Търсене и извличане на информация. Приложение на дълбоко машинно обучение

Стоян Михов

Лекция 10: Влагане на думи с невронни мрежи. Невронен езиков модел.

План на лекцията

1. **Формалности за курса (5 мин)**
2. Преглед на използването на влагане на думи за класификация на документи (15 мин)
3. Невронен езиков модел на Бенджио и съавтори (15 мин)
4. Моделът Word2Vec CBOW (20 мин)
5. Моделът Word2Vec skip-gram negative-sampling (20 мин)
6. Оценяване на влагане на думи и невронни езикови модели (15 мин)

Формалности

- Засега ще провеждаме занятията онлайн всяка сряда от 8:15 до 12:00 часа.
- Засега ще използваме платформата Google meet: meet.google.com/hue-frfx-axb
- Днес ще използваме едновременно слайдове и бяла дъска. Моля следете съответния екран.
- Второто домашното задание ще бъде обявено преди празниците.
- Десетата лекция се базира на глави 10 и 11 от втория учебник.

План на лекцията

1. Формалности за курса (5 мин)

2. **Преглед на използването на влагане на думи за класификация на документи (15 мин)**

3. Невронен езиков модел на Бенджио и съавтори (15 мин)

4. Моделът Word2Vec CBOW (20 мин)

5. Моделът Word2Vec skip-gram negative-sampling (20 мин)

6. Оценяване на влагане на думи и невронни езикови модели (15 мин)

От миналите лекции

- В лекция 8 разгледахме логистична регресия за класифициране на документи.

- Вероятността за документ представен с документен вектор $\mathbf{x}$ да бъде от клас $y = c$ моделирахме:

$$\Pr_{W,\mathbf{b}}[y = c \,|\, \mathbf{x}] = \text{softmax}(W\mathbf{x} + \mathbf{b})_c$$

- Този подход ни даде значително подобрение на резултатите спрямо наивния Бейсов класификатор.

- Всъщност, подобрението се дължи в голяма степен на представянето на документите в гъсто векторно пространство.

- Как получихме документните вектори?

От миналите лекции

- В лекция 6 разгледахме влагане на термове в гъсто, нискомерно векторно пространство, чрез използване на принципен компонентен анализ.

- Матрицата на влагането (Embedding Matrix) означаваме $E \in \mathbb{R}^{M \times |V|}$, където M е размернстта на латентното семантично векторно пространство, L е речника на термовете, а $|L|$ е броят на думите в речника,

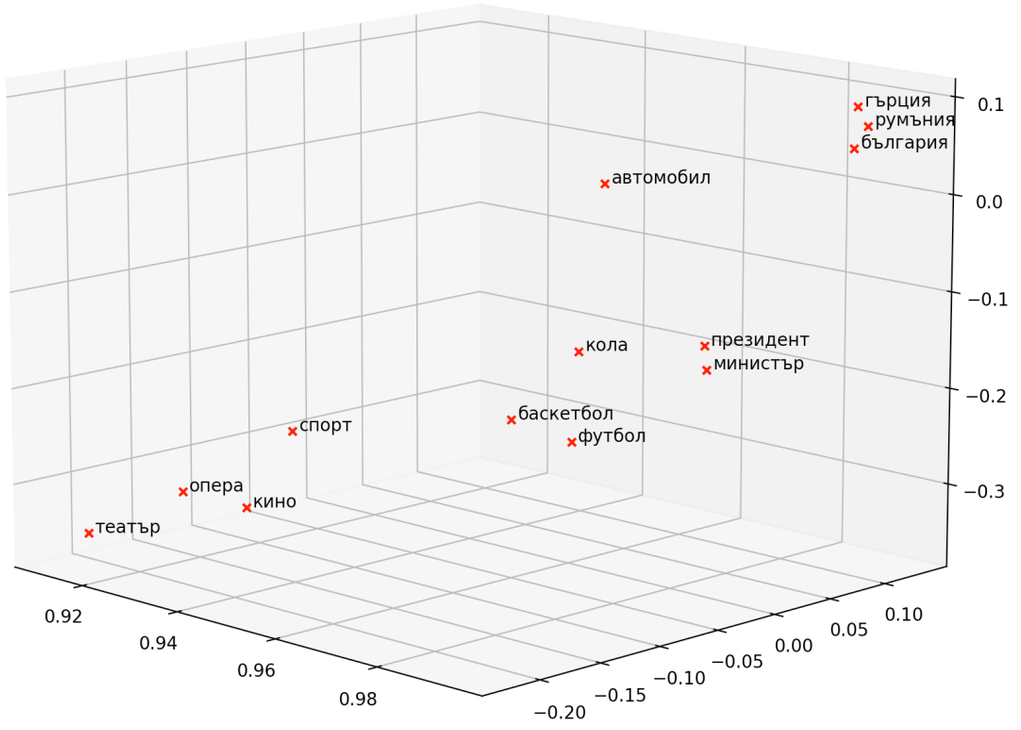
- Ако документа d се състои от термовете $t_1, t_2, \ldots, t_{|d|}$, а one-hot вектора за терма t означаваме с $\chi_t \in \mathbb{R}^{|L|}$, то влагането CBOW (Continious Bag of Words) дефинираме като
$\mathbf{x} = \text{CBOW}(d) = \text{norm}(\sum_{t_i \in d} E\chi_{t_i}) = \text{norm}(E\sum_{t_i \in d} \chi_{t_i})$, където
$\text{norm} : \mathbb{R}^M \to \mathbb{R}^M$ е нормиране на вектори: $\text{norm}(\mathbf{u}) = \frac{1}{\|\mathbf{u}\|}\mathbf{u}$.

От лекции 5 и 6

- **Дистрибутивна семантика**: Значението на дадена дума се определя от думите, които често се срещат около нея.

