

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

- 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:
$$\Pr_{W,\mathbf{b}}[y = c \mid \mathbf{x}] = \text{softmax}(W\mathbf{x} + \mathbf{b})_c$$
- Този подход ни даде значително подобрение на резултатите спрямо наивния Байсов класификатор.
- Въщност, подобрението се дължи в голяма степен на представянето на документите в гъсто векторно пространство.
- Как получихме документните вектори?

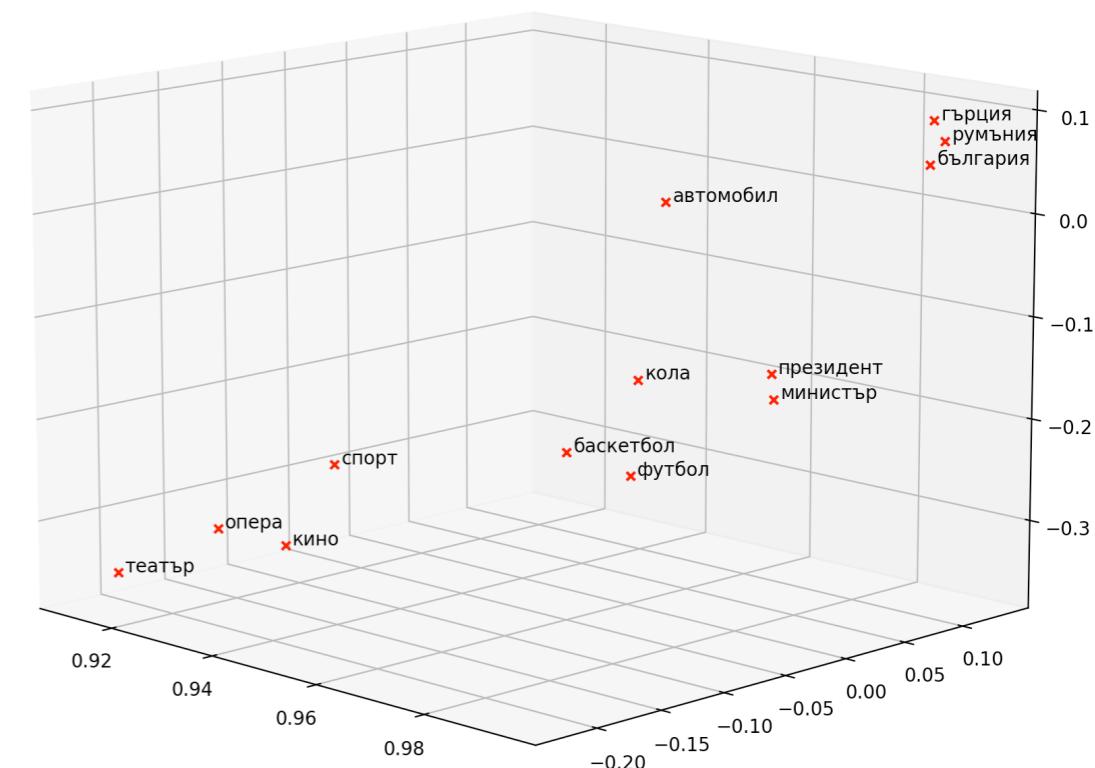
От миналите лекции

- В лекция 6 разглеждахме влагане на термове в гъсто, нискомерно векторно пространство, чрез използване на принципен компонентен анализ.
- Матрицата на влагането (Embedding Matrix) означаваме $E \in \mathbb{R}^{M \times |L|}$, където M е размерността 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 $t_1, t_2, \dots, t_{|d|}$, а one-hot вектора за терма t означаваме с $\chi_t \in \mathbb{R}^{|L|}$, то влагането CBOW (Continious Bag of Words) дефинираме като
$$\mathbf{x} = \text{CBOW}(d) = \text{norm}\left(\sum_{t_i \in d} E\chi_{t_i}\right) = \text{norm}(E \sum_{t_i \in d} \chi_{t_i}),$$
 където
$$\text{norm} : \mathbb{R}^M \rightarrow \mathbb{R}^M$$
 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:

