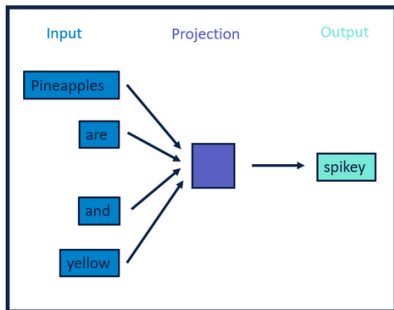


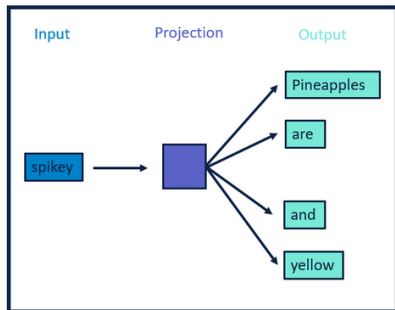
NLP

Wydział Biochemii, Biofizyki i Biotechnologii
Adrian Kania

- **word2vec** - technika wyznaczania embeddingów



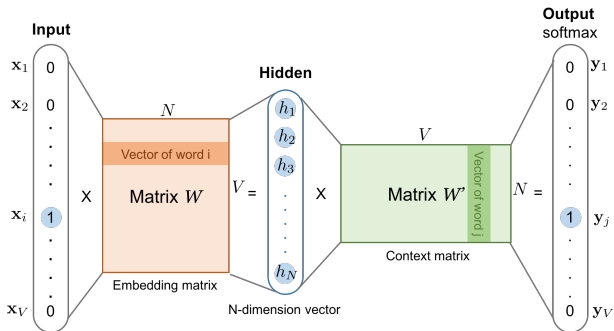
CBOW



Skip-gram

- $J_{CBOW} = -\frac{1}{T} \sum_{t=1}^T \log p(w_t | w_{t-n}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+n})$
- $J_{SG} = -\frac{1}{T} \sum_{t=1}^T (\log p(w_{t-n} | w_t) + \dots + \log p(w_{t-1} | w_t) + \log p(w_{t+1} | w_t) + \dots + \log p(w_{t+n} | w_t))$

SG = Skip-Gram



- wejście/wyjście - one hot encoding
- W - word embedding matrix
- W' - word context matrix
- Przykład: "The man who passes the sentence should swing the sword.--> ("swing", "sentence"), ("swing", "should"), ("swing", "the"), ("swing", "sword")

CBOW = Continuous Bag of Words

Input

0
.
1
.
.
0

X

0
.
1
.
.
0

X

0
.
.
1
.
0

X

N

Matrix W

Hidden

$V =$
avg

h_1
 h_2
 h_3
.
.
.
 h_N

X

V

Matrix W'

$N =$

Output
softmax

0 y_1
.
.
1 y_j
.
0 y_V

N-dimension vector
(**Average** of vectors of
all input words)

Skip-Gram - funkcja kosztu (Cross-Entropy)

$$p(w_O|w_I) = \frac{e^{v'_{w_O} v_{w_I}}}{\sum_i e^{v'_{w_i} v_{w_I}}}$$

- v_{w_I} - odpowiedni wiersz W (embedding vector) dla słowa wejściowego w_I ,
- v'_{w_i} - odpowiednia kolumna W' (context vector) dla słowa kontekstu w_i .

$$J = -\log p(w_O|w_I)$$

Skip-Gram - funkcja kosztu (Cross-Entropy)

$$J = -\log \frac{e^{v'_{wO} v_{wI}}}{\sum_i e^{v'_{wi} v_{wI}}} = -v'_{wO} v_{wI} + \log \sum_i e^{v'_{wi} v_{wI}}$$

