Számítógépes szemantika

Berend Gábor

Értékelés

- A tárgy két részből áll, vizsga sikeres gyakorlat után tehető
- Gyakorlati jegy: projektmunka + aktivitás
- Előadás: szakirodalom-feldolgozás (és előadás)

Lehetséges projektmunkák

- Megegyező tartalmú kérdések felismerése
  - https://www.kaggle.com/c/quora-question-pairs
- Etikusság
  - https://blog.conceptnet.io/2017/04/24/conceptnet-numberbatch-17-04-better-less-stereotyped-word-vectors/
- Fedőnevek ágens
  - https://jamesmullenbach.github.io/2018/01/02/codenames-fun.html
- Saját témák

Mikről lesz szó a tárgyon?

- Szavak jelentését megragadni (közelíteni) képes számítógépes reprezentációk létrehozása
- Mire használhatók ezek a reprezentációk?
- Hogyan értékelhetők ki ezek a reprezentációk?
- Többnyelvű reprezentációk?
- Python + matek (valószínűség és lineáris algebra)
  - Numpy, scipy, sklearn csomagok

Python használata

- Python 2 vs. **Python 3**
  - https://docs.python.org/3/tutorial/
- Virtualenv vs. **anaconda**
  - https://www.continuum.io/downloads
- Nem csak Python
  - pip, ipython, jupyter, ...

Python alapok

- OO nyelv, dinamikus típusokkal
- `-type(valtozonev)` utasítással megtudhatjuk a változó aktuális típusát
- Nem muszáj ;-vel zárni a sorokat
- Cserébe az indentálásra ügyelni kell (szóközök)
- Asszociatív tömbök használatának támogatása

Python alapok – Fontos tárolók

- tuple
  - t = (“Sanyi”, 22, “körte”)
  - Tipikusan eltérő “szemantikájú” adatokat csomagolunk össze segítségével
  - Inmutable (t[1]=23 hibát eredményezne)
- list
  - l = [5,3,1,4]
  - Tipikusan azonos típusú adatok összefűzésére használjuk
  - Mutable (l[1]=4 nem okoz hibát)

Python alapok – Fontos tárolók

- dict
  - d = {“Sanyi”:22, “Manyi”:33, “Enci”:52}
  - Asszociatív tömböt (kulcs-érték párok) valósít meg
  - KeyError, ha nem létező kulcsra hivatkozunk (pl. d[‘Éva’])
- set
  - Halmazt valósít meg (tulajdonképp egy olyan dict, ami nem rendel értékeket a kucsokhoz)
- Továbbiak: defaultdict, Counter, …
  - A `collections` csomagból importálhatjuk be őket

Python vezérlési szerkezetek

- Ismétléses vezérlés

```
for valtozo in range(20):
utasitasok
```

- Feltételes vezérlés (&&→ and, ||→ or)

```
if feltetel:
utasitas(ok)
elif feltetel:
utasitas(ok)
else:
utasitas(ok)
```

Python I/O

- Plain text fájlbeolvasás

```
for line in open('be.txt', 'r', encoding='utf-
8'):
    print(line)
```

- Plain text kiíratás

```
f = open('workfile', 'w')
f.write('Dummy content...')
f.close()
```

- Alternatív megoldás (with statement használatával)

```
with open("x.txt") as f:
```

Python függvények

```
def fuggveny_nev(parameterek):
    fuggvenytorzs
    [return valtozo(k)]
```

- Például

```
def hello_bello(hello)
if hello:
    print('Hello')
else:
    print('Bello')
```

Importálás lehetőségei

- Külső modulok funkcionalitásának használatához

```
import modulnev
import modulnev as alias
from modulnev import metodus
```

- A külső modulnak persze már telepítve kell legyen

```
-pip install[ --upgrade] modulnev
```

Numpy vs. Matlab

- Sok hasonlóság mellett fontos különbségek is

	Numpy	Matlab
Transzponálás	Y=X.T	Y=X’
Mátrixszorzás	C=A.dot(B)	C=A*B
Elemenkénti szorzás	D=A*B	D=A.*B