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

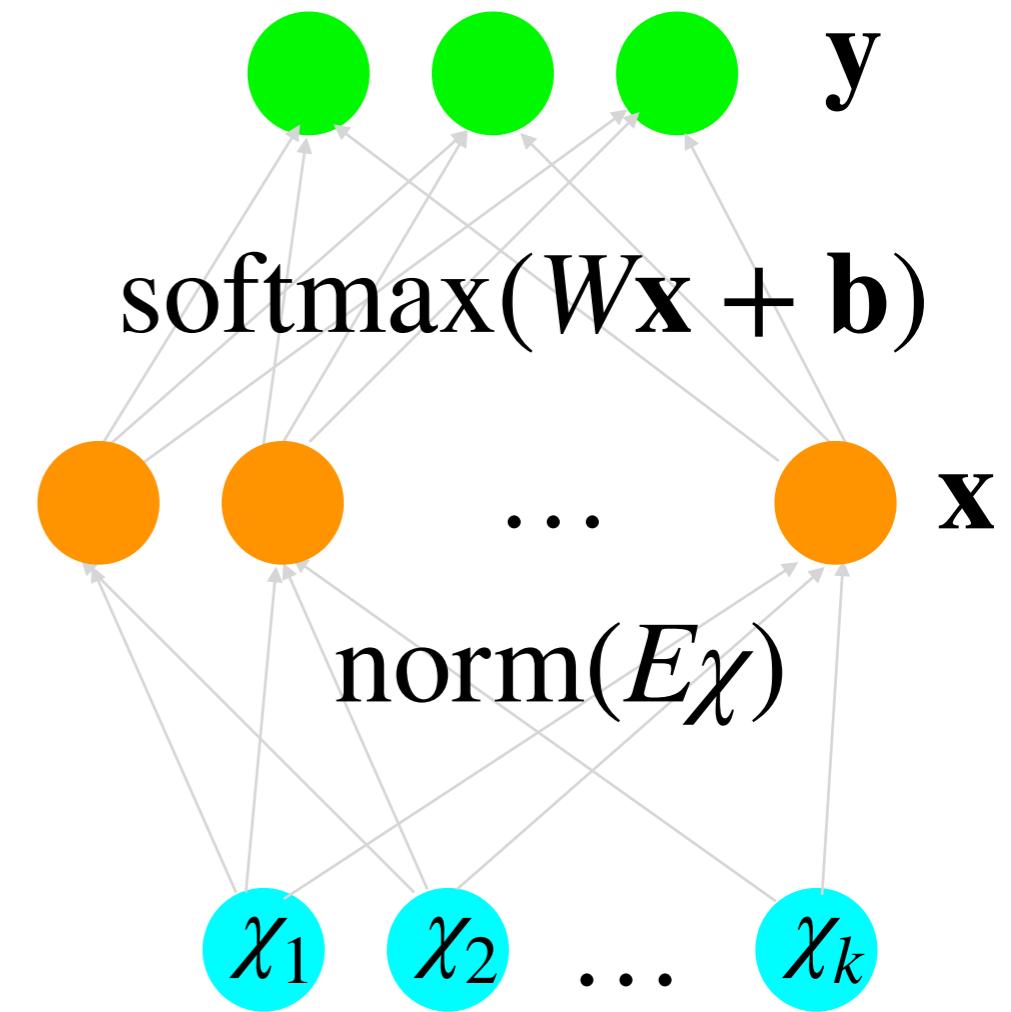
От миналите лекции

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

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

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

- 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 тренираме пълния модел, като едновременно тренираме 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 текстове без анотации.
- Затова е целесъобразно да тренираме влагането предварително с повече данни, така че да научим правилно семантичните връзки между думите.
- На втори етап можем да тренираме само горния слой на мрежата. На този етап може евентуално да дотренираме и предварителното влагане.
- Как да научим влагането от текстове без анотация?

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 4 дефинирахме езиков модел като фамилия от контекстни локални разпределения $\{\Pr[x | x_1x_2\dots x_n]\}_{x_1x_2\dots x_n \in V^*}$. При Марковските езикови модели от ред k имаме $\Pr[x | \alpha x_1x_2\dots x_{k-1}] = \Pr[x | x_1x_2\dots x_{k-1}]$ за всяко начало $\alpha \in V^*$.
- Невронен k -грамен езиков модел от статията *Yoshua Bengio et al., A neural probabilistic language model. Journal of Machine Learning Research, 3:1137–1155, March 2003.*

$$\Pr[\cdot | w_1w_2\dots w_{k-1}] = \mathbf{y} = \text{softmax}(W^{(2)}\mathbf{h} + \mathbf{b}^{(2)})$$

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

$$\mathbf{x} = \begin{bmatrix} E\chi_{w_1} \\ \vdots \\ E\chi_{w_{k-1}} \end{bmatrix}, \text{ където}$$