Biorąc pod uwagę wszystkie słowa:

$$J = -\sum_{O,I} (-v'_{wO} v_{wI} + \log \sum_i e^{v'_{wi} v_{wI}})$$

ale... $\sum_i e^{v'_{wi} v_{wI}}$ kosztowne obliczeniowo ;(

Skip-Gram - funkcja kosztu (Cross-Entropy)

$$J = - \sum_{O,I} (-v'_{wO} v_{wI} + \log \sum_i e^{v'_{wi} v_{wI}})$$

ale... $\sum_i e^{v'_{wi} v_{wI}}$ kosztowne obliczeniowo ;(

Negative Sampling

$$J = -(\sum_{pos > O,I} \log \sigma(v'_{wO} v_{wI}) + \sum_{neg > O,I} \log \sigma(-v'_{wO} v_{wI}))$$

- każdy zestaw modyfikuje tylko mały procent wag (a nie wszystkie) - wybieramy niewielki podzbiór negatywnych słów i modyfikujemy ich wagi.

Różne kwestie

- Różne wagi dla słów z kontekstu (im dalsze słowo od celu tym mniejszy ma przyrządek).
- Uczenie fraz - np 'New york' nie jest tylko zlepkiem słów 'new' i 'york' lecz nazwą własną. Takie nazwy własne mogą być rozpoznane np przy użyciu bigramów. Definiujemy $s = \frac{C(w_i w_j)^{-\delta}}{C(w_i)C(w_j)}$, gdzie $C()$ - funkcja zliczająca, δ - pewien parametr.
- Podpróbkowanie częstych słów - prawdopodobieństwo pozostawienia słowa:

$$P(w_i) = \min\left(\left(\sqrt{\frac{f(w_i)}{0.001}} + 1\right) \cdot \frac{0.001}{f(w_i)}, 1\right)$$

gdzie $f(w_i)$ - częstość występowania słowa w_i w korpusie.

- Jak wylosować negatywne przykłady?

$$P(w_i) = \frac{f(w_i)}{\sum_j f(w_j)}$$

częściej jednak stosuje się $P(w_i) = \frac{f(w_i)^{3/4}}{\sum_j f(w_j)^{3/4}}$ (potęgowanie $\frac{3}{4}$ sprawia że wzrasta prawdopodobieństwo słów mniej częstych, a maleje tych najczęstszych (np. stop-words).

Podsumowanie:

- dwa podejścia: CBOW, Skip-Gram
- płytka sieć neuronowa (1 warstwa ukryta)
- embeddingi - wagi warstwy ukrytej

Wady:

- jeden wektor dla wyrazów wieloznacznych
- problem z OOV (out-of-vocabulary)

Dla dwóch słów w_i oraz w_j rozważamy prawdopodobieństwo warunkowe:

$$P(w_i|w_j) = \frac{C(w_i, w_j)}{C(w_i)}$$

Niech w_k będzie innym słowem. Jeżeli $\frac{P(w_k|w_i)}{P(w_k|w_j)}$ przyjmuje:

- duże wartości to w_k ma większy związek ze słowem w_i (niż w_j)
- wartość w przbliżeniu 1, to oznacza to, że w_k ma podobną relacje ze słowem w_i jak i w_j .

Idea:

Znaczenie słowa zawarte jest bardziej w stosunkach współwystępowania niż samych prawdopodobieństwach.

Założmy, że:

$$P(w_k|w_i) \sim e^{w_i^T w_k}.$$

Wtedy:

$$w_i^T w_k + b_k = \log P(w_k|w_i) = \log \frac{C(w_i, w_k)}{C(w_i)} = \log C(w_i, w_k) - \log C(w_i)$$

i dalej:

$$w_i^T w_k + b_k = \log C(w_i, w_k) - b_i$$

gdzie b_k związana jest z normalizacją prawdopodobieństwa, a $b_i = \log C(w_i)$. Celem jest zminimalizowanie

$$(w_i^T w_k + b_k + b_i - \log C(w_i, w_k))^2$$

Celem jest zminimalizowanie

$$(w_i^T w_k + b_k + b_i - \log C(w_i, w_k))^2$$

gdzie:

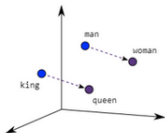
- w_i - word vector (embedding) dla i -tego słowa,
- w_k - context word vector dla k -tego słowa,
- b_i, b_k interpretowane są jako biasy dla w_i i w_k .