- Матрица на съвместните срещания

	Иван	Мария	кара	купи	обича	кола	колело
Иван	0	0	1	1	2	0	0
Мария	0	0	1	1	2	0	0
кара	1	1	0	0	0	1	1
купи	1	1	0	0	0	1	1
обича	2	2	0	0	0	1	1
кола	0	0	1	1	1	0	0
колело	0	0	1	1	1	0	0



- Близост или подобие между термовете t_i, t_k дефинираме:

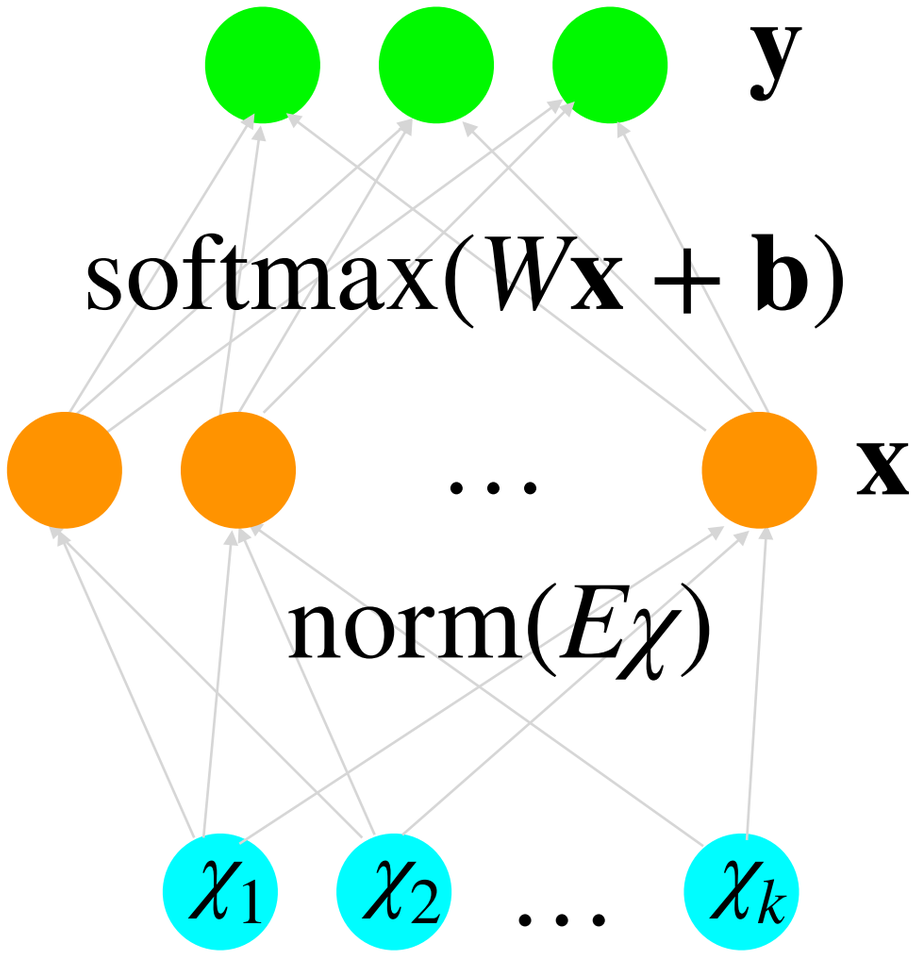
$$\text{sim}_{\cos}(t_i, t_k) = \cos(E_{\bullet,i}, E_{\bullet,k}) = \frac{E_{\bullet,i} \cdot E_{\bullet,k}}{|E_{\bullet,i}||E_{\bullet,k}|}$$

От миналите лекции

- Да разгледаме пълната задача като невронна мрежа с два слоя:

$$\mathbf{y} = \text{softmax}(W\mathbf{x} + \mathbf{b})$$

$$\mathbf{x} = \text{norm}(E \sum_{t_i \in d} \chi_{t_i})$$



- В миналите лекции първо научихме влагането E чрез принципен компонентен анализ на матрицата на съвместни срещания относно поточкова взаимна информация. След това тренирахме $W, \mathbf{b}$, чрез минимизиране на кросентропията със спускане по градиента.

- Може ли директно да трениграме пълния модел, като едновременно трениграме $E, W, \mathbf{b}$? Имаме ли достатъчно данни?

Предварително натренирано влагане на думи

- Проблем: много често за конкретната задача — в случая класификация на документи — нямаме достатъчно анотирани данни.

- Но може да предполагаме, че разполагаме с почти неограничени количества неанотирани данни.

- Затова е целесъобразно да тренираме влагането предварително с повече данни, така че да научим правилно семантичните връзки между думите.

- На втори етап ще тренираме горния слой на мрежата. На този етап може евентуално да дотренираме и предварителното влагане.

- Как да научим влагането от неанотирани текстове?

План на лекцията

1. Формалности за курса (5 мин)
2. Преглед на използването на влагане на думи за класификация на документи (15 мин)
3. **Невронен езиков модел на Бенджио и съавтори (15 мин)**
4. Моделът Word2Vec CBOW (20 мин)
5. Моделът Word2Vec skip-gram negative-sampling (20 мин)
6. Оценяване на влагане на думи и невронни езикови модели (15 мин)

Да научим близостта на думите от езиков модел

- В лекция 5 дефинирахме езиков модел като $\Pr[w \mid w_1 w_2 \ldots w_{n-1}]$ за всяко $w \in L$. В по-общ вариант може да дефинираме езиков модел като $\Pr[w \mid \mathbf{c}]$, където $\mathbf{c} \in L^*$ е списък от думи. Например при Марковските езикови модели от ред k за думата w_n конкеста е $\mathbf{c} = w_{n-k+1} w_{n-k+2} \cdots w_{n-1}$.