- Mátrixszorzás esetén $A \in \mathbb{R}^{kxl}, B \in \mathbb{R}^{lxm}$
- Elemenkénti szorzás esetében $A \in \mathbb{R}^{kxl}, B \in \mathbb{R}^{kxl}$

Gyakorlás

- Írjunk egy függvényt, ami kiszámolja az n-edik Fibonacci számot
- Hozzunk létre egy listát, ami a Fibonacci sor első 20 elemét tartalmazza
- Érjük el, hogy az előző lista kétszer egymás után legyen fűzve
- Az előző lista minden elemét emeljük négyzetre
  - List comprehension

Szemantika (jelentéstan)

- Disztribúciós hipotézis
  - Hasonló jelentésű szavak környezetében hasonló szavak találhatók
- Az elmélet >50 éves, az első számítógépes megvalósítás is már >20
  - Igazán népszerű csak az elmúlt években lett

Folytonos reprezentáció

alma [1 0 0 0 … 0 0 0 0 0 … 0] ⟶ [3,2 -1,5]

…

körte [0 0 0 0 … 1 0 0 0 0 … 0] ⟶ [2,8 -1,6]

…

lapát [0 0 0 0 … 0 0 1 0 0 … 0] ⟶ [-4,1 12,6]

…

zebra [0 0 0 0 … 0 0 0 0 0 … 1] ⟶ [3,8 0,5]

Versengő paradigmák

- 2013: word2vec megjelenése
  - Korábbi években is voltak már NN-alapú modellek

Versengő paradigmák

- 2013: word2vec megjelenése
  - Korábbi években is voltak már NN-alapú modellek

***Don't count, predict!* A systematic comparison of context-counting vs. context-predicting semantic vectors**

Marco Baroni and **Georgiana Dinu** and **Germán Kruszewski**

Versengő paradigmák

- 2013: word2vec megjelenése
  - Korábbi években is voltak már NN-alapú modellek

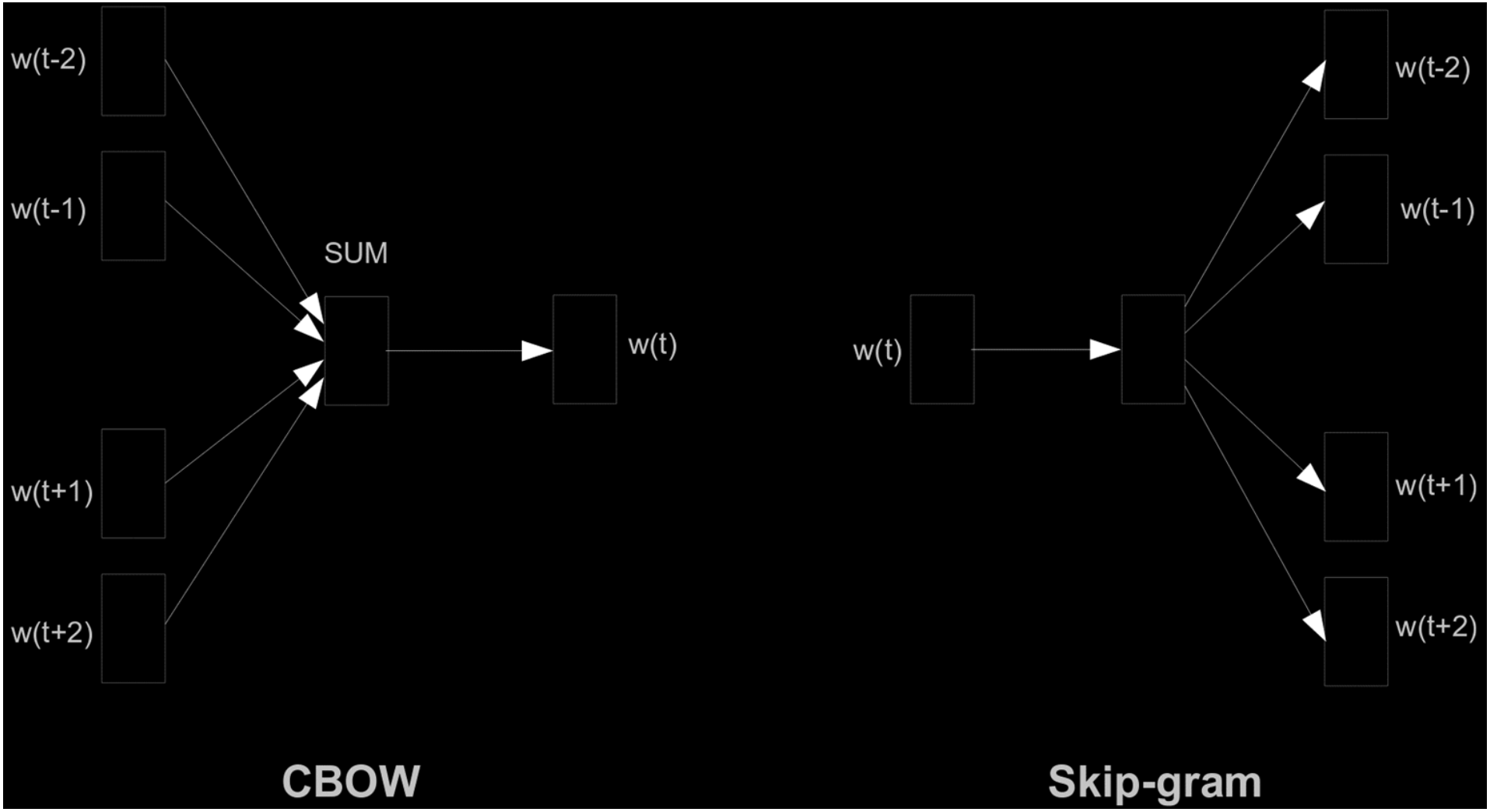
***Don't count, predict!* A systematic comparison of context-counting vs. context-predicting semantic vectors**