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

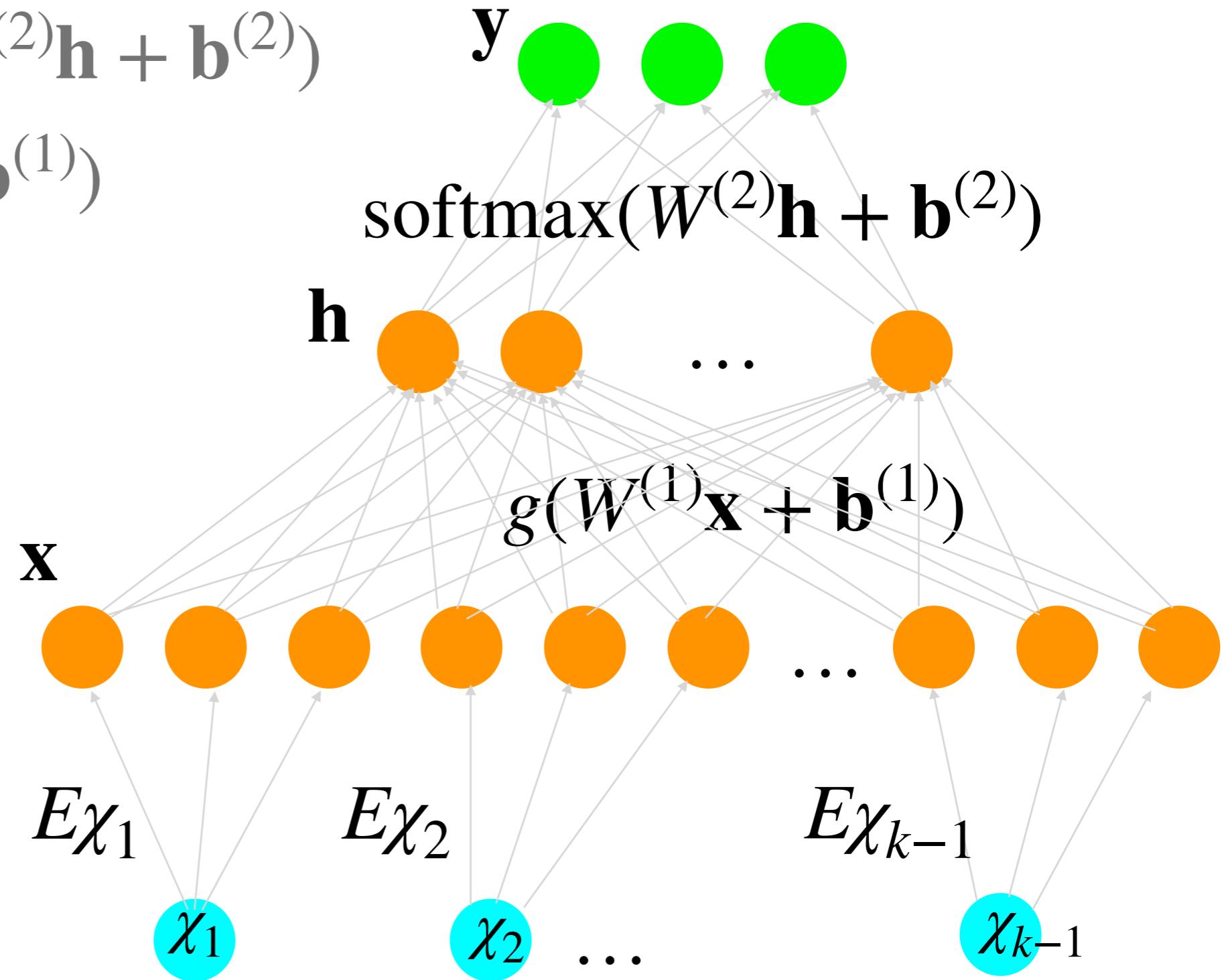
- В модела на Бенджио влаганията на думите от контекста се конкатенират за получаването на входния вектор $\mathbf{x} \in \mathbb{R}^{(k-1)M}$.
- В междинния слой, чрез линеен перцепtron се получава скрит вектор $\mathbf{h} \in \mathbb{R}^N$, който отразява контекста.
- В последния слой, контекстния вектор \mathbf{h} се преобразува през втори перцепtron и softmax, 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 изисква $M = N$ и $E^T = W^{(2)}$.
- 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.

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

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

$$\mathbf{x} = \begin{bmatrix} E\chi_{w_1} \\ \vdots \\ E\chi_{w_{k-1}} \end{bmatrix}$$



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

$$H_{\mathbf{X}} = -\frac{1}{|\mathbf{X}|} \sum_{w \in \mathbf{X}} \log \Pr[w | \mathbf{c}_w],$$
 като за локалното вероятностно разпределение

$\Pr[w | \mathbf{c}]$ ще използваме по-прост модел, който формално може да не води до езиков модел.

- Ако се интересуваме само от влагането на думите то разпределението $\Pr[w | \mathbf{c}]$ не ни е нужно и може да учим други — по-прости разредения.
- Миколов и съавтори разработват през 2013 няколко по-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.