- Невронен езиков модел от статията *Yoshua Bengio et al., A neural probabilistic language model. Journal of Machine Learning Research, 3:1137–1155, March 2003.*

$$\mathbf{y} = \text{softmax}(W^{(2)}\mathbf{h} + \mathbf{b}^{(2)})$$

$$\mathbf{h} = g(W^{(1)}\mathbf{x} + \mathbf{b}^{(1)})$$

$$\mathbf{x} = \begin{bmatrix} E\chi_{w_1} \\ \vdots \\ E\chi_{w_k} \end{bmatrix}$$, където

$\chi_{w_i} \in \mathbb{R}^{|L|}, E \in \mathbb{R}^{M \times |L|}, \mathbf{x} \in \mathbb{R}^{kM}, W^{(1)} \in \mathbb{R}^{N \times kM}, \mathbf{b}^{(1)} \in \mathbb{R}^N, \mathbf{h} \in \mathbb{R}^N,$
$W^{(2)} \in \mathbb{R}^{|L| \times N}, \mathbf{b}^{(2)} \in \mathbb{R}^{|L|}, \mathbf{y} \in \mathbb{R}^{|L|}$

- В модела на Бенджио влаганията на думите от контекста се конкатенират за получаването на входния вектор $\mathbf{x} \in \mathbb{R}^{kM}$.

- В междинния слой, чрез линеен перцептрон се получава скрит вектор $\mathbf{h} \in \mathbb{R}^{N}$, който отразява контекста.

- В последния слой, контекстния вектор $\mathbf{h}$ се преобразува през втори перцептрон и софтмакс, за да се получи вероятностно разпределение за следващата дума.

- В този модел матриците $E \in \mathbb{R}^{M \times |L|}$ и $W^{(2)} \in \mathbb{R}^{|L| \times N}$ са влагания на думи.

- В някои варианти се предполага, че $M = N$ и $E^{\top} = W^{(2)}$.

- Моделът може да се обучи чрез минимизиране на кросентропията със спускане по градиента от корпус.

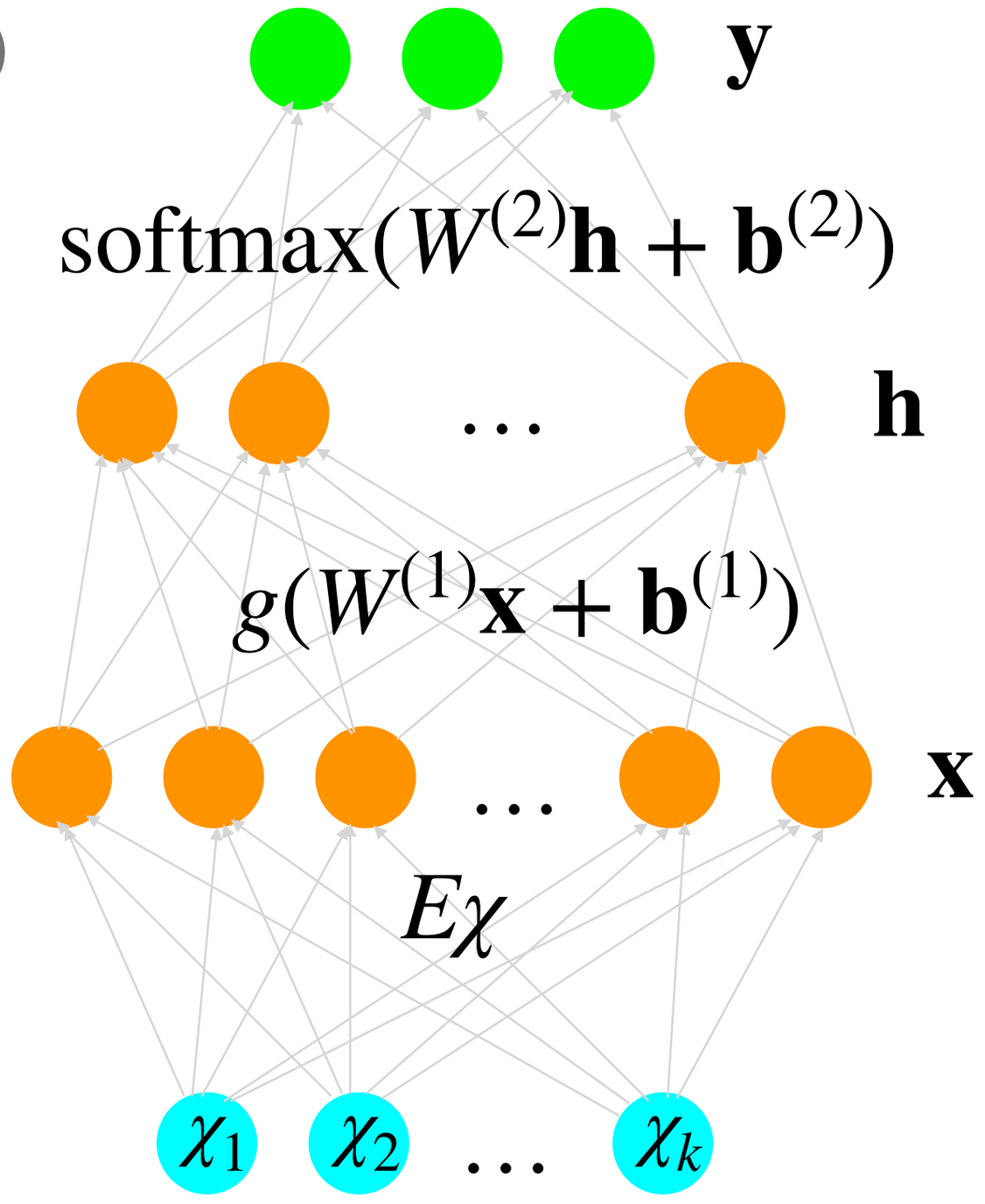
- **Проблем**: Този модел е сравнително сложен.