Marco Baroni and **Georgiana Dinu** and **Germán Kruszewski**

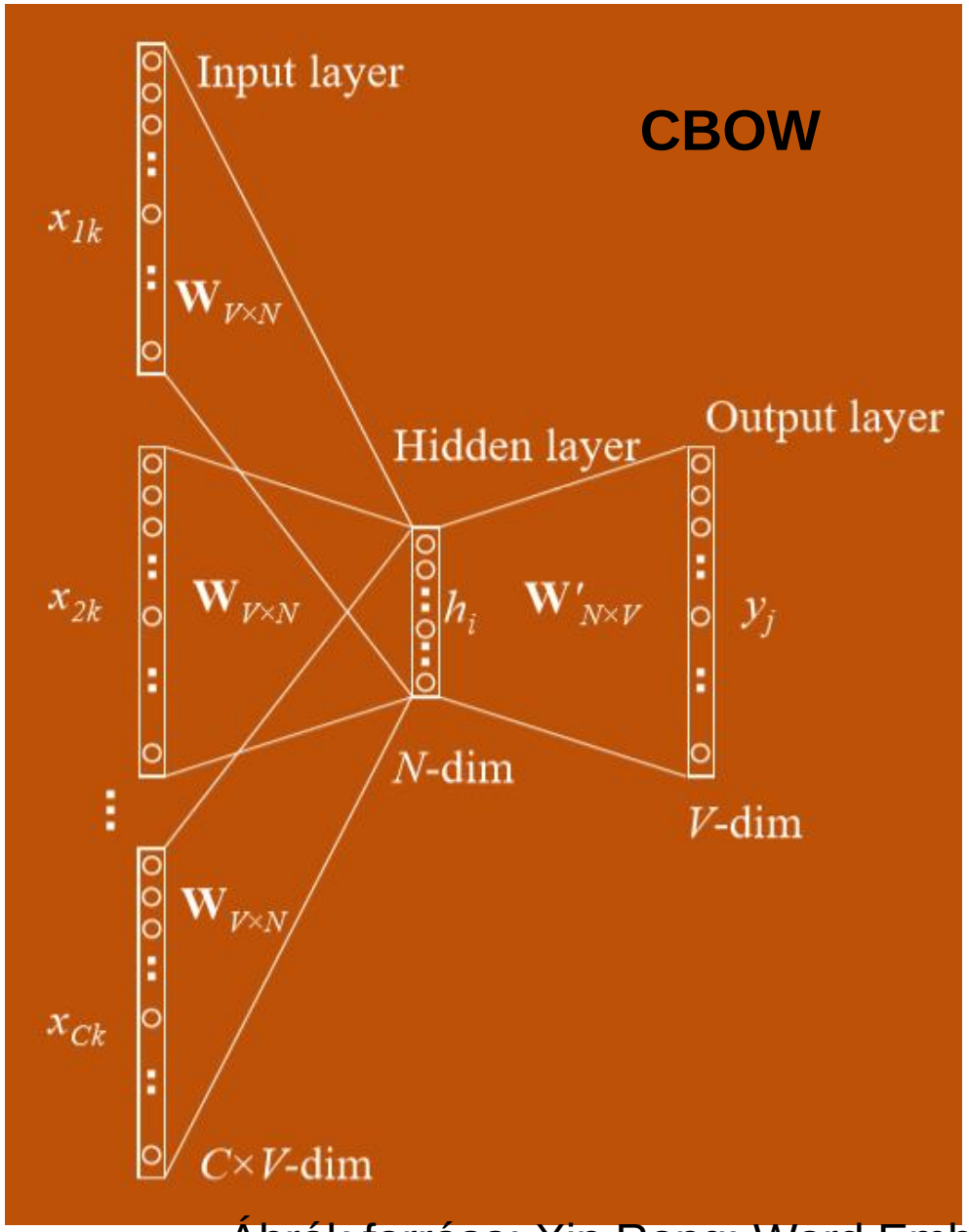
Rehabilitation of Count-based Models for Word Vector Representations

