Deep-Learning for structured data

Alex Allauzen

2017-11-21

Outline

Plan

Learning from structured data

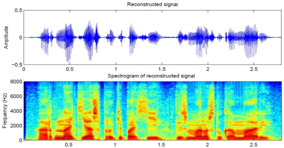
Structure in data:

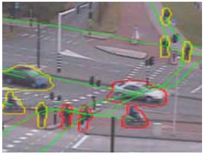
- sequence / image
- document / video
- graphs

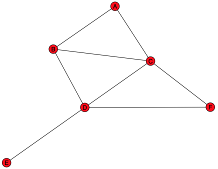
Heterogeneity / Ubiquity

- multi-view
- multi-task

$\rightarrow$ data representation

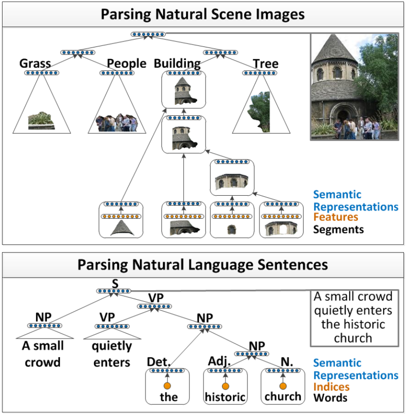






Learning structure from data

- Large vocabulary
- Sequence 2 sequence (transduction)
- Parsing
- Language / Sequence generation



Application to Natural Language Processing

Does the structure matter ?

Grammar	Jane went to the store
	store to Jane went the.
	Jane went the store.
	Jane go to the house.
Noise	Jane goed to the store.
Semantic	The store went to Jane.
	The food truck went to Jane.

"Linguistic" and machine learning issues

Linguistic / The grammar

- Morphology
- Syntax
- Semantics/World Knowledge Discourse
- Pragmatics
- Multilinguality

Machine (Deep) Learning issues

- The architecture of the model must capture the structure (grammar) of the data,
- of the task.
- Trade-off between complexity, expressivity and efficiency
- How to define a loss function ?

Applications

- Speech Recognition / Synthesis
- Machine translation
- Image reconstruction, Captioning
- Recommendation (products, music, ...)
- Prediction (diagnosis, rating, ...)

Outline of the course

- Word embeddings
- Language modeling (word prediction)
- Sequence representation
- Large vocabulary issues
- Sequence 2 sequence models
- Attention model

Plan

Opinion analysis in texts

Some applications

- Online customer reviews
- Advertisement targeting
- Public relations/marketing
- Analytics/reputation mining
- Web content filtering ...

Case study: movie reviews

given the text: is the text positive / negative

Is it difficult ?

Contextual polarity

- The movie was (**not**) predictable
- The movie was unpredictable
- The car steering is unpredictable

Idioms ...

How can anyone sit through this movie?

Different kinds of information

My wonderful boyfriend took me to see this movie for our anniversary. It was terrible.

When negative is positive

The slow, methodical way he spoke. I loved it! It made him seem more arrogant and even more evil.

Words and structure

To represent text, consider:

- a text is a structured sequence made of words;
- a word is a discrete symbol;
- belonging to a *finite* set, the vocabulary.

Compositionality

- The meaning of a complex expression is determined by the meanings of its constituent,

  words, lexical semantic, morphology

- and the rules used to combine them.

  syntax, pragmatic

This principle is also called Frege's principle.

Plan

Bag of words (BOW)

this movie is just great , with a great music , while a bit long

vocabulary	binary bag	count bag	tfidf bag	...
the	0	0	0.01	...
awesome	0	0	1.2	...
this	1	1	0.1	...
long	1	1	2.5	...
great	1	2	0.9	...
...	...	...	...	...

Binary bag

The text:

this *movie is just* ***great*** *,*
with a ***great*** *music , while a bit* ***long***

The vocabulary : (the, **this**, awesome, **long**, **great**)

$$\Rightarrow \boldsymbol{x} = \begin{array}{cl} \left(\begin{array}{c} 0 \\ \mathbf{1} \\ 0 \\ \mathbf{1} \\ \mathbf{1} \end{array}\right) & \begin{array}{l} \textit{the} \\ \textbf{this} \\ \textit{awesome} \\ \textbf{long} \\ \textbf{great} \end{array} \end{array}$$

Scoring for one class / linear classifier

$$w_0 + \mathbf{w}^t \boldsymbol{x} = w_0 + \begin{pmatrix} 0 \\ \mathbf{1} \\ 0 \\ \mathbf{1} \\ \mathbf{1} \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \end{pmatrix} = w_0 + w_2 + w_4 + w_5$$

The class is positive ($y = 1$) if

$$\begin{aligned} w_0 + w_2 + w_4 + w_5 &> 0 \\ w_2 + w_4 + w_5 &> -w_0 \\ w_{this} + w_{long} + w_{great} &> \text{threshold} \end{aligned}$$

Multi-class

With four classes, do it four times:

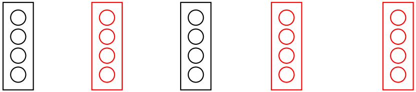
$$\begin{pmatrix} w_1^1 & w_2^1 & w_3^1 & w_4^1 & w_5^1 \\ w_1^2 & \dots & & & w_5^2 \\ w_1^3 & \dots & & & w_5^3 \\ w_1^4 & \dots & & & w_5^4 \end{pmatrix} \times \begin{pmatrix} 0 \\ \mathbf{1} \\ 0 \\ \mathbf{1} \\ \mathbf{1} \end{pmatrix}$$

Look-up and composition

$\boldsymbol{x} = \begin{pmatrix} 0 & \mathbf{1} & 0 & \mathbf{1} & \mathbf{1} \end{pmatrix}$

$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow$

look-up



$\mathbf{w}_1 \quad \mathbf{w}_2 \quad \mathbf{w}_3 \quad \mathbf{w}_4 \quad \mathbf{w}_5$

$\downarrow \quad \downarrow \quad \downarrow$

composition $\quad \mathbf{w}_{this} + \mathbf{w}_{long} + \mathbf{w}_{great}$

$$\Downarrow$$

softmax $\quad \left(\mathbf{w}_{this} + \mathbf{w}_{long} + \mathbf{w}_{great} \right)$

The simplest model

Assumptions

- A document is a (binary) bag of features: words, POS, bigram, ...
- For each class and and each feature: one parameter $w_{i,j}$.
- The composition is the sum.