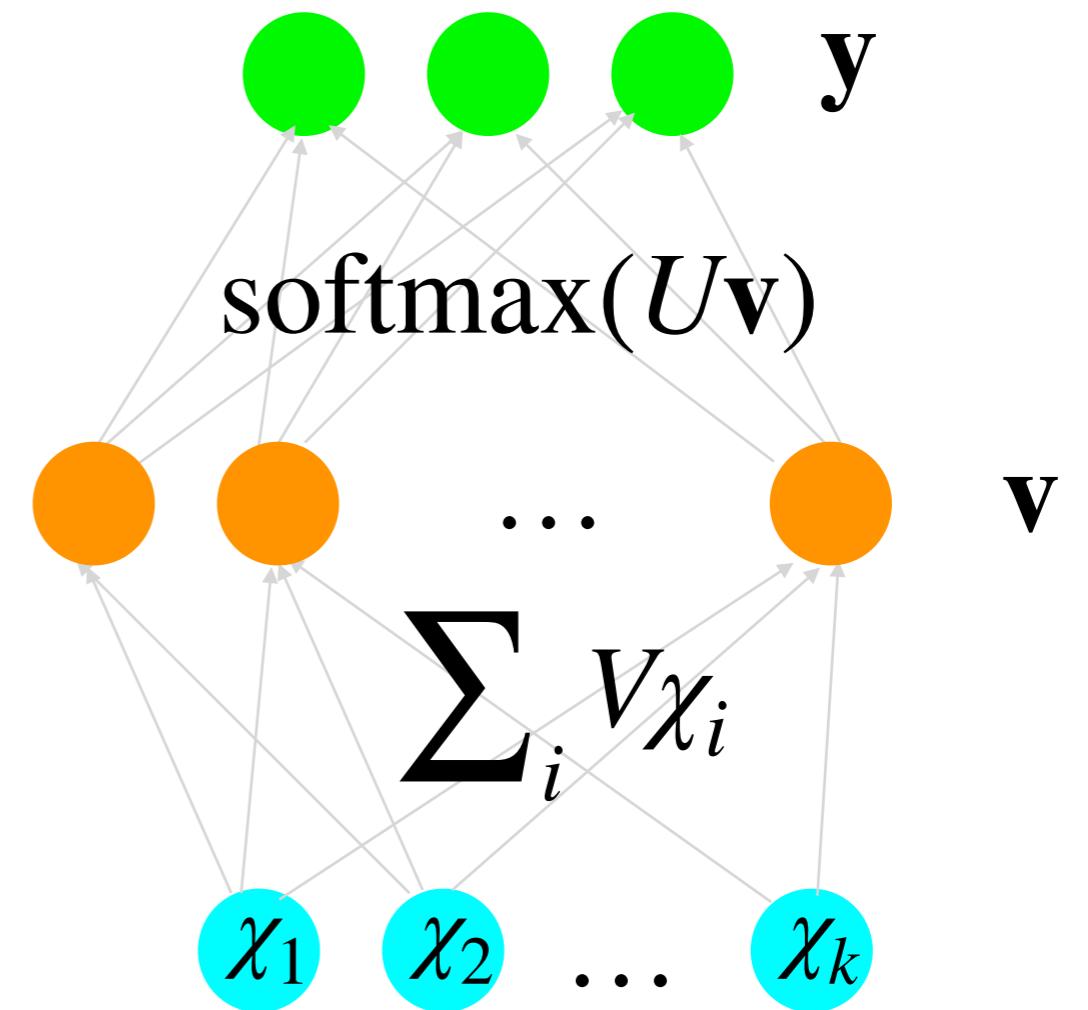
Моделът Word2Vec CBOW

- Ще предполагаме, че контекстът \mathbf{c} на думата $w = t_n$ (n -тата дума в корпуса \mathbf{X}) е списък от k думи в околност t_n . Може $\mathbf{c} = t_{n-k} \dots t_{n-1}$. Но често се избира контекста да са думите около w т.e. $\mathbf{c} = t_{n-k/2} \dots t_{n-1} t_{n+1} \dots t_{n+k/2}$.
- 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: $\mathbf{v}_c = \sum_{c_i \in \mathbf{c}} V\chi_{c_i} = \sum_{c_i \in \mathbf{c}} \mathbf{v}_{c_i}$
- Ще моделираме вероятността $\Pr[w | \mathbf{c}] = \text{softmax}(U^\top \mathbf{v}_c)_w = \frac{e^{\mathbf{u}_w^\top \mathbf{v}_c}}{\sum_{t \in V} e^{\mathbf{u}_t^\top \mathbf{v}_c}}$ и ще минимизираме кросентропията: $H_{\mathbf{X}}(U, V) = -\frac{1}{|\mathbf{X}|} \sum_{(w, \mathbf{c}) \in \mathbf{X}} \log \Pr[w | \mathbf{c}] = -\frac{1}{|\mathbf{X}|} \sum_{(w, \mathbf{c}) \in \mathbf{X}} \log \text{softmax}(U^\top \mathbf{v}_c)_w$
- Забележка:** Ако $\mathbf{c} = t_{n-k} \dots t_{n-1}$, то ще получим префиксни локални разпределения, които дефинират k -грамен марковски езиков модел. В противен случай ще получем фамилия от локални разпределения, които в общия случай не определят езиков модел.

