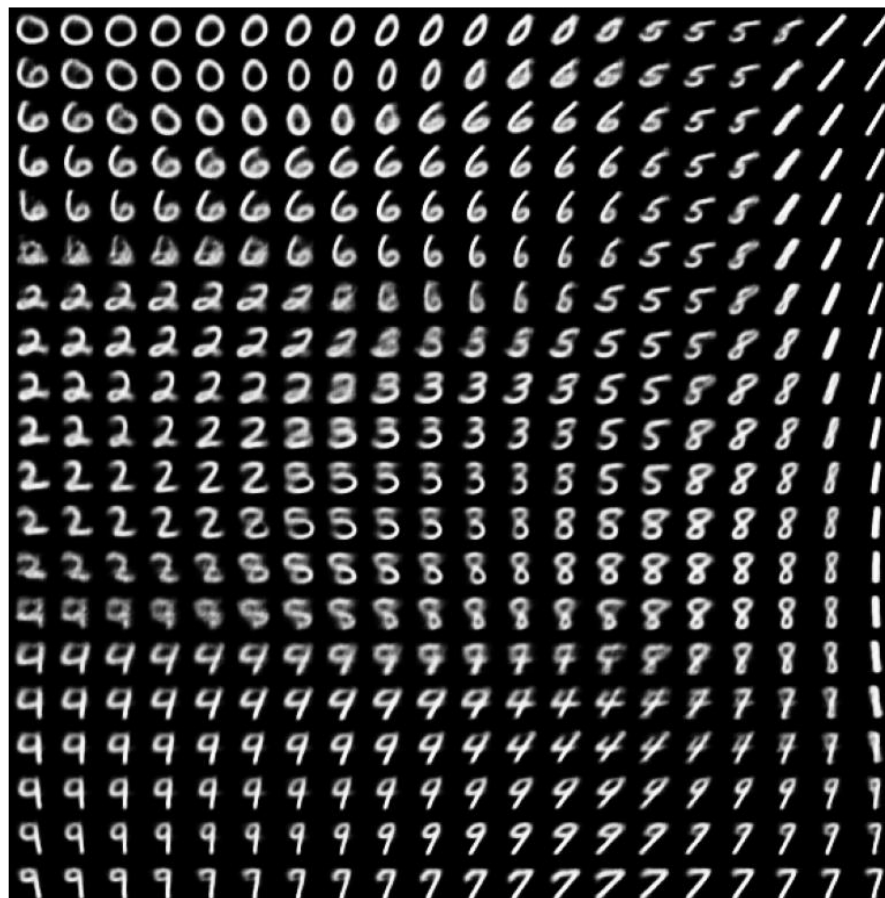


CNN

vnútro hyperkocky



Podobnosť: euklidovská vzdialenosť

Euklidovská podobnosť

vektor (u_1, u_2, \dots, u_N) má **veľkosť**

$$|u| = \sqrt{(u_1^2 + u_2^2 + \dots + u_N^2)}$$

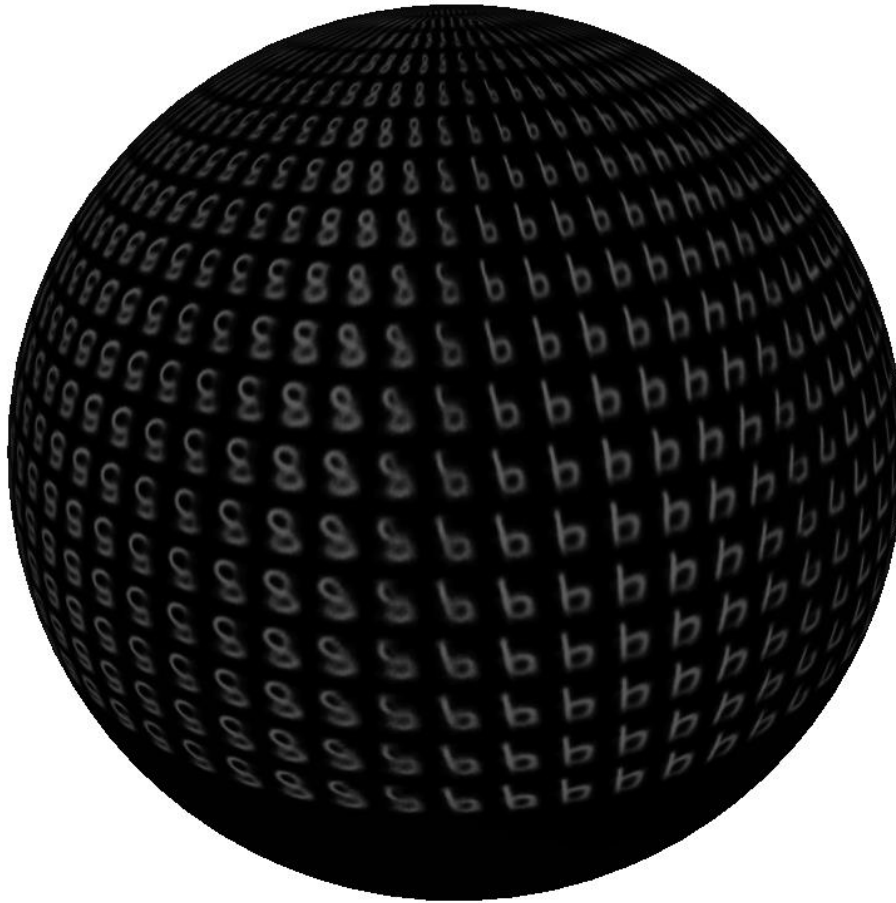
(odmocnina súčtu druhých mocnín koordinátov)