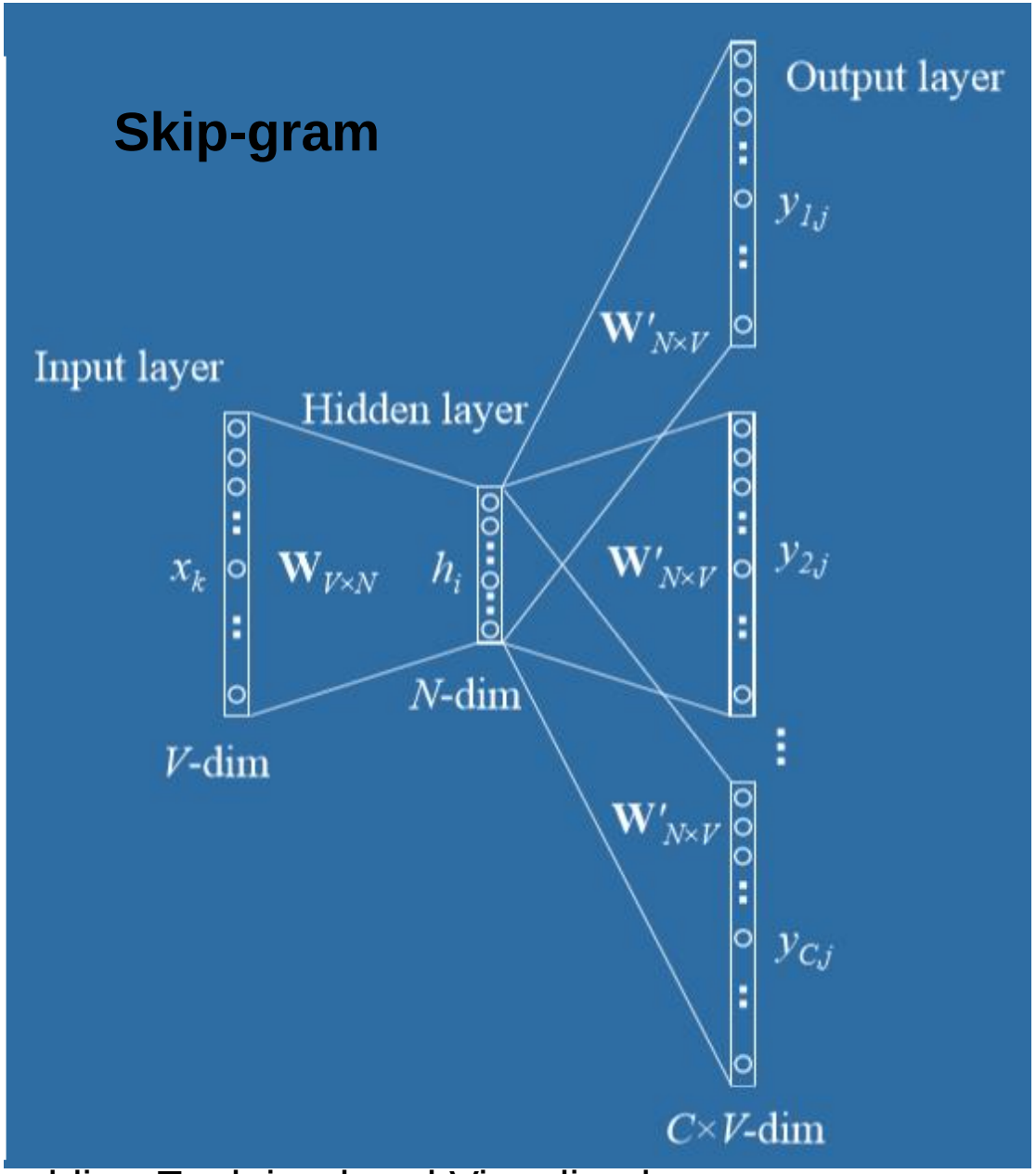
Rémi Lebret[1,2] and **Ronan Collobert**[1]

