma})$*, il existe un LSTM avec une mémoire de taille* $\mathcal{O}(nd)$ *tel qu'on ait avec probabilité* $(1-\varepsilon)(1-2\delta)$*,*

$$l_{\mathcal{D}}(\hat{w}_{LSTM}) \leq l_{\mathcal{D}}(w_0) + \mathcal{O}\left(||w_0||_2 \sqrt{\varepsilon + \frac{1}{m} \log \frac{1}{\delta}}\right) \tag{2}$$

LSTM Embeddings

Etapes de la preuve:

1) Il existe un LSTM vérifiant

$$z_{LSTM} = z_{DisC}$$

$$\mathcal{T}_f(v_{w_t}, h_{t-1}) = \begin{pmatrix} \mathbf{1}_{nd} \\ \mathbf{0}_{(n-1)d} \end{pmatrix}$$

$$\mathcal{T}_i(v_{w_t}, h_{t-1}) = \begin{pmatrix} \mathbf{0}_{d\times nd} & \cdots & \mathbf{0}_{d\times d} \\ \vdots & I_{(n-2)d} & \mathbf{0}_{(n-2)d\times d} \\ \vdots & \ddots & I_d \\ \vdots & \ddots & \mathbf{0}_{d\times d} \\ \mathbf{0}_{(n-2)d\times nd} & I_{(n-2)d} & \mathbf{0}_{(n-2)d\times d} \end{pmatrix} h_{t-1} + \begin{pmatrix} \mathbf{1}_d \\ \mathbf{0}_{(n-1)d} \\ \mathbf{1}_d \\ \mathbf{0}_{(n-2)d} \end{pmatrix}$$

$$\mathcal{T}_g(v_{w_t}, h_{t-1}) = \begin{pmatrix} C_1 I_d \\ \vdots \\ C_n d^{\frac{n-1}{2}} I_d \\ I_d \\ \vdots \\ I_d \end{pmatrix} v_{w_t}$$

Etapes de la preuve:

2) La matrice d'embedding A_{DisC} vérifie la propriété *RIP*.
$\rightarrow$ Basé principalement sur le résultat suivant:

Theorem

Si $\sqrt{d}A$ est une matrice associé à un système orthonormal borné (pour une valuer spécifique de d), ie

$$\begin{cases} \sqrt{d}A_{i,j} = \varphi_i(x_j) \\ \mathbb{E}[\varphi_i(x)\varphi_j(x)] =_{i=j} \\ \|\varphi_i\|_\infty \leq B \end{cases}$$

Alors A vérifie la propriété RIP.

LSTM Embeddings

Remarques:

- Word embeddings aléatoires
- Une configuration de LSTM bien précise

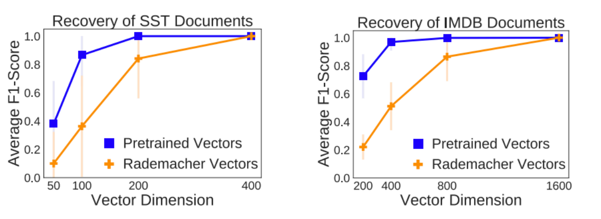
Reconstruction du signal

Reconstruction du signal

Une autre façon d'évaluer la qualité de compression des embeddings DisC (et d'évaluer la qualité des word embeddings sur lesquels ils sont basés) consiste à observer la qualité de reconstruction des Bag-of-n-Grams.
Peut-on reconstruire qualitativement le signal des Bag-of-n-grams à l'aide des embeddings DisC?

Reconstruction du signal

- Si les DisC utilisent des embeddings aléatoires : reconstruction en temps polynomial.
- Si les DisC utilisent des embeddings pré-entraînés : plus difficile d'utiliser la théorie en compressed sensing, mais résultats empiriques : meilleure reconstruction que les embeddings aléatoires!



Conclusion