dva vektory (u_1, u_2, \dots, u_N) a (v_1, v_2, \dots, v_N) majú

podobnosť $|u - v|$

Transformery

Povrch hypergul'e



podobne
Zemi aj tu
sú pohoria a
priekopy

Podobnosť: cosínusová podobnosť

Kosínusová podobnosť

vektor (u_1, u_2, \dots, u_N) má **veľkosť**

$$|u| = \sqrt{u_1^2 + u_2^2 + \dots + u_N^2}$$

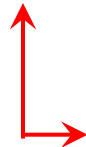
dva vektory (u_1, u_2, \dots, u_N) a (v_1, v_2, \dots, v_N) zvierajú **uhol** ϕ

$$\cos \phi = \frac{u_1 v_1 + u_2 v_2 + \dots + u_N v_N}{|u| |v|}$$

(kosínus uhla je podiel **skalárneho súčinu** koordinátov a súčinu veľkostí vektorov)



$$\cos \phi = 1$$



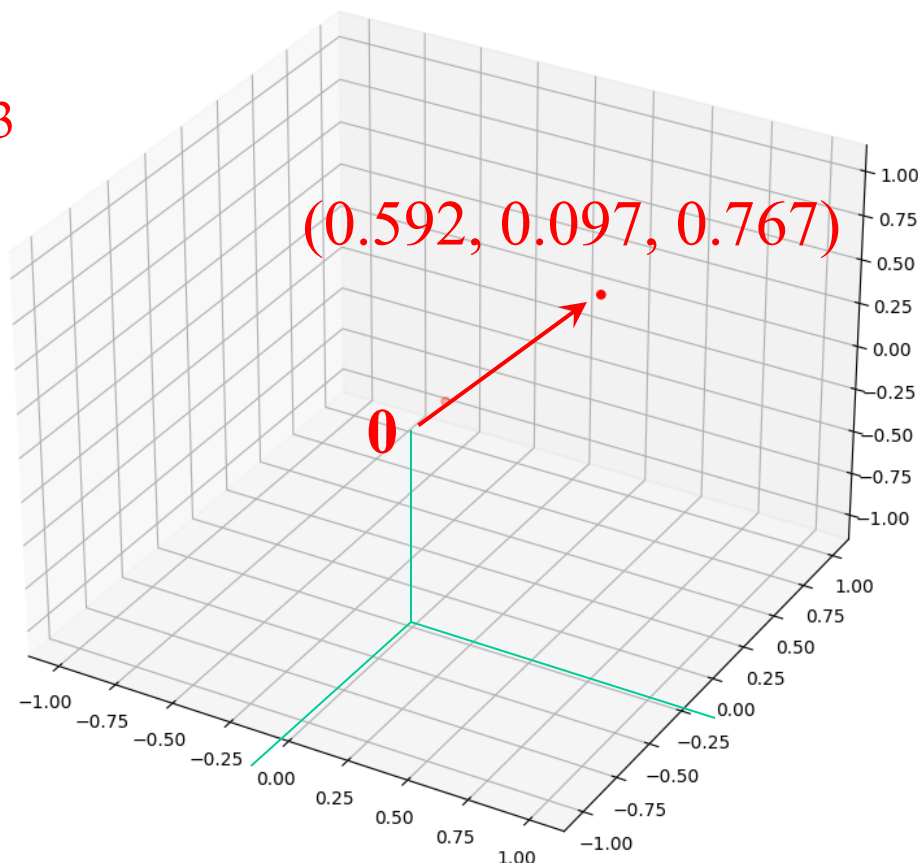
$$\cos \phi = 0$$



$$\cos \phi = -1$$

Embedding (Vnoreníe)

$N = 3$

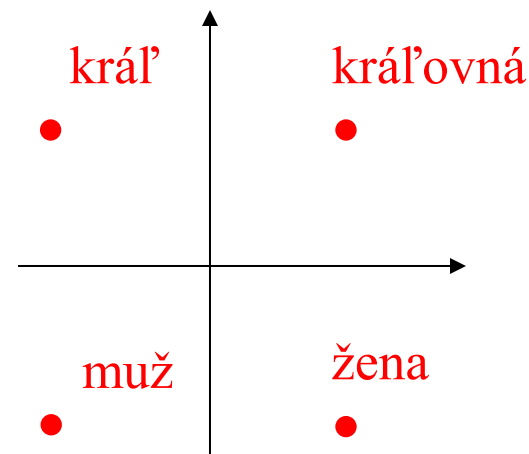


Význam reprezentujeme vektorom (či bodom) v
mnohorozmernom priestore príznačkov

Reprezentácia slova vektorom

slovník

kráľ	$(-1,1)$
kráľovná	$(1,1)$
muž	$(-1,-1)$
žena	$(1,-1)$



v slovníku máme 4 slová a každému priradíme vektor dimenzie 2
máme dva príznaky: pohlavie a vládnutie

Aritmetika významov

Kto sa má k žene tak, ako kráľ ku kráľovnej ?

Riešime rovnicu:

$$x - \text{žena} = \text{kráľ} - \text{kráľovná}$$

$$x - (1, -1) = (-1, 1) - (1, 1)$$

$$x - (1, -1) = (-2, 0)$$

$$x = (-2, 0) + (1, -1)$$

$$x = (-1, -1)$$

$$x = \text{muž}$$

slovník

kráľ	$(-1, 1)$
kráľovná	$(1, 1)$
muž	$(-1, -1)$
žena	$(1, -1)$

Aritmetika významov

Kto sa má k žene tak, ako kráľ ku kráľovnej ?