Модельт Word2Vec CBOW

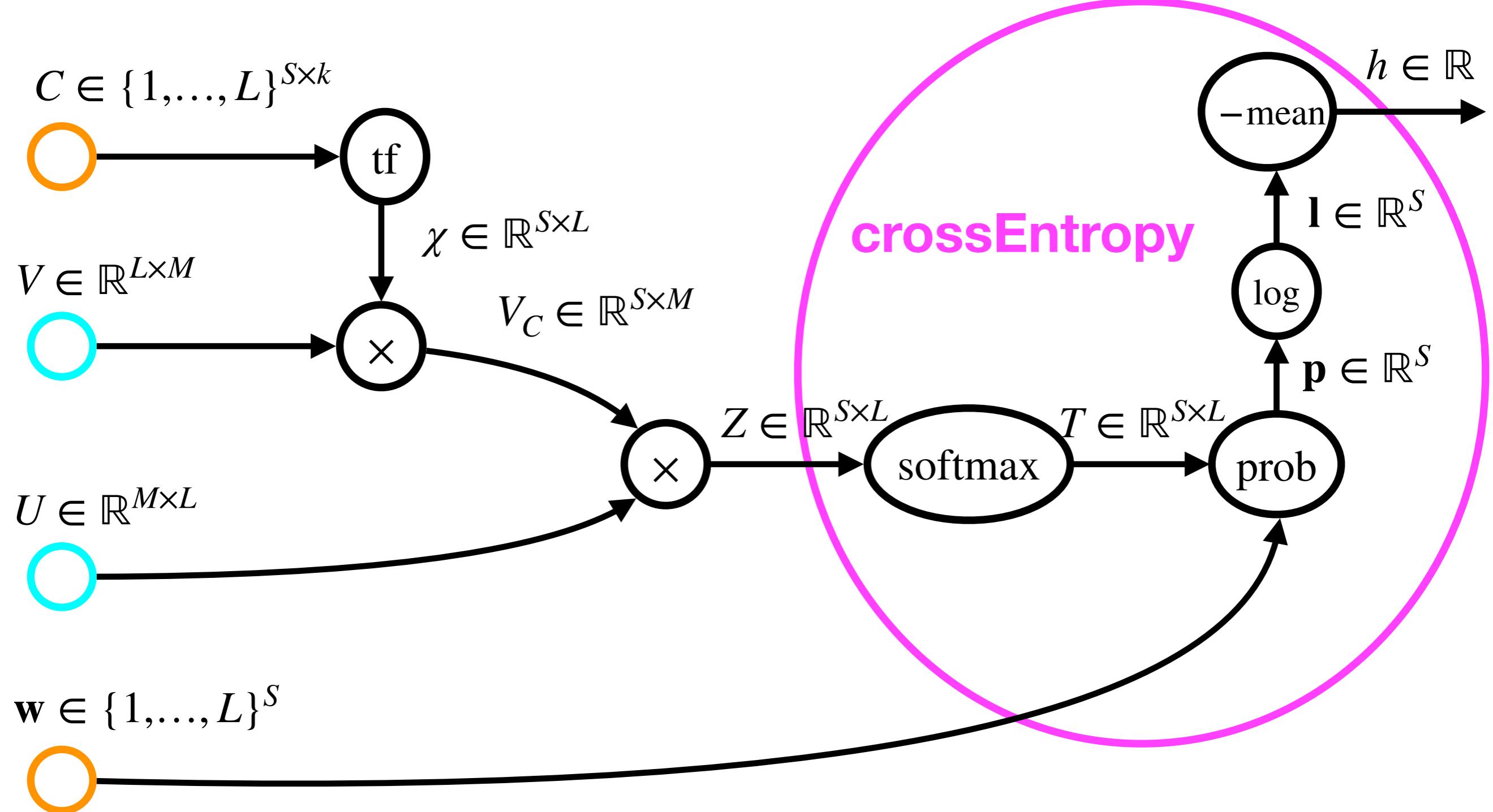
$$\mathbf{y} = \text{softmax}(U\mathbf{v})$$

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



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

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)

Приближение чрез контрастиране с шум Noise-Contrastive Estimation

- Търсим модел $p_\theta[u]$ за истинското разпределение на данните u .
- Ще използваме разпределение $q[u]$ в пространството на данните, което ни е известно, за да генерираме “шум”
- Нека разгледаме двумерна (съвместна) случайна величина (U, C) , където C е Бернулиева случайна величина с $p = 1/2$ и U е случайна величина, **независима** с C , в пространството на данните и (U, C) са със съвместното разпределение:

$$\Pr[U = u, C = c] = \begin{cases} (1/2) p_\theta[u] & c = 1 \\ (1/2) q[u] & c = 0 \end{cases}.$$

- Тогава $\Pr[u | C = 1] = p_\theta[u]$ и $\Pr[u | C = 0] = q[u]$. Използвайки дефиницията и независимостта:

$$\Pr[C = 1 | u] = \frac{p_\theta[u]}{p_\theta[u] + q[u]}, \quad \Pr[C = 0 | u] = 1 - \Pr[C = 1 | u]$$

- Лесно се проверява, че $\Pr[C = 1 | u] = \sigma(\log p_\theta[u] - \log q[u])$ Проверете го!