Невронен езиков модел на Bengio et al.

$$\mathbf{y} = \text{softmax}(W^{(2)}\mathbf{h} + \mathbf{b}^{(2)})$$

$$\mathbf{h} = g(W^{(1)}\mathbf{x} + \mathbf{b}^{(1)})$$

$$\mathbf{x} = \begin{bmatrix} E\chi_{w_1} \\ \vdots \\ E\chi_{w_k} \end{bmatrix}$$



План на лекцията

1. Формалности за курса (5 мин)
2. Преглед на използването на влагане на думи за класификация на документи (15 мин)
3. Невронен езиков модел на Бенджио и съавтори (15 мин)
4. **Моделът Word2Vec CBOW (20 мин)**
5. Моделът Word2Vec skip-gram negative-sampling (20 мин)
6. Оценяване на влагане на думи и невронни езикови модели (15 мин)

По-ефективни методи за научаване на влагане

- Обучението може да извършваме като минимизираме кросентропията $H_{\mathbf{X}} = -\frac{1}{|\mathbf{X}|}\sum_{w \in \mathbf{X}} \log \Pr[w \mid \mathbf{c}_w]$, като за вероятността $\Pr[w \mid \mathbf{c}]$ ще използваме по-прост модел.

- Ако се интересуваме само от влагането на думите то разпределението $\Pr[w \mid \mathbf{c}]$ не ни е нужно експлицитно.

- Миколов и съавтори разработват през 2013 няколко високо-ефективни модела за научаване на влагания на думи известни като **Word2Vec**

  - *Tomáš Mikolov. Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. arXiv:1301.3781*

  - *Tomáš Mikolov. Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems 26, pages 3111–3119, 2013.*

Моделът Word2Vec CBOW

- Ако w е n-тата дума в корпуса $\mathbf{X}$, т.е. $w = t_n$ то избираме контекста да са думите около w т.е. $\mathbf{c} = t_{n-k/2} \dots t_{n-1} t_{n+1} \dots t_{n+k/2}$.

- Нека влаганията за целевата дума са $U \in \mathbb{R}^{M \times |L|}$, а влаганията за контекстните думи са $V \in \mathbb{R}^{M \times |L|}$. Тогава влагането на w е $\mathbf{u}_w = U\chi_w = U_{\bullet,w}$, а влагането на c_i е $\mathbf{v}_{c_i} = V\chi_{c_i} = V_{\bullet,c_i}$.

- Ще използваме CBOW за моделиране на контекста: $\mathbf{v}_\mathbf{c} = \sum_{c_i \in \mathbf{c}} V\chi_{c_i} = \sum_{c_i \in \mathbf{c}} \mathbf{v}_{c_i}$

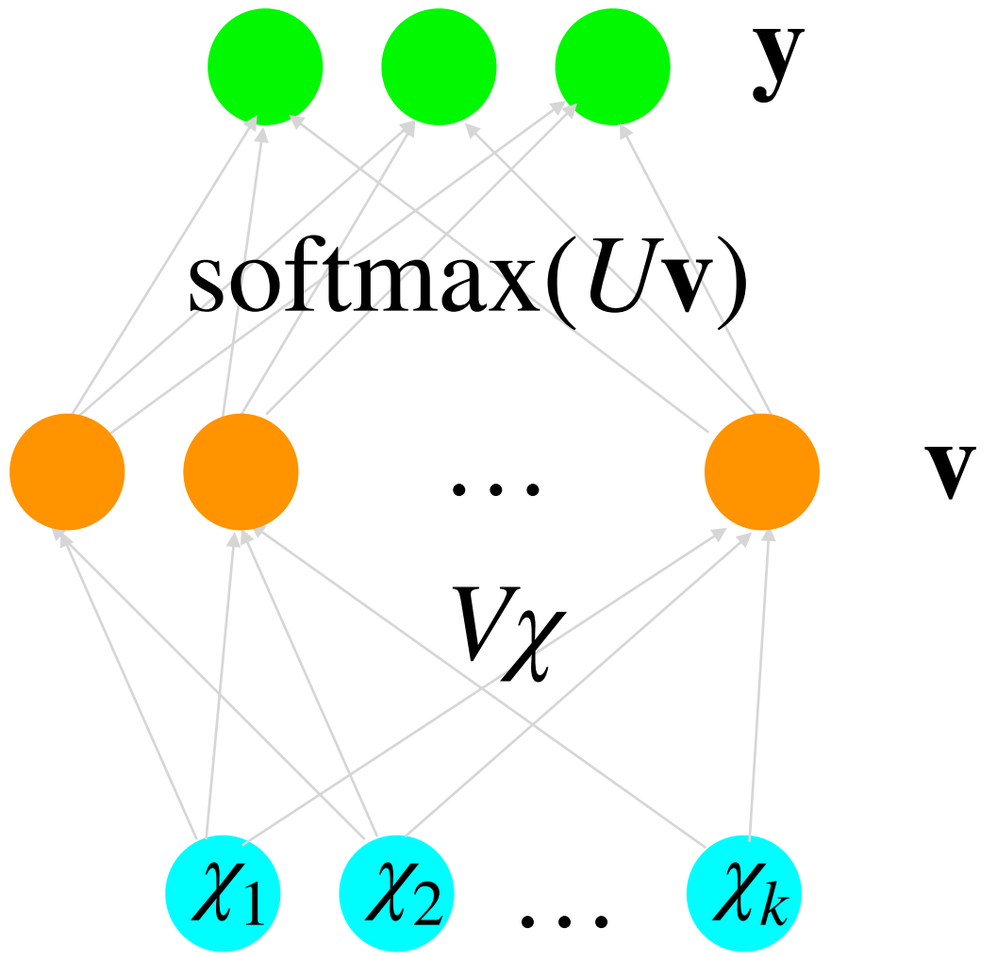
- Ще моделираме вероятността $\Pr[w \mid \mathbf{c}] = \mathrm{softmax}(U^\top \mathbf{v}_\mathbf{c})_w = \frac{e^{\mathbf{u}_w^\top \mathbf{v}_\mathbf{c}}}{\sum_{t \in V} e^{\mathbf{u}_t^\top \mathbf{v}_\mathbf{c}}}$

