EDAN95

Applied Machine Learning

http://cs.lth.se/edan95/

Lecture 7: Recurrent Neural Networks

Pierre Nugues

Lund University
Pierre.Nugues@cs.lth.se
http://cs.lth.se/pierre_nugues/

November 26, 2018



Natural Language Processing

High-level applications:

- Spoken interaction: Apple Siri, Google Assistant, Amazon Echo
- Speech dictation of letters or reports: Windows 10, macOS
- Question answering: IBM Watson and Jeopardy!

The inner engines of these applications are powered by neural networks. Big change from the

- 1980's (rules),
- 1990's (Bayes), and
- 2000's (SVM and logistic regression, still usable for most tasks).

Neural net expansion started in 2010. What in 2030?

Scope of this Course

Applications we will consider:

- Text categorization
- Word or segment categorization
- 3 Translation: Google Translate, DeepL, Bing translator, etc.

Text Categorization

- spam/not spam
- Language identification, French, English, or Spanish? (https://github.com/google/cld3)
- Sentiment analysis: Is this comment on my favorite tooth paste positive, neutral, or negative?
- Newswire categorization.

Text Categorization: The Reuters Corpus

```
<title>USA: Tylan stock jumps; weighs sale of company.</title>
<headline>Tylan stock jumps; weighs sale of company.</headline>
<dateline>SAN DIEGO</dateline>
<text>
The stock of Tylan General Inc. jumped Tuesday after the maker of
process-management equipment said it is exploring the sale of the company and
added that it has already received some inquiries from potential buyers.
Tylan was up $2.50 to $12.75 in early trading on the Nasdaq market.
The company said it has set up a committee of directors to oversee the sale and
that Goldman, Sachs & amp; Co. has been retained as its financial adviser.
</text>
<metadata>
<codes class="bip:topics:1.0">
<code code="C15"/>
<code code="C152"/>
<code code="C18"/>
<code code="C181"/>
```

</codes>

<code code="CCAT"/>

Text Categorization: The Categories

In total 103 topic categories:

C11	STRATEGY/PLANS	C15	PERFORMANCE
C12	LEGAL/JUDICIAL	C151	ACCOUNTS/EARNINGS
C13	REGULATION/POLICY	C1511	ANNUAL RESULTS
C14	SHARE LISTINGS	C152	COMMENT/FORECASTS
C15	PERFORMANCE	C16	INSOLVENCY/LIQUIDITY
C151	ACCOUNTS/EARNINGS	C17	FUNDING/CAPITAL
C1511	ANNUAL RESULTS	C171	SHARE CAPITAL
C152	COMMENT/FORECASTS	C172	BONDS/DEBT ISSUES
C16	INSOLVENCY/LIQUIDITY	C173	LOANS/CREDITS
C17	FUNDING/CAPITAL	C174	CREDIT RATINGS
C171	SHARE CAPITAL	C18	OWNERSHIP CHANGES
C172	BONDS/DEBT ISSUES	C181	MERGERS/ACQUISITIONS
C173	LOANS/CREDITS	C182	ASSET TRANSFERS
C174	CREDIT RATINGS	C183	PRIVATISATIONS
C11	STRATEGY/PLANS	C21	PRODUCTION/SERVICES
C12	LEGAL/JUDICIAL	C22	NEW PRODUCTS/SERVICES
C13	REGULATION/POLICY	C23	RESEARCH/DEVELOPMENT
C14	SHARE LISTINGS		

Word Categorization: The Parts of Speech

Sentence:

That round table might collapse

Annotation:

Words	Parts of speech	POS tags
that	Determiner	DT
round	Adjective	JJ
table	Noun	NN
might	Modal verb	MD
collapse	Verb	VB

The automatic annotation uses predefined POS tagsets such as the Penn Treebank tagset for English

Ambiguity

Words	Possible tags	Example of use		
that	Subordinating conjunction	That he can swim is good		
	Determiner	That white table		
	Adverb	It is not that easy		
	Pronoun	That is the table		
	Relative pronoun	The table that collapsed		
round	Verb	Round up the usual suspects		
	Preposition	Turn round the corner		
	Noun	A big round		
	Adjective	A round box		
	Adverb	He went round		
table	Noun	That white table		
	Verb	I table that		
might	Noun	The might of the wind		
	Modal verb	She might come		
collapse	Noun	The collapse of the empire		
	Verb	The empire can collapse		

Segment Recognition

```
Group detection – chunking –:
    Brackets: [NG] The government NG has [NG]
              other agencies and instruments NG for pursuing NG
              these other objectives NG .
       Tags: The/I government/I has/O other/I agencies/I and/I
             instruments/I for/O pursuing/O these/I other/I objectives/I ./O
    Brackets: Even [NG] Mao Tse-tung NG [NG] 's China NG began in [NG]
              1949 _{NG} with [_{NG} a partnership _{NG} between [_{NG}
              the communists N_G and N_G a number N_G of N_G
              smaller, non-communists parties NG.
```

Tags: Even/O Mao/I Tse-tung/I 's/B China/I began/O in/O 1949/I with/O a/I partnership/I between/O the/I communists/I and/O a/I number/I of/O smaller/I ,/I non-communists/I parties/I ./O

Segment Categorization

Tages extendible to any type of chunks: nominal, verbal, etc.