Riešime rovnicu:

$$x - \text{žena} = \text{kráľ} - \text{kráľovná}$$

$$x - (1, -1) = (-1, 1) - (1, 1)$$

$$x - (1, -1) = (-2, 0)$$

$$x = (-2, 0) + (1, -1)$$

$$x = (-1, -1)$$

$$x = \text{muž}$$

Embedding (Vnorenie)

slovník

kráľ	(-1,1)
kráľovná	(1,1)
muž	(-1,-1)
žena	(1,-1)

Wipeout (Vynorenie)

Ako nájsť správne hodnoty jednotlivých vektorov automaticky ?

Zoberieme korpus:

V kráľovskom dvore si kráľ s kráľovnou vychutnávali plesy a hudobné predstavenia. Muž a žena, ktorí boli oddaní manželským záväzkom, sa stali kráľom a kráľovnou v ich vlastnej domácnosti. Na námestí sa stretol muž so ženou a spolu snívali o tom, že by mohli byť kráľom a kráľovnou svojej vlastnej krajiny. Kráľ so svojou kráľovnou prechádzali rozkvitnutým záhradným labyrintom, objímajúc sa pod každým rozkvitnutým stromom. Muž sa rozhodol, že chce byť kráľom svojho osudu, a žena ho podporovala, aby sa stala jeho vernou kráľovnou. Na večernom bankete si kráľ so svojou kráľovnou vymieňali nežné pohľady a tajné úsmevy. V dedine si obyvatelia vybrali muža a ženu, aby sa stali kráľom a kráľovnou na miestnom jarmoku. Kráľovná pozvala mladú ženu do svojho paláca a pomohla jej pripraviť sa na večernú hostinu s kráľom. Muž a jeho milovaná žena spolu snívali o kráľovských príbehoch, ktoré by mohli prežiť. Na slávnostnom obrade bola mladá žena korunovaná za kráľovnú a jej muž sa stal kráľom. ...

a spočítame ako často sa jedno slovo vyskytuje v kontexte druhého (t.j. v texte maximálne tri slová pred ním alebo za ním)

Ako nájsť správne hodnoty jednotlivých vektorov automaticky ?

Dostaneme tabuľku početností
a v ideálnom prípade dostaneme:

	kráľ	kráľovná	muž	žena
kráľ	44	22	22	0
kráľovná	22	44	0	22
muž	22	0	44	22
žena	0	22	22	44

Ako nájsť správne hodnoty jednotlivých vektorov automaticky ?

Dostaneme tabuľku početností, ktorú môžeme znormalizovať a v ideálnom prípade dostaneme:

	kráľ	kráľovná	muž	žena
kráľ	1	0.5	0.5	0
kráľovná	0.5	1	0	0.5
muž	0.5	0	1	0.5
žena	0	0.5	0.5	1

Ako nájsť správne hodnoty jednotlivých vektorov automaticky ?

Dostaneme tabuľku početností, ktorú môžeme znormalizovať a v ideálnom prípade dostaneme:

	kráľ	kráľovná	muž	žena
kráľ	1	0.5	0.5	0
kráľovná	0.5	1	0	0.5
muž	0.5	0	1	0.5
žena	0	0.5	0.5	1



<i>slovník</i>	
kráľ	(1,0.5,0.5,0)
kráľovná	(0.5,1,0,0.5)
muž	(0.5,0,1,0.5)
žena	(0,0.5,0.5,1)

a opäť tu platí:

$$\text{muž} - \text{žena} = \text{kráľ} - \text{kráľovná}$$

$$(0.5, 0, 1, 0.5) - (0, 0.5, 0.5, 1) = (1, 0.5, 0.5, 0) - (0.5, 1, 0, 0.5)$$

$$(0.5, -0.5, 0.5, -0.5) = (0.5, -0.5, 0.5, -0.5)$$

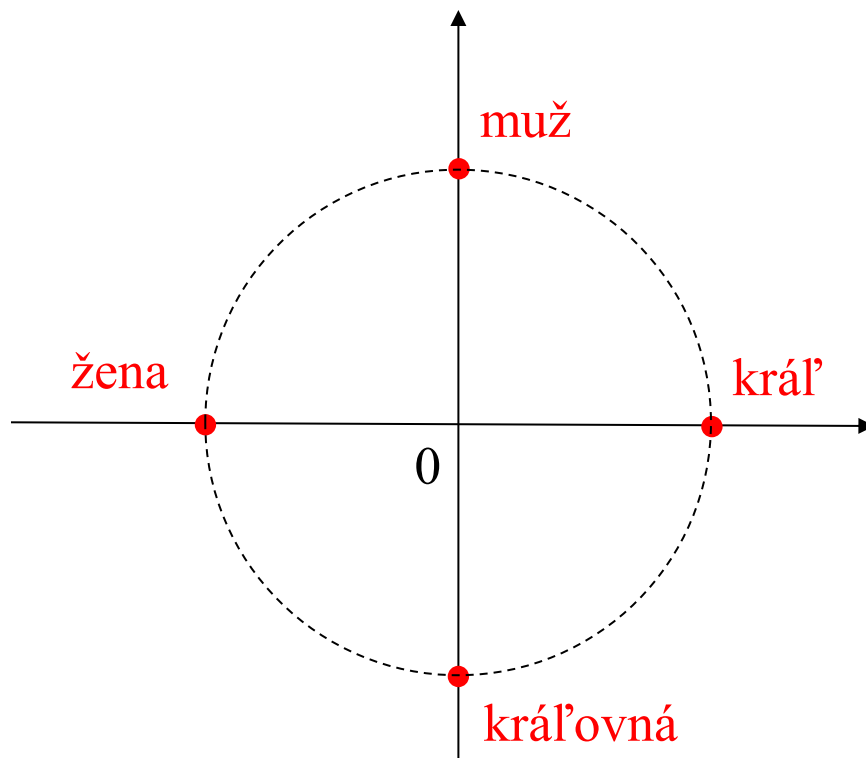
Je možné z automatického slovníka získať podobný tomu ručne urobenému?