- Минимизираме кросентропията:
$$H_\mathbf{X}(U, V) = -\frac{1}{|\mathbf{X}|} \sum_{(w,\mathbf{c}) \in \mathbf{X}} \log \Pr[w \mid \mathbf{c}] = -\frac{1}{|\mathbf{X}|} \sum_{(w,\mathbf{c}) \in \mathbf{X}} \log \mathrm{softmax}(U^\top \mathbf{v}_\mathbf{c})_w$$

Моделът Word2Vec CBOW

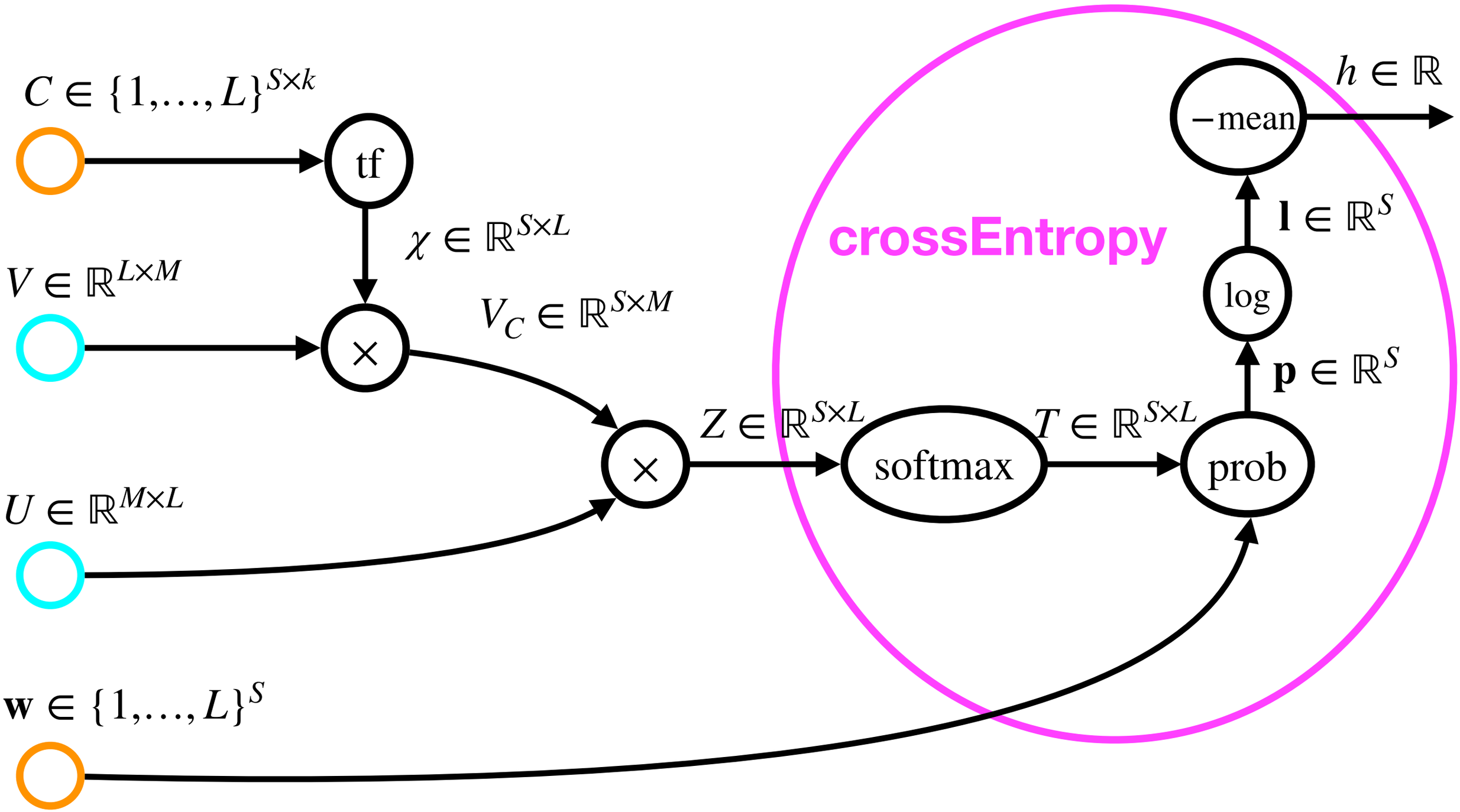
$$\mathbf{y} = \text{softmax}(U\mathbf{v})$$

$$\mathbf{v} = V \sum_{i=1}^{k} \chi_i$$



- $\frac{\partial}{\partial \mathbf{z}} \log \text{softmax}(\mathbf{z})_w = \bar{\delta}_w - \text{softmax}(\mathbf{z})$
- $\mathbf{z} = U^\top \mathbf{v_c}$
- $\frac{\partial \log \text{softmax}(U^\top \mathbf{v_c})_w}{\partial \mathbf{v}_c} = (\bar{\delta}_w - \text{softmax}(U^\top \mathbf{v_c}))U^\top$
- $\frac{\partial \log \text{softmax}(U^\top \mathbf{v_c})_w}{\partial U} = (\bar{\delta}_w - \text{softmax}(U^\top \mathbf{v_c})) \otimes \mathbf{v_c}$
- За всяко наблюдение градиента по U е гъста матрица. Следователно презаписа на параметрите за партида с големина B е пропорционална на $BMk|L|$