Dodatkowo często dokonuje się ważenia powyższego wyrażenia za pomocą $f(x)$ - zapobiega ona uczeniu tylko najczęściej występujących słów.

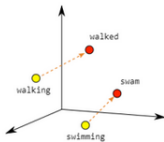
Typowo $f(x) = \min((\frac{c}{c_{max}})^\alpha, 1)$, gdzie α jest pewnym parametrem, a c_{max} - maksymalną wartością na c . Ostatecznie nasza funkcja kosztu wygląda następująco:

$$I = \sum_{i,k} f(C(w_i, w_k)) \cdot (w_i^T w_k + b_k + b_i - \log C(w_i, w_k))^2$$

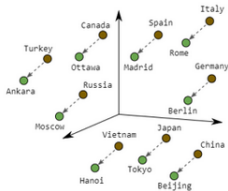
Word2Vec



Male-Female

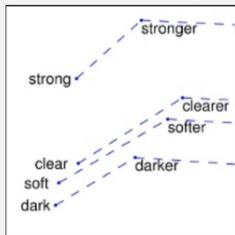
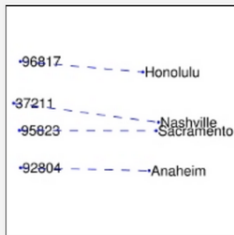
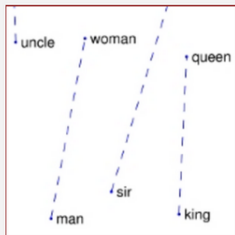


Verb Tense



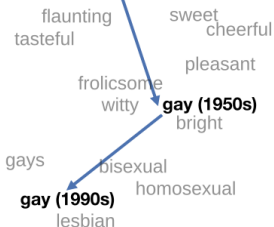
Country-Capital

GloVe



Śledzenie ewolucji znaczenia słów

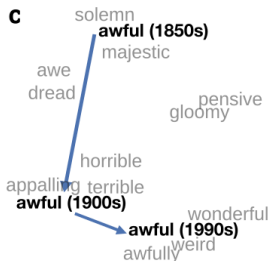
a gay (1900s)



b

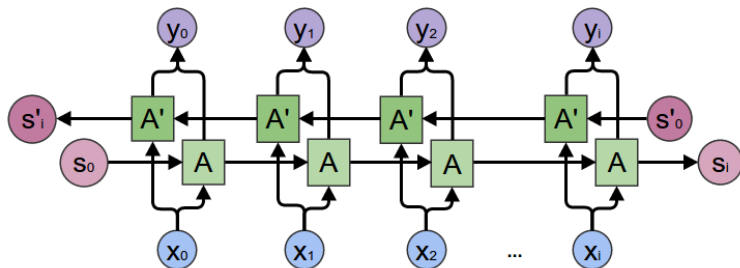


c



Inne modele

- ELMo (2018) - oparty o bi-LSTM



- BERT (2019) = Bidirectional Encoder Representations from Transformers

... offers an advantage over models like Word2Vec, because while each word has a fixed representation under Word2Vec regardless of the context within which the word appears, BERT produces word representations that are dynamically informed by the words around them.

Algorytmy do redukcji wymiarowości

Cel: Przekształcić dane znajdujące się w przestrzeni wysokowymiarowej $x_1, x_2, \dots, x_n \in R^N$ na przestrzeń niskowymiarową $y_1, y_2, \dots, y_n \in R^K$ ($N \gg K$). Czyli szukamy przekształcenia:

$$f : R^N \rightarrow R^K,$$

$$f(x_i) = y_i$$

dla $i = 1, \dots, n$. Przykładowe rozwiązania:

- PCA,
- SVD,
- Rzutowanie Sammona (MDS),
- t-SNE,
- inne...