Napodiv, odpoveď je kladná. Dokážeme to metódou lineárnej algebry zvanou PCA (Principal Component Analysis), keď jej dáme zredukovať dimenziu automatického slovníka (4) na 2

kráľ	$(1, 0.5, 0.5, 0)$
kráľovná	$(0.5, 1, 0, 0.5)$
muž	$(0.5, 0, 1, 0.5)$
žena	$(0, 0.5, 0.5, 1)$



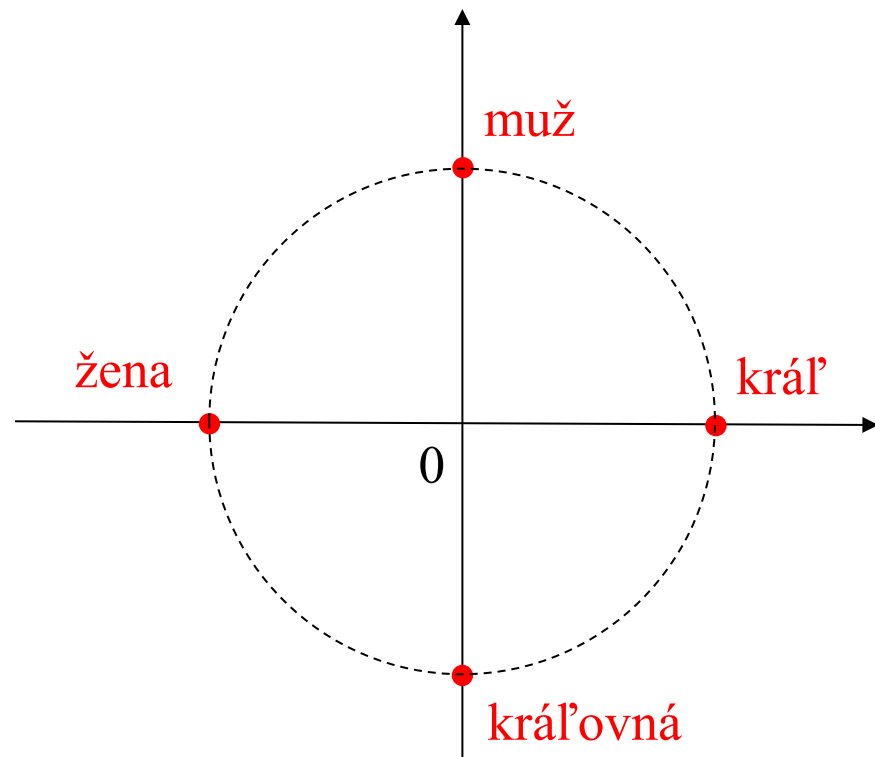
kráľ	$(0.707, 0)$
kráľovná	$(0, -0.707)$
muž	$(0, 0.707)$
žena	$(-0.707, 0)$



Kosínusová podobnosť

muž je teraz podobnejší **kráľovi** ($\cos 90^\circ = 0$) viac než **kráľovnej** ($\cos 180^\circ = -1$) a najpodobnejší sebe ($\cos 0^\circ = 1$)

*(to je vďaka tomu, že PCA
dáta vycentruje, t.j.
priemer je v nule a v tomto
prípade aj otočí o 45°)*

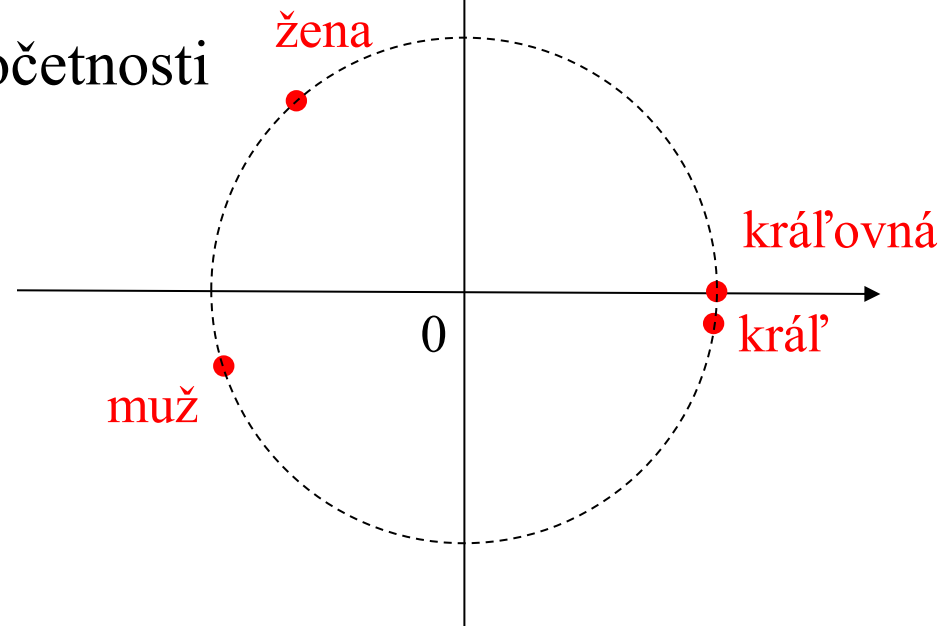


Čo by sa stalo keby početnosti v korpuse neboli optimálne?