For the IOB scheme, this means tags such as I.Type, O.Type, and B.Type, Types being NG, VG, PG, etc.

In CoNLL 2000, ten types of chunks

Word	POS	Group	Word	POS	Group
Не	PRP	B-NP	to	TO	B-PP
reckons	VBZ	B-VP	only	RB	B-NP
the	DT	B-NP	£	#	I-NP
current	JJ	I-NP	1.8	CD	I-NP
account	NN	I-NP	billion	CD	I-NP
deficit	NN	I-NP	in	IN	B-PP
will	MD	B-VP	September	NNP	B-NP
narrow	VB	I-VP	•		O

Noun groups (NP) are in red and verb groups (VP) are in blue.



IOB Annotation for Named Entities

	NLL 2002	CoNLL 2003					
Words	Named entities	Words	POS	Groups	Named entities		
Wolff	B-PER	U.N.	NNP	I-NP	I-ORG		
,	0	official	NN	I-NP	0		
currently	0	Ekeus	NNP	I-NP	I-PER		
a	0	heads	VBZ	I-VP	0		
journalist	0	for	IN	I-PP	0		
in	0	Baghdad	NNP	I-NP	I-LOC		
Argentina	B-LOC			0	0		
,	0						
played	0						
with	0						
Del	B-PER						
Bosque	I-PER						
in	0						
the	0						
final	0						
years	0						
of	0						
the	0						
seventies	0						
in	0						
Real	B-ORG						
Madrid	I-ORG						
	0						

Evaluation

Accuracy, precision, and recall.

For noun groups with the predicted output:

Word	POS	Group	Word	POS	Group
Не	PRP	B-NP	to	TO	B-PP
reckons	VBZ	B-VP	only	RB	B-NP
the	DT	B-NP	£	#	I-NP
current	JJ	B-NP	1.8	CD	B-NP
account	NN	I-NP	billion	CD	I-NP
deficit	NN	I-NP	in	IN	B-PP
will	MD	B-VP	September	NNP	B-NP
narrow	VB	I-VP	•		0

Accuracy =
$$\frac{14}{16}$$
, recall = $\frac{2}{4}$ = 0.5, precision = $\frac{2}{6}$ = 0.33 harmonic mean = $2 \times \frac{0.33 \times 0.5}{0.33 + 0.5}$ = 0.4

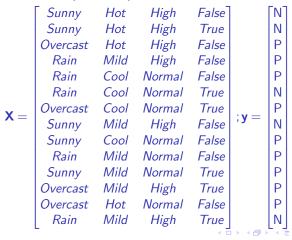


Encoding Words

Neural networks can only handle numbers
We need then to encode the words or the characters with numbers.
Using ordinal numbers (a:1, b:2, c:3, d:4, etc) is impossible.
Is a closer to b, than c?
The most simple encoding is the one-hot encoding (or contrast encoding), that we have seen in the 3rd lecture.

Matrix Notation

- A feature vector (predictors): x, and feature matrix: X;
- The class: y and the class vector: y;
- The predicted class (response): \hat{y} , and predicted class vector: \hat{y}



Converting Symbolic Attributes into Numerical Vectors

Linear classifiers are numerical systems.

Symbolic – nominal – attributes are mapped onto vectors of binary values.

This is called a one-hot encoding

A conversion of the weather data set.

Object	Attributes								Class		
		Outlook		T	Temperature			Humidity		Windy	
	Sunny	Overcast	Rain	Hot	Mild	Cool	High	Normal	True	False	
1	1	0	0	1	0	0	1	0	0	1	N
2	1	0	0	1	0	0	1	0	1	0	N
3	0	1	0	1	0	0	1	0	0	1	P
4	0	0	1	0	1	0	1	0	0	1	P
5	0	0	1	0	0	1	0	1	0	1	P
6	0	0	1	0	0	1	0	1	1	0	N
7	0	1	0	0	0	1	0	1	1	0	P
8	1	0	0	0	1	0	1	0	0	1	N
9	1	0	0	0	0	1	0	1	0	1	P
10	0	0	1	0	1	0	0	1	0	1	P
11	1	0	0	0	1	0	0	1	1	0	P
12	0	1	0	0	1	0	1	0	1	0	P
13	0	1	0	1	0	0	0	1	0	1	P
14	0	0	1	0	1	0	1	0	1	0	N

Code Example

```
Jupyter Notebook: Chollet 6.1 https:
//github.com/fchollet/deep-learning-with-python-notebooks
```

Dimension Reduction

One-hot encoding can produce very long vectors:

Imagine a vocabulary one one million words per language with 100 languages.