Приближение чрез контрастиране с шум Noise-Contrastive Estimation

- Нека ни е дадено множество наблюдения $X = \{x_1, x_2, \dots, x_S\}$ от търсеното разпределение и множество от наблюдения $Y = \{y_1, y_2, \dots, y_S\}$ на шум с разпределение $q(u)$.
- Логаритъм от правдоподобието на съвместната случайна величина за наблюденията X (при условие $C = 1$) е: $\sum_{i=1}^S \log \sigma(\log p_\theta[x_i] - \log q[x_i])$
- Логаритъм от правдоподобието на съвместната случайна величина за наблюденията Y (при условие $C = 0$) е: $\sum_{i=1}^S \log(1 - \sigma(\log p_\theta[y_i] - \log q[y_i])) = \sum_{i=1}^S \log \sigma(\log q[y_i] - \log p_\theta[y_i])$
- Логаритъм от правдоподобието на съвместната случайната величина за наблюденията $X \cup Y$ е:

$$J(\theta) = \sum_{i=1}^S \log \sigma(\log p_\theta[x_i] - \log q[x_i]) + \sum_{i=1}^S \log \sigma(\log q[y_i] - \log p_\theta[y_i])$$

Приближение чрез контрастиране с шум Noise-Contrastive Estimation

- **Теорема** Нека истинското разпределение на нашите данни е $p_d[u]$. Тогава функцията $f_m = \log p_d$ максимизира $\bar{J}(f) = \mathbb{E} \log \sigma(f(x) - \log q[x]) + \mathbb{E} \log \sigma(\log q[y] - f(y))$. Ако разпределението $q[u]$ е ненулево за всички точки, в които $p_d[u]$ е ненулево, то f_m е единствен екстремум.
 - Gutmann, Michael & Hyvärinen, Aapo. (2012). *Noise-Contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics*. Journal of Machine Learning Research 13 (2012) 307-361
- **Свойство:** Тъй като резултатът от минимизацията е логаритъм от разпределение, то не се налага моделът експлицитно да изисква резултатът да е нормализирано разпределение (това се гарантира за максимума).
- Най-често метода се прилага за softmax, т.е. $p_\theta[u] = \frac{\exp(g_\theta(u))}{\sum_{u'} \exp(g_\theta(u'))}$. В този случай, ако фамилията от функции g_θ е затворена относно умножение с константа, то може да се положи $p_\theta^0[u] = \exp(g_\theta(u))$ и съответно $\log p_\theta^0[u] = g_\theta(u)$.
- Методът се обобщава директно за случая когато $\Pr[C = 1] = \frac{1}{n}$ за $n \geq 2$:
Задача: Обобщете метода за случая когато $\Pr[C = 1] = 1/n$.

Моделът Word2Vec skip-gram negative-sampling

- Вместо $\Pr[w \mid \mathbf{c}]$ се разглежда $\Pr[\mathbf{c} \mid w]$ и се предполага 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 стремежът е да се минимизира кросентропията:
$$H_{\mathbf{X}}(U, V) = -\frac{1}{|\mathbf{X}|} \sum_{(w, \mathbf{c}) \in \mathbf{X}} \sum_{c_i \in \mathbf{c}} \log \Pr[c_i \mid w] = -\frac{1}{|\mathbf{X}|} \sum_{(w, \mathbf{c}) \in \mathbf{X}} \sum_{c_i \in \mathbf{c}} \log \text{softmax}(V^T \mathbf{u}_w)_{c_i}$$
- Разпределението $\Pr[c_i \mid w]$ се търси с Noise-Contrastive Estimation. За целта се моделира $\log \Pr[c_i \mid w] = \mathbf{u}_w^T \mathbf{v}_{c_i}$. Търсят се параметрите U, V , които максимизират:
$$J_{\mathbf{Z}, \bar{\mathbf{Z}}}(U, V) = \sum_{(w, c) \in \mathbf{Z}} \log \sigma(\mathbf{u}_w^T \mathbf{v}_c - \log nq(c)) + \sum_{(w, \bar{c}) \in \bar{\mathbf{Z}}} \log \sigma(\log nq(\bar{c}) - \mathbf{u}_w^T \mathbf{v}_{\bar{c}}),$$
 където \mathbf{Z} е множество от коректни двойки от целева дума и контекстна дума, а $\bar{\mathbf{Z}}$ е множество от некоректни двойки (шум), $q[u]$ е разпределението на шума и $n = |\bar{\mathbf{Z}}| / |\mathbf{Z}|$.

- 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 $j = 1, 2, \dots, n$ избираме случайно от нашия речник, така че $\bar{c}_j \neq c$, използвайки

$$\text{монограмно разпределение } \Pr_1(\bar{c}) = \frac{\#(\bar{c})}{\sum_{w \in V} \#(w)}.$$

- Вместо класическото монограмно разпределение, за да се повиши вероятността да се избират по-редки думи често се използва разпределението

$$\Pr_{0.75}(\bar{c}) = \frac{\#(\bar{c})^{0.75}}{\sum_{w \in V} \#(w)^{0.75}}.$$

- Миколов и съавтори използват опростяване на Noise-Contrastive Estimation, което наричат **Negative-Sampling**. Опростяването се състои в игнорирането на вероятността на шума, което не е математически коректно. Според авторите това опростяване не влошава качеството на влагането на думите:

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

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

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

- 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 $B M n$ и не зависи от $|L|$.