	kráľ	kráľovná	muž	žena
kráľ	44	18	12	5
kráľovná	18	32	3	24
muž	12	3	440	220
žena	5	24	220	260

aplikujeme PCA priamo na početnosti
a normalizujeme, dostaneme:

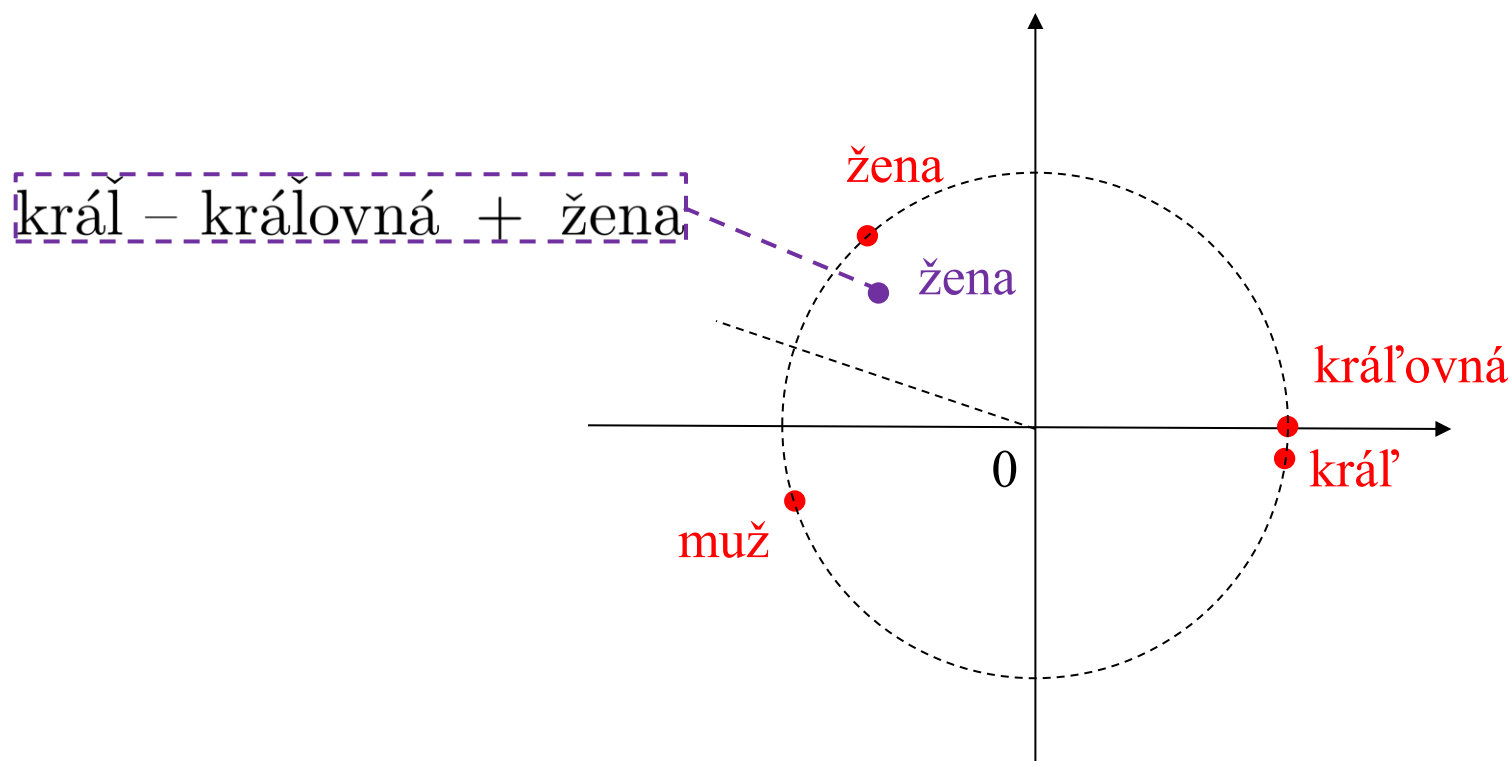
kráľ	(0.99, -0.14)
kráľovná	(1.00, 0.00)
muž	(-0.98, -0.2)
žena	(-0.78, 0.63)



Proporcie máme dobré, ale aritmetika už nefunguje

Kto sa má k žene tak, ako kráľ ku kráľovnej ?