Векторен изчислителен граф на Word2Vec CBOW



План на лекцията

1. Формалности за курса (5 мин)

2. Преглед на използването на влагане на думи за класификация на документи (15 мин)

3. Невронен езиков модел на Бенджио и съавтори (15 мин)

4. Моделът Word2Vec CBOW (20 мин)

5. **Моделът Word2Vec skip-gram negative-sampling (20 мин)**

6. Оценяване на влагане на думи и невронни езикови модели (15 мин)

Моделът Word2Vec skip-gram negative-sampling

- Вместо $\Pr[w \mid \mathbf{c}]$ ще разглеждаме $\Pr[\mathbf{c} \mid w]$ и ще предполагаме независимост $\Pr[\mathbf{c} \mid w] = \prod_{c_i \in \mathbf{c}} \Pr[c_i \mid w]$. В статията на Миколов и съавтори този подход се нарича **skip-gram**.

- По принцип се стремим да минимизираме кросентропията:
$$H_{\mathbf{X}}(U, V) = -\frac{1}{|\mathbf{X}|} \sum_{(w,\mathbf{c}) \in \mathbf{X}} \sum_{c_i \in \mathbf{c}} \log \Pr[c_i \mid w] = -\frac{1}{|\mathbf{X}|} \sum_{(w,\mathbf{c}) \in \mathbf{X}} \sum_{c_i \in \mathbf{c}} \log \operatorname{softmax}(V^\top \mathbf{u}_w)_{c_i}$$

- За да се избегне изчисляването на $\operatorname{softmax}$, Миколов и съавтори използват т.н. **negative-sampling**. Тази техника в по-общ случай е развита в

  - *Gutmann, Michael & Hyvärinen, Aapo. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. Journal of Machine Learning Research 9. 297-304.*

- В нашия курс ще покажем само как се прилага за Word2Vec. Ще заменим намирането на разпределението за $\Pr[c_i \mid w]$ с разпределението на бинарна случайна величина D, която приема стойност 1, ако набюдаваме думата w в контекст на думата c и 0 в противен случай. Вероятностното разпределение на случайната величина D моделираме като: $\Pr[D = 1 \mid w, c_i] = \sigma(\mathbf{u}_w^\top \mathbf{v}_{c_i})$.

- Нека $\mathbf{Z}$ е множество от коректни двойки от целева дума и контекстна дума, а $\bar{\mathbf{Z}}$ е множество от некоректни двойки. Тогава ще целим да минимицираме функцията:

$$J_{\mathbf{Z},\bar{\mathbf{Z}}}(U,V) = -\left(\sum_{(w,c)\in\mathbf{Z}} \log \Pr[D=1\,|\,w,c] + \sum_{(w,c)\in\bar{\mathbf{Z}}} \log \Pr[D=0\,|\,w,c]\right) =$$

$$= -\left(\sum_{(w,c)\in\mathbf{Z}} \log \sigma(\mathbf{u}_w^\top \mathbf{v}_c) + \sum_{(w,\bar{c})\in\bar{\mathbf{Z}}} \log(1-\sigma(\mathbf{u}_w^\top \mathbf{v}_{\bar{c}}))\right) =$$

$$= -\left(\sum_{(w,c)\in\mathbf{Z}} \log \sigma(\mathbf{u}_w^\top \mathbf{v}_c) + \sum_{(w,\bar{c})\in\bar{\mathbf{Z}}} \log \sigma(-\mathbf{u}_w^\top \mathbf{v}_{\bar{c}})\right)$$

- Извадка от негативни примери $\bar{\mathbf{Z}}$ ще подберем, като за всеки положителен пример $(w,c)\in\mathbf{Z}$ избираме n отрицателни примера $(w,\bar{c}_j)$, като думите $\bar{c}_j$ за $j=1,2,\ldots,n$ избираме случайно от нашия речник, така че $\bar{c}_j \neq c$, използвайки монограмно разпределение $\Pr_1(\bar{c}) = \frac{\#(\bar{c})}{\sum_{w\in V}\#(w)}$.

- Вместо класическото монограмно разпределение, за да се повиши вероятността да се избират по-редки думи ще използваме разпределението $\Pr_{0.75}(\bar{c}) = \frac{\#(\bar{c})^{0.75}}{\sum_{w\in V}\#(w)^{0.75}}$.

- Ще минимизираме: $J_{\mathbf{X}}(U,V) = -\frac{1}{|\mathbf{X}|}\sum_{(w,\mathbf{c})\in\mathbf{X}}\left(\sum_{c_i\in\mathbf{c}}\left(\log\sigma(\mathbf{u}_w^\top\mathbf{v}_{c_i}) + \sum_{j=1}^{n}\log\sigma(-\mathbf{u}_w^\top\mathbf{v}_{\bar{c}_j})\right)\right)$

- $$\frac{\partial \log \sigma(\mathbf{u}^\top \mathbf{v})}{\partial \mathbf{u}} = (1 - \sigma(\mathbf{u}^\top \mathbf{v}))\frac{\partial \mathbf{u}^\top \mathbf{v}}{\partial \mathbf{u}} = (1 - \sigma(\mathbf{u}^\top \mathbf{v}))\mathbf{v}$$
- $$\frac{\partial \log \sigma(\mathbf{u}^\top \mathbf{v})}{\partial \mathbf{v}} = (1 - \sigma(\mathbf{u}^\top \mathbf{v}))\frac{\partial \mathbf{u}^\top \mathbf{v}}{\partial \mathbf{v}} = (1 - \sigma(\mathbf{u}^\top \mathbf{v}))\mathbf{u}$$

- За всяка двойка от целева дума и контекстна дума (w, c) градиента е ненулев само за векторите $\mathbf{u} = U_{\bullet,w}$ и $\mathbf{v} = V_{\bullet,c}$.