План на лекцията

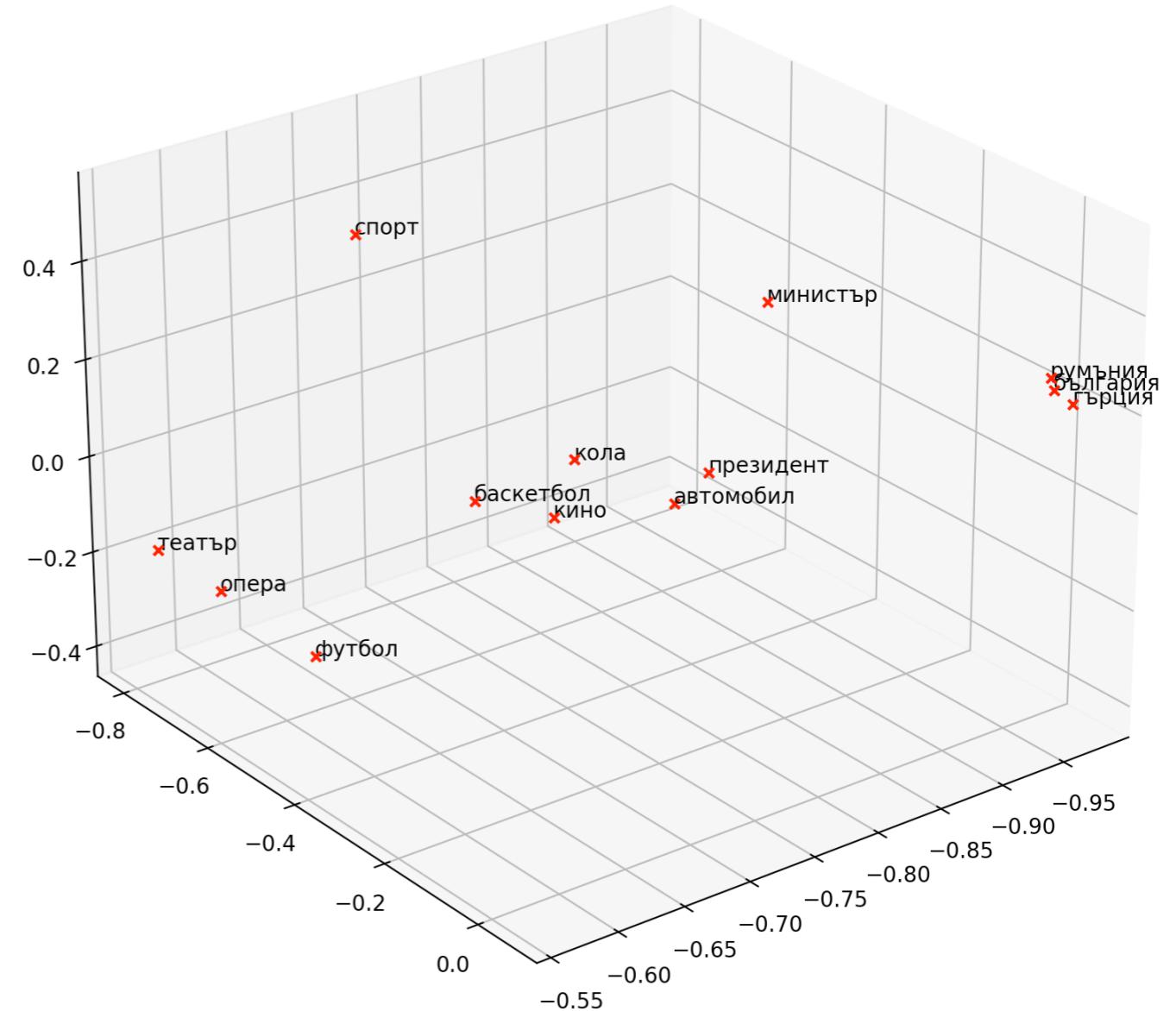
1. Формалности за курса (5 мин)
2. Преглед на използването на влагане на думи за класификация на документи (15 мин)
3. Невронен езиков модел на Бенджио и съавтори (15 мин)
4. Моделът Word2Vec CBOW (20 мин)
5. Моделът Word2Vec skip-gram negative-sampling (20 мин)
- 6. Оценяване на влагане на думи и невронни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