word2vec sematikusan



word2vec neurális hálós értelmezése





Ábrák forrása: Xin Rong: Word Embedding Explained and Visualized

word2vec célja

- Hasonló jelentésű input szavak hasonló outputot eredményezzenek

$$y(x) = softmax\left(W'(W1_x)\right)$$

- a és b szó jelentése minél hasonlóbb, y(a) és y(b) (eloszlás)vektorok annál inkább hasonlítani fognak
- CBOW: x “környező” szavak reprezentációi alapján akarjuk a “középső” szót előrejelezni y(x)-szel
- Skipgram: x “középső” szó reprezentációja alapján akarjuk a “környező” szavakat előrejelezni y(x)-szel

RepEval 2016

Analysis Track

- **Problems With Evaluation of Word Embeddings Using Word Similarity Tasks** [pdf]
  Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, Chris Dyer

- **Intrinsic Evaluations of Word Embeddings: What Can We Do Better?** [pdf]
  Anna Gladkova, Aleksandr Drozd

- **Issues in Evaluating Semantic Spaces Using Word Analogies** [pdf]
  Tal Linzen

- **Intrinsic Evaluation of Word Vectors Fails to Predict Extrinsic Performance** [pdf]
  Billy Chiu, Anna Korhonen, Sampo Pyysalo

- **A Critique of Word Similarity as a Method for Evaluating Distributional Semantic Models** [pdf]
  Miroslav Batchkarov, Thomas Kober, Jeremy Reffin, Julie Weeds, David Weir

Folytonos reprezentáció

alma [1 0 0 0 … 0 0 0 0 0 … 0] ⟶ [3,2 -1,5]

…

körte [0 0 0 0 … 1 0 0 0 0 … 0] ⟶ [2,8 -1,6]

…

lapát [0 0 0 0 … 0 0 1 0 0 … 0] ⟶ [-4,1 12,6]

…

zebra [0 0 0 0 … 0 0 0 0 0 … 1] ⟶ [3,8 0,5]

Ritka folytonos reprezentáció

alma [3,2 -1,5] ⟶ [0 1,7 0 0 -0,2 0]

...

körte [2,8 -1,6] ⟶ [0 1,1 0 0 -0,4 0]

...

lapát [-4,1 12,6] ⟶ [1,7 0 -2,1 0 0 -0,8]

...

zebra [3,8 0,5] ⟶ [0 0 1,3 0 -1,2 0]