This is a multinomial logistic regression model !

Two views

One parameter per word and per class !

Multi-class model

For each class j , a set of parameters: $(w_{i,j})_{i=1}^{K}$

Symbol embeddings

For each word i , a set of parameters: $(w_{i,j})_{j=1}^{C}$

The word representation could be shared among classes.

Representing words in a high-dimension space (K)

$$\begin{array}{l} \textit{the} \\ \textbf{this} \\ \textit{awesome} \\ \textbf{long} \\ \textbf{great} \end{array} \begin{pmatrix} 0 \\ \mathbf{1} \\ 0 \\ \mathbf{1} \\ \mathbf{1} \end{pmatrix} \rightarrow \underbrace{(\mathbf{v}_2, \mathbf{v}_4, \mathbf{v}_5)}_{\text{one vector per word}} \rightarrow \underbrace{(\mathbf{v}_2 + \mathbf{v}_4 + \mathbf{v}_5)}_{\text{document = sum}}$$

Motivation

- the cat is walking in the bedroom
- the dog is running in the room

Learn similar representations ($\mathbf{v}$) for similar words.
A shared representation for prediction.

A simple model of a document

$$\mathbf{R} \times \boldsymbol{x} = \begin{pmatrix} \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 & \mathbf{v}_4 & \mathbf{v}_5 \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix} \times \begin{pmatrix} 0 \\ \mathbf{1} \\ 0 \\ \mathbf{1} \\ \mathbf{1} \end{pmatrix} = \mathbf{v}_2 + \mathbf{v}_4 + \mathbf{v}_5 = \mathbf{d}$$

Classification:

$$P(y|\boldsymbol{x}) = softmax(\mathbf{W^o d})$$

Parameters :

$$\boldsymbol{\theta} = (\mathbf{R}, \mathbf{W^o})$$

$\rightarrow$ to learn

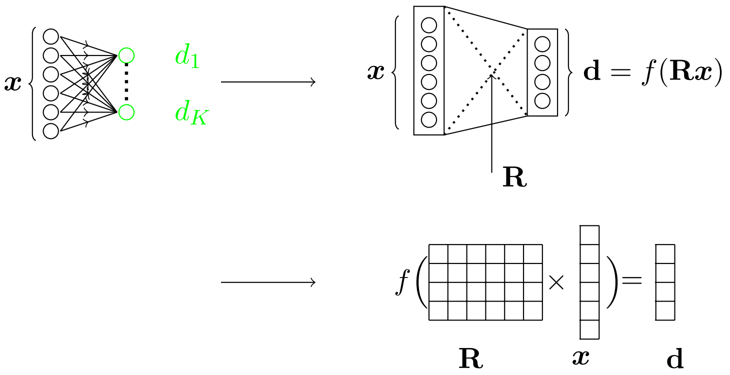
Word embeddings

Definitions

- To each word, a continous vector is associated: its embedding.
- The matrix $\mathbf{R}$ is called the look-up table and store the word embeddings.

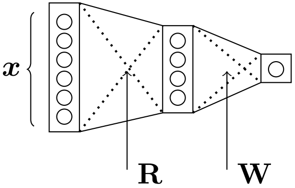
- The term *look-up* comes from the real operation: $\mathbf{R} \times \boldsymbol{x}$ is only theoritical !
- No computational cost, only storage and trainability issues.
- Pre-trained, fine-tuned, ...

A first neural network - 1



- The input x is a bag of (binary) features.
- $\mathbf{d}$: an internal representation of x, a hidden layer of parameters $\mathbf{R}$

A first neural network - 2



$P(y=1|\boldsymbol{x};\boldsymbol{\theta}) = \sigma(\boldsymbol{W}^o \times \mathbf{d})$

- $\boldsymbol{x}$: $(|\mathcal{V}|, 1)$
- $\mathbf{R}$: $(K, |\mathcal{V}|)$
- $\mathbf{d}$: $(K, 1)$ $\quad \mathbf{d} = \mathbf{R} \times \boldsymbol{x}$
- $\mathbf{W}$: $(1, K)$
- y : $(1, 1)$ $\quad y = \sigma(\mathbf{W^o} \times \mathbf{d})$

Learning the parameters

For $\mathbf{W}$

- Given $\mathbf{R}$, it's easy !
- Compute the loss gradient *w.r.t* $\mathbf{W}$

For $\mathbf{R}$

- Compute the loss gradient *w.r.t* $\mathbf{R}$

$\rightarrow$ Back-propagation of the gradient

Plan

Summary

How to represent a structured set of discrete symbols/words ?

- Each discrete symbol is associated to real valued vector, its embedding.
- Embeddings are considered as trainable parameters.
- We now need a composition method of these symbols.

Continuous Bag of words

- Bag of words assumption + word embeddings.
- The structure is simply discarded.
- Can we keep the idea of word embeddings and really handle structured inputs ?