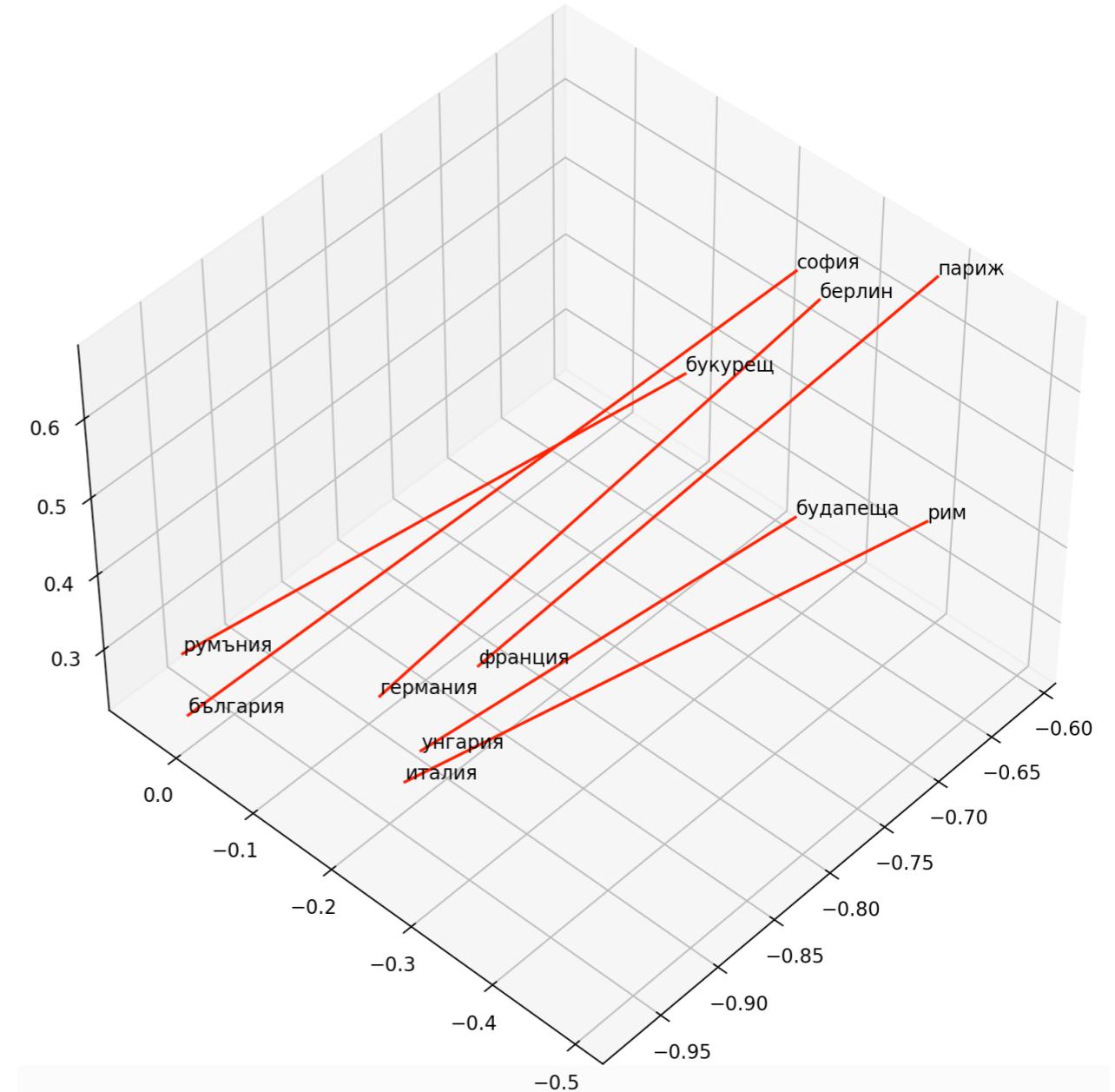
Резултати с word2vec CBOW модел

```
[('гърция', 1.0),  
 ('турция', 0.7382911131458996),  
 ('сирия', 0.6989079915753336),  
 ('армения', 0.6982876011999191),  
 ('израел', 0.691376240721976)]  
[['футбол', 1.0000000000000002),  
 ('хандбал', 0.8490022078714494),  
 ('водна', 0.8460694962033068),  
 ('баскетбол', 0.8407562047487237),  
 ('топка', 0.8288446830880019)]  
[['град', 0.9999999999999999],  
 ('курорт', 0.7977045542374148),  
 ('район', 0.7571128029073406),  
 ('село', 0.7281806378402222),  
 ('окръг', 0.7039934400673319)]
```



Представяне на аналогии

- Една и съща **аналогия** между различни думи се влага в близки вектори в семантичното пространство
- Виена : Дунав ::
Париж : x
- Германия : Берлин ::
Франция : x



$$x = \arg \max_{x \in V} \cos(\overrightarrow{E(x)}, \overrightarrow{E(\text{Франция})} + \overrightarrow{E(\text{Берлин})} - \overrightarrow{E(\text{Германия})})$$

Заключение

- Влаганията Word2Vec бяха широко използвани преди 5-10 години 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 в някакъв смисъл 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-добри невронни езикови модели, с които се постига значително по-добра перплексия.