Rzutowanie Sammona

- Idea: wykrycie charakterystycznych struktur danych w przestrzeni oryginalnej i przetransformowanie ich do przestrzeni zredukowanej przy najmniejszym zniekształceniu tych struktur.
- Jak? Minimalizując:

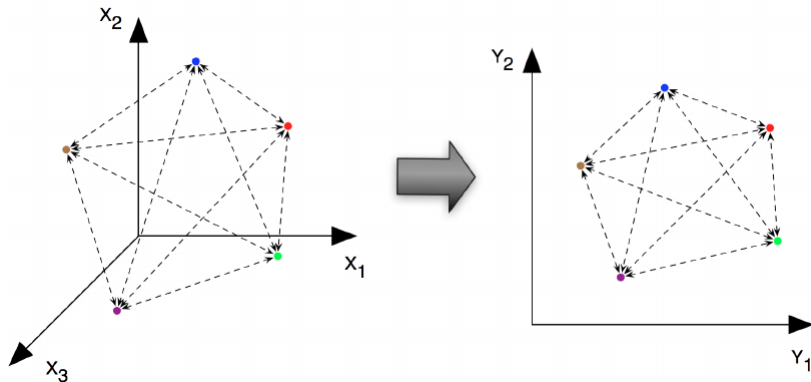
$$J = \frac{1}{\sum_{i < j} d_{ij}^*} \sum_{i < j} \frac{(d_{ij}^* - d_{ij})^2}{d_{ij}^*}$$

gdzie:

- d_{ij} - odległość pomiędzy punktami x_i oraz x_j (w oryginalnej przestrzeni),
- d_{ij}^* odległość pomiędzy punktami y_i oraz y_j (w zredukowanej przestrzeni).

Rzutowanie Sammona

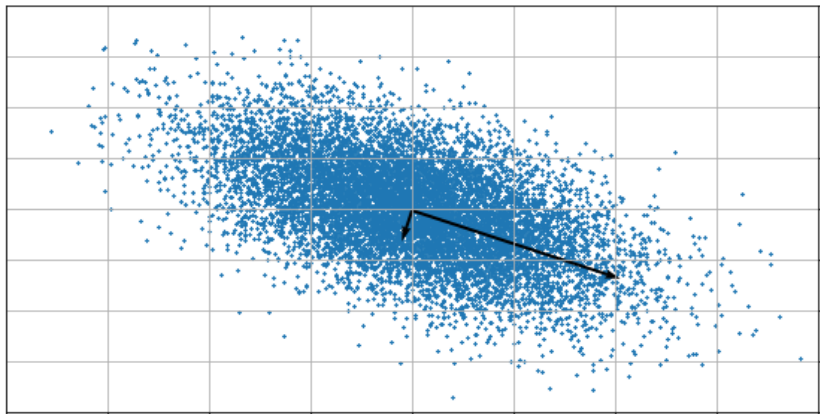
$$J = \frac{1}{\sum_{i < j} d_{ij}^*} \sum_{i < j} \frac{(d_{ij}^* - d_{ij})^2}{d_{ij}^*}$$



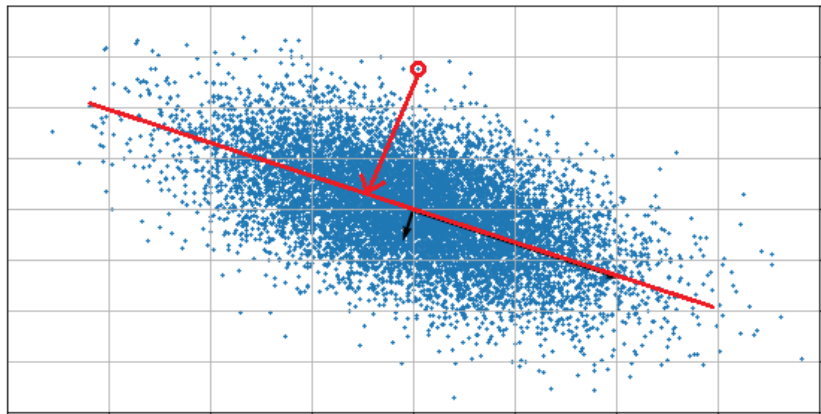
Sama idea podobna jak przy rzutowaniu Sammona - chcemy zachować relacje odległościowe pomiędzy punktami w oryginalnej i zredukowanej przestrzeni, tym razem używając **prawdopodobieństw warunkowych** - podobieństwo punktów x_i do x_j jest traktowane jako prawdopodobieństwo warunkowe, że punkt x_i jest najbliższym sąsiadem x_j (zakładamy Gaussowski rozkład prawdopodobieństwa). Podobnie dla przestrzeni zredukowanej z tym że tym razem wykorzystujemy rozkład t-Studenta (z powodu tzw. problemu zatłoczenia centrum). Rozkłady porównujemy za pomocą dywergencji Kullbacka - Leiblera.