Aritmetika odpovie: žena



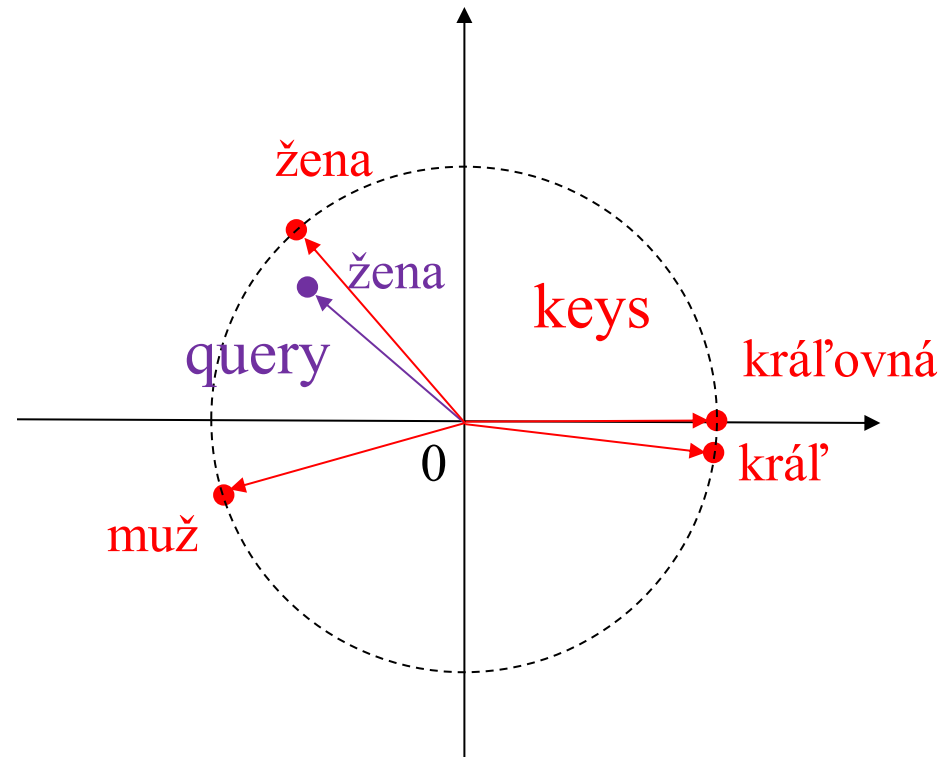
Wipeout (Vynorenie)

Výstupný embedding

Nepovie teda presne „žena“, len približne

Ako spočítame ku
ktorému zo vzorov je
výsledok najbližšie?

Hľadáme maximum
skalárneho súčinu **query**
so všetkými **keys**



Wipeout (Vynorenie)

Výstupný embedding

Máme keys $K = \begin{pmatrix} k_1 \\ k_2 \\ \dots \\ k_l \end{pmatrix}$ a query q

qK^T je vektor skalárnych súčinov q s k_i

Hľadáme teda $\arg \max_i qK^T$

Pokiaľ chceme pravdepodobnosti $\text{softmax}(qK^T)$
pre každú kategóriu:

Implementácia Wipeout-u

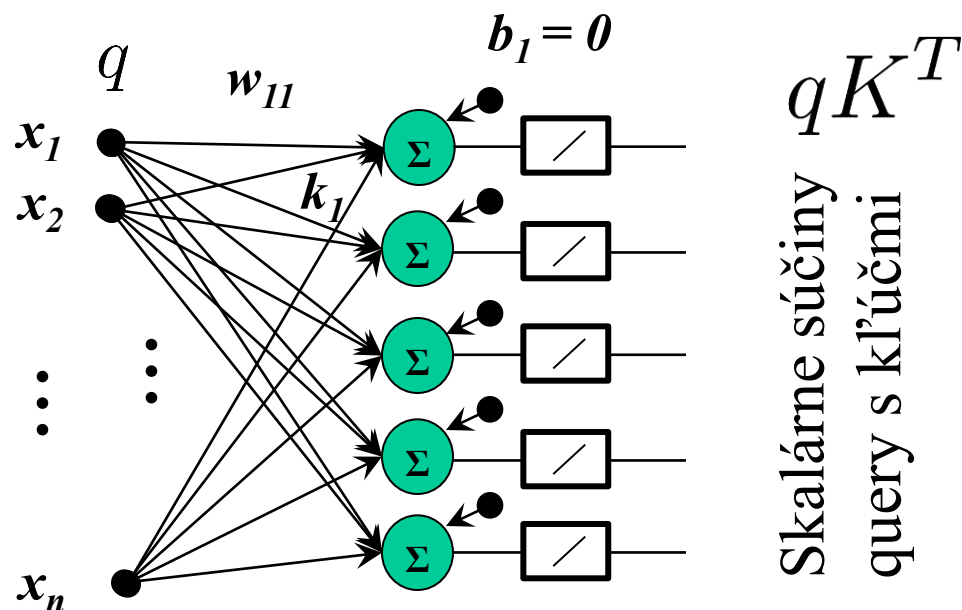
Počet vstupov x_i
= dimenzia priestoru

Počet neurónov
= počet kľúčov

Kľúče sú váhy
 $k_j = w_{1j}, w_{2j}, \dots$

Biasy sú nulové

Aktivácia lineárna



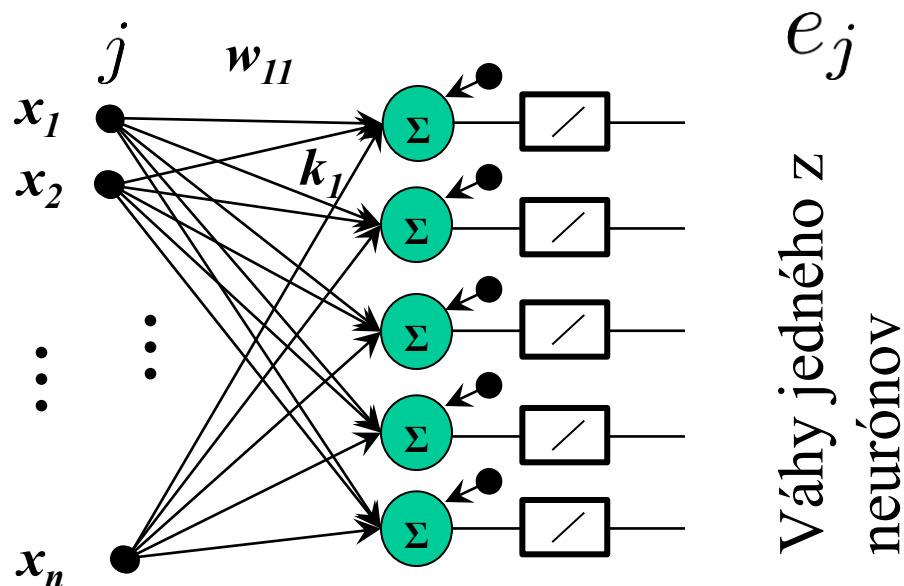
Lineárna projekcia

Implementácia Embedding-u

Vstupom je index j
= token id

Počet neurónov
= počet tokenov

Embedingy sú váhy
 $e_j = w_{1j}, w_{2j}, \dots$



Implementácia Embedding-u

Vstupom sú farby pixelov

$x = \text{flatten}(\text{Patch})$

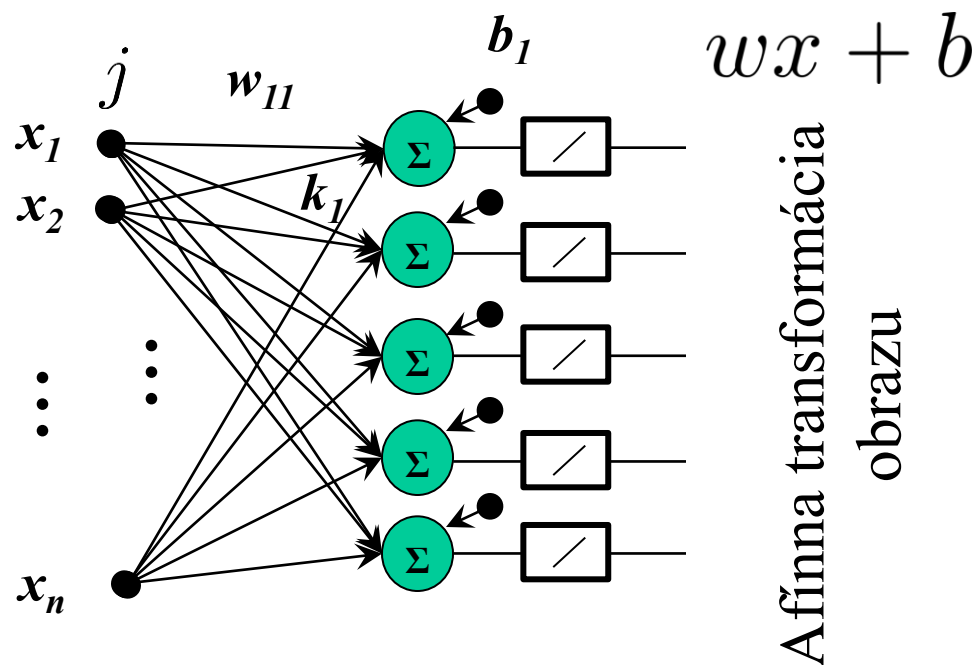
Patch $C \times H \times W$

$3 \times 8 \times 8 \dots 3 \times 32 \times 32$

Počet neurónov je
dimenziou priestoru

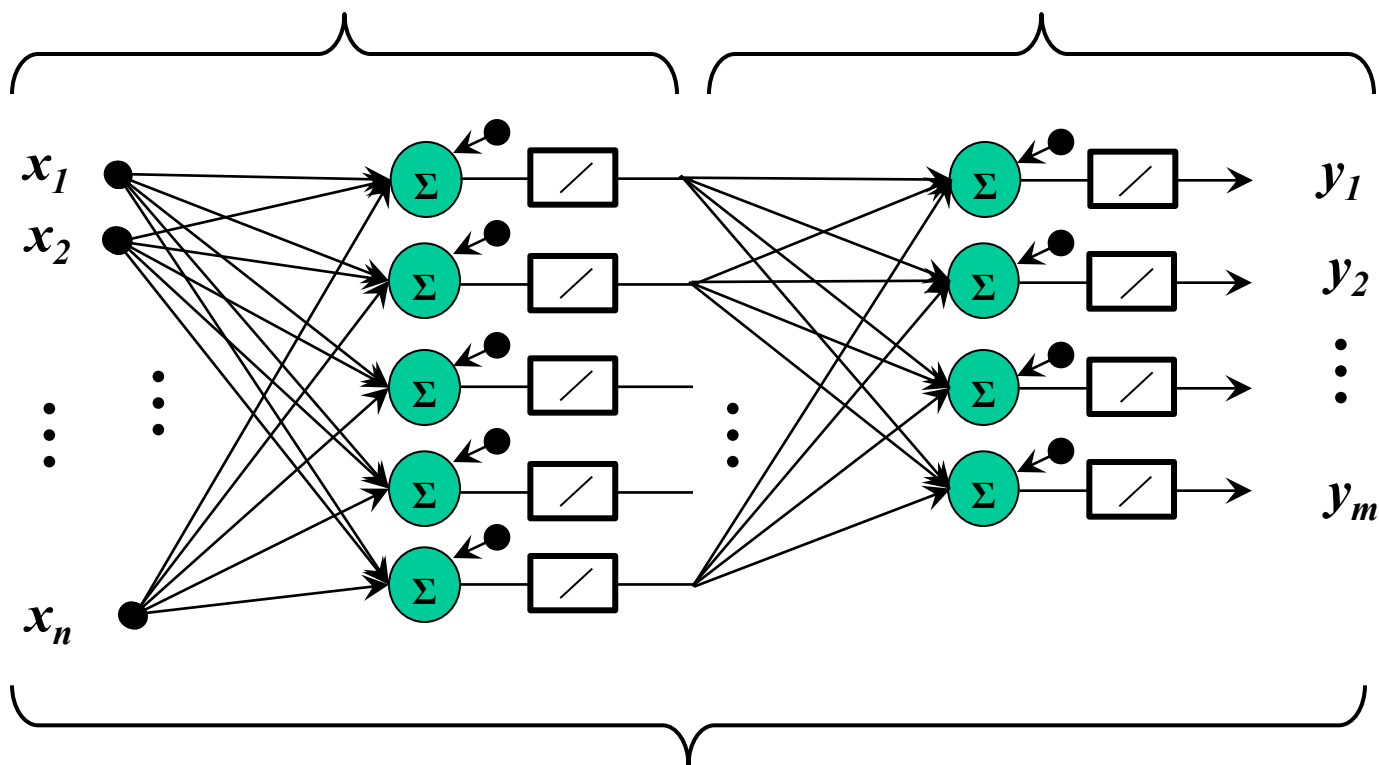
embeddingov

$384 \dots 12288$



Embedding

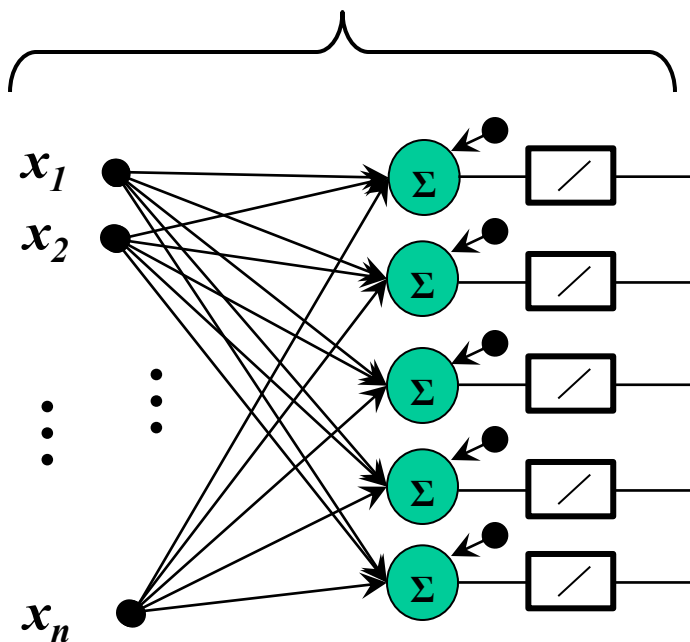
Wipeout



Perceptron

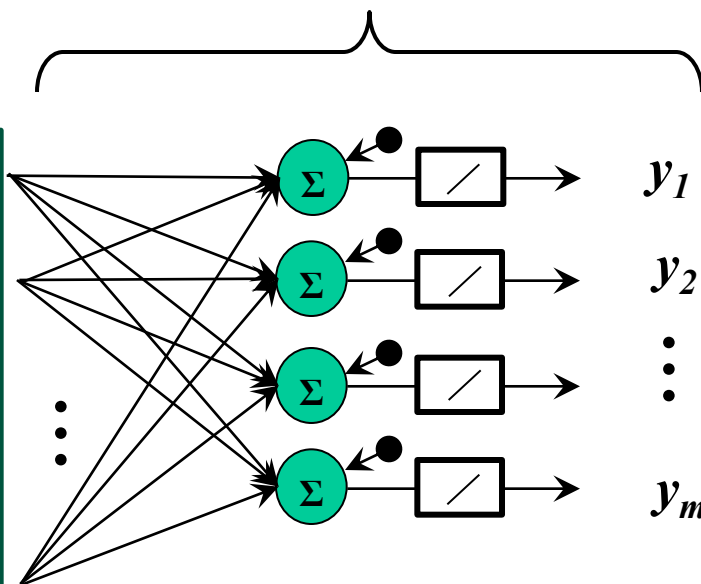
Obraz

Embedding



Layer
norm

Wipeout



Obraz

Layer Normalization