A solution is to produce dense vectors also called word embeddings using a dimension reduction

This reduction is very close to principal component analysis or singular value decomposition

It can be automatically obtained through training or initialized with pretrained vectors

Principal Component Analysis

We will use a small dataset to explain principal component analysis: The characters in *Salammbô*

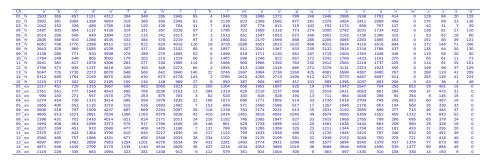


Table: Character counts per chapter, where the fr and en suffixes designate the language, either French or English

Each chapter is modeled by a vector of characters.

Character Counts

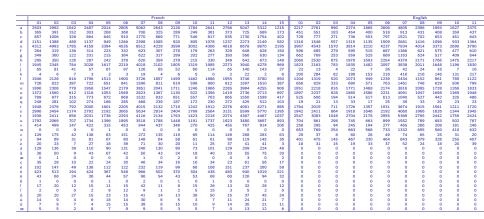


Table: Character counts per chapter in French, left part, and English, right part

Each characters is modeled by a vector of chapters.

Singular Value Decomposition

There are as many as 40 characters: the 26 unaccented letters from a to z and the 14 French accented letters

Singular value decomposition (SVD) reduces these dimensions, while keeping the resulting vectors semantically close

X is the $m \times n$ matrix of the letter counts per chapter, in our case, m = 30 and n = 40.

We can rewrite X as:

$$X = U\Sigma V^{\mathsf{T}},$$

where **U** is a matrix of dimensions $m \times m$, Σ , a diagonal matrix of dimensions $m \times n$, and **V**, a matrix of dimensions $n \times n$.

The diagonal terms of Σ are called the **singular values** and are traditionally arranged by decreasing value.

We keep the highest values and set the rest to zero.



Code Example

Jupyter Notebook 3.1-SVD



Word Embeddings

We can extend singular value decomposition from characters to words.

The rows will represent the words in the corpus, and the columns, documents,

We can replace documents by a context of a few words to the left and to the right of the focus word: w_i .

A context C_j is then defined by a window of 2K words centered on the word:

$$W_{i-K}, W_{i-K+1}, ..., W_{i-1}, W_{i+1}, ..., W_{i+K-1}, W_{i+K},$$

where the context representation uses a bag of words.

We can even reduce the context to a single word to the left or to the right of w_i and use bigrams.

Word Embeddings

We store the word-context pairs (w_i, C_j) in a matrix.

Each matrix element measures the association strength between word w_i and context C_j , for instance mutual information.

D#\Words	C_1	C_2	C ₃	 C _n
w_1	$MI(w_1, C_1)$	$MI(w_1, C_2)$	$MI(w_1, C_3)$	 $MI(w_1, C_n)$
W ₂	$MI(w_2, C_1)$	$MI(w_2,C_2)$	$MI(w_2, C_3)$	 $MI(w_2, C_n)$
w ₃	$MI(w_3, C_1)$	$MI(w_3, C_2)$	$MI(w_3, C_3)$	 $MI(w_3, C_n)$
W _m	$MI(w_m, C_1)$	$MI(w_m, C_2)$	$MI(w_m, C_3)$	 $MI(w_m, C_n)$

Word Embeddings

We compute the word embeddings with a singular value decomposition, where we truncate the matrix $U\Sigma$ to 50, 100, 300, or 500 dimensions. The word embeddings are the rows of this matrix.

We usually measure the similarity between two embeddings \vec{u} and \vec{v} with the cosine similarity:

$$\cos(\vec{u}, \vec{v}) = \frac{\vec{u} \cdot \vec{v}}{||\vec{u}|| \cdot ||\vec{v}||},$$

ranging from -1 (most dissimilar) to 1 (most similar) or with the cosine distance ranging from 0 (closest) to 2 (most distant):

$$1 - \cos(\vec{u}, \vec{v}) = 1 - \frac{\vec{u} \cdot \vec{v}}{||\vec{u}|| \cdot ||\vec{v}||}.$$



Popular Word Embeddings

Embeddings from large corpora are obtained with iterative techniques Popular embedding algorithms with open source programs are:

```
word2vec: https://github.com/tmikolov/word2vec
```

GloVe: Global Vectors for Word Representation

https://nlp.stanford.edu/projects/glove/

ELMo: https://allennlp.org/elmo

fastText: https://fasttext.cc/

To derive word embeddings, you will have to apply these programs on a very large corpus

Embeddings for many languages are also publicly available. You just download them

gensim is a Python library to create word embeddings from a corpus.

https://radimrehurek.com/gensim/index.html



Semantic Similarity

Word embeddings mitigate the dimension problem relatively to one-hot encoding

The also cope with unseen words in a training set as similar words will have similar vectors

Demo: http://bionlp-www.utu.fi/wv_demo/

Code Example

```
Jupyter Notebook: Chollet 5.3 https:
//github.com/fchollet/deep-learning-with-python-notebooks
```