- Idea: Rzutujemy dane na ten kierunek (kierunki) gdzie zmienność jest największa.
- Jak? Etapy:
 - 1 Wyznaczenie średnich dla wszystkich cech (wierszy).
 - 2 Wyznaczenie macierzy odchyleń.
 - 3 Wyznaczenie macierzy kowariancji.
 - 4 Wyznaczenie wartości własnych macierzy kowariancji.
 - 5 Wybieramy określoną przez nas liczbę największych wartości własnych (na moduł).
 - 6 Liczymy odpowiadające im wektory własne.
 - 7 Rzutujemy dane na wektory własne.

- Idea: Rzutujemy dane na ten kierunek (kierunki) gdzie zmienność jest największa.



- Idea: Rzutujemy dane na ten kierunek (kierunki) gdzie zmienność jest największa.

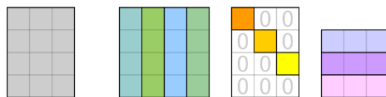


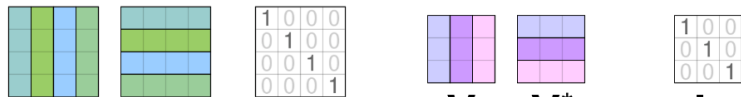
SVD

Każdą macierz o wyrazach rzeczywistych M można rozłożyć jako iloczyn trzech macierzy:

$$M = U \Sigma V^T$$

gdzie U , V - macierze ortogonalne, a Σ - macierz diagonalna.


$$M = U \Sigma V^*$$


$$U U^* = I_m \quad V V^* = I_n$$

Truncated - SVD

Każdą macierz o wyrazach rzeczywistych M można rozłożyć jako iloczyn trzech macierzy:

$$M = U\Sigma V^T$$

gdzie U, V - macierze ortogonalne, a Σ - macierz diagonalna. Macierz M' powstaje jako:

$$M' = U_t \Sigma_t V_t^T,$$

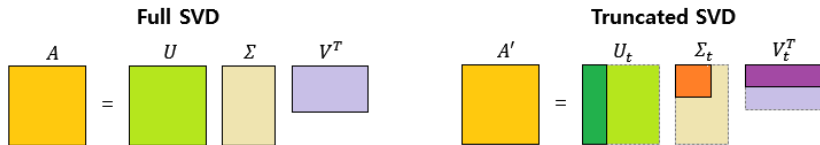
gdzie U_t/V_t powstały poprzez pozostawienie t kolumn/wierszy z macierzy U/V odpowiadających największym t wartościom z Σ .

Truncated - SVD

Macierz zredukowana M' powstaje jako:

$$M' = U_t \Sigma_t V_t^T,$$

gdzie U_t/V_t powstały poprzez pozostawienie t kolumn/wierszy z macierzy U/V odpowiadających największym t wartościom z Σ .



Korzyść: Załóżmy, że macierz M zawiera 10000 kolumn i 20000 wierszy - zatem łącznie $200000000 = 2 \cdot 10^8$ liczb. Zostawiając największych 100 wartości własnych, z satysfakcjonującym przybliżeniem możemy zachować informacje o tej macierzy przy pomocy $20000 \cdot 100 + 100 \cdot 100 + 10