- Сложността за спускането по градиента е пропорционална на $BMkn$ и не зависи от $|L|$.

План на лекцията

1. Формалности за курса (5 мин)

2. Преглед на използването на влагане на думи за класификация на документи (15 мин)

3. Невронен езиков модел на Бенджио и съавтори (15 мин)

4. Моделът Word2Vec CBOW (20 мин)

5. Моделът Word2Vec skip-gram negative-sampling (20 мин)

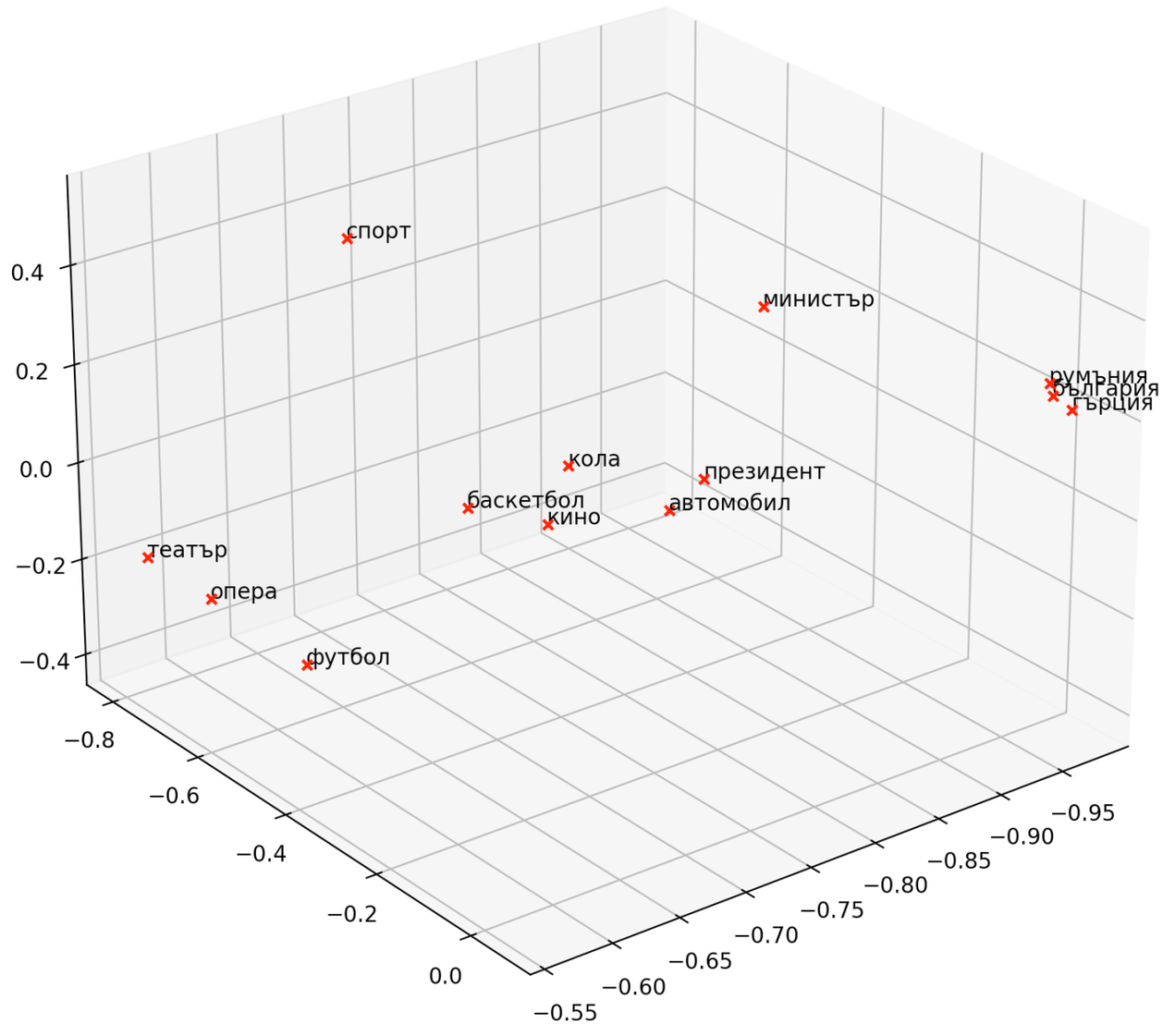
6. **Оценяване на влагане на думи и невронни езикови модели (15 мин)**

Оценяване на влагане на думи

- Вътрешно оценяване:
  - чрез сравняване с ръчно направени корпуси за семантична близост между думи,
  - чрез синонимни речници,
  - чрез аналогии.
- Външно оценяване:
  - Чрез оценяване на качеството на резултатите при вграждане в други задачи — за езиков модел, за класификация на документи, и т.н.