$$\text{LayerNorm}(x) = \gamma \frac{x - \mu}{\sqrt{\frac{1}{d} \sum_{i=1}^d (x_i - \mu)^2 + \epsilon}} + \beta$$
$$\mu = \frac{1}{d} \sum_{i=1}^d x_i$$

Pre jeden vstup je x výstupom z nejakej vrstvy

Z týchto výstupov odčítame priemer

Zrátame smerodajnú odchýlku a vydelíme ňou všetky príznaky

Pritom sa pre každý príznak zvlášť učíme škálu γ a biás β

(počet trénovateľných parametrov je 2 x dimenzia)

LN zabezpečuje, že príznaky budú mať zhruba rovnakú veľkosť

RMS Norm

Root Mean Square Normalization

$$\text{RMSNorm}(x) = \gamma \frac{x}{\sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2 + \epsilon}}$$

Pre jeden vstup je x výstupom z nejakej vrstvy

Zrátame smerodajnú odchýlku a vydelíme ňou všetky príznaky

Pritom sa pre každý príznak zvlášť učíme škálu β

(počet trénovateľných parametrov je rovný dimenzii)

RMS Norm zabezpečuje, že príznaky budú mať zhruba rovnakú veľkosť a budú vycentrované (nemáme biásy)

RMS Norm je rýchlejšia než LN