Резултати с word2vec CBOW модел

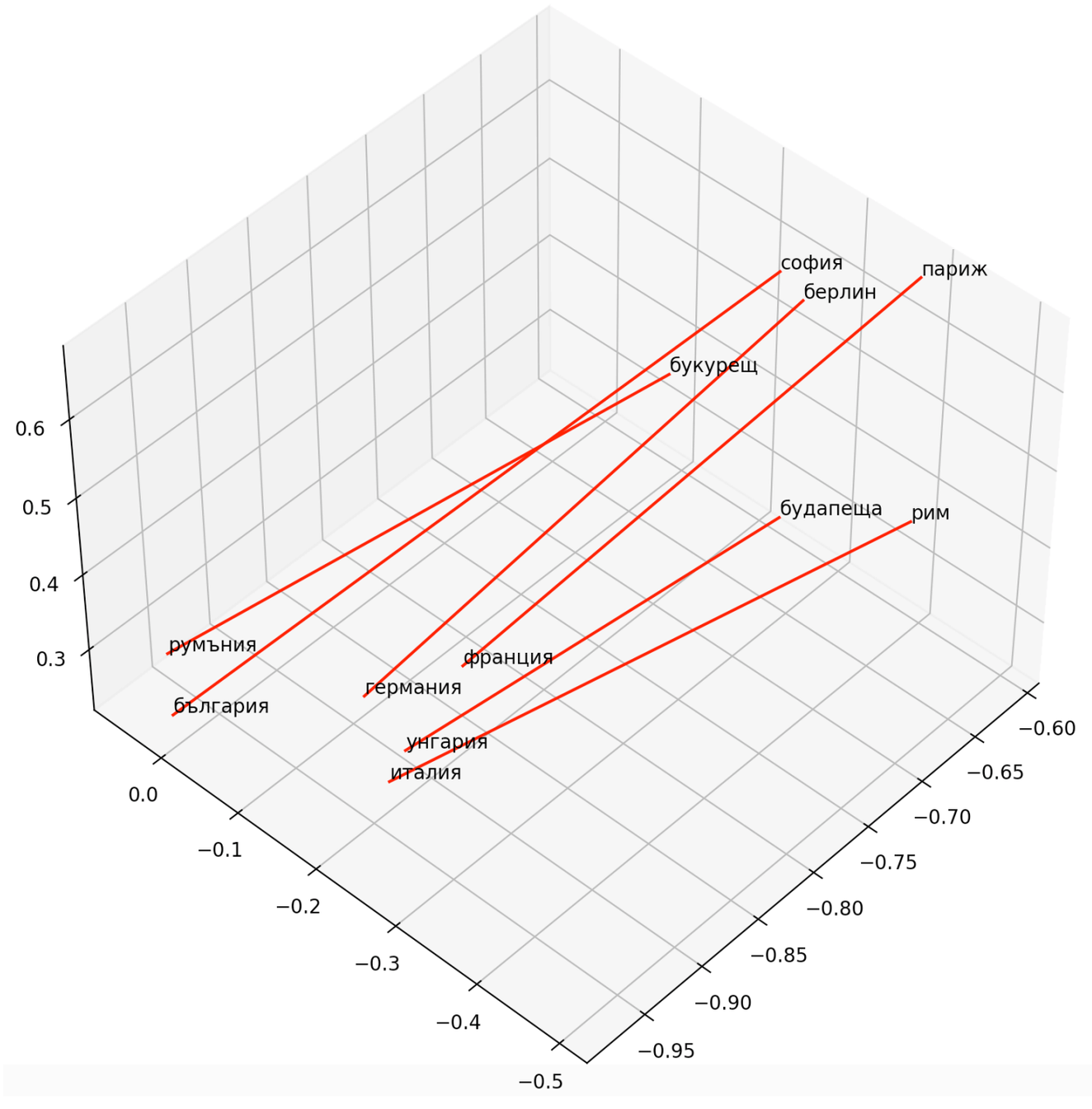
```
[('гърция', 1.0),
 ('турция', 0.7382911131458996),
 ('сирия', 0.6989079915753336),
 ('армения', 0.698287601199191),
 ('израел', 0.691376240721976)]
[('футбол', 1.0000000000000002),
 ('хандбал', 0.8490022078714494),
 ('водна', 0.8460694962033068),
 ('баскетбол', 0.840756204748723 7),
 ('топка', 0.8288446830880019)]
[('град', 0.9999999999999999),
 ('курорт', 0.7977045542374148),
 ('район', 0.7571128029073406),
 ('село', 0.7281806378402222),
 ('окръг', 0.7039934400673319)]
```



Представяне на аналогии и перплексия

- Една и съща аналогия между различни думи се влага в близки вектори в семантичното пространство

- Перплексията на Word2Vec CBOW модела е: **56.1** (за сравнение за 3-грамния модел е над 70)



Сравнение между n-грамен езиков модел и невронен езиков модел

- За невронен езиков модел

    1. Размерът на модела не зависи от контекста (при CBOW) и от големината на корпуса.

    2. Няма нужда от експлицитно изглаждане — естествено се научава обобщение за нови контексти.

    3. **Перплексията е по-ниска!**

- За n-грамен езиков модел

    1. Обучението на модела се свежда до броене на срещания в корпуса и става значително по-бързо.

    2. Прилагането на модела върху даден текст става за време пропорционална на големината на текст и не зависи от речника.

Заключение

- Влаганията Word2Vec са широко използвани за получаване на предварителни влагания на думи. В интернет може да се намерят готови натренирани влагания за много езици.

- Показва се, че моделът Word2Vec Negative-Sampling всъщност е еквивалентен на принципен компонентен анализ върху матрицата на съвместни срещания получена с поточкова взаимна информация.

  - *Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., Advances in Neural Information Processing Systems 27, pages 2177–2185, 2014.*

- Съществуват много други ефективни модели за невронно влагане на думи. Сред по-известните е моделът GloVe:

  - *Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: global vectors for word representation. In Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543, October 2014.*

- Следващата лекция ще разгледаме по-съвършенни невронни езикови модели, с които се постига още по-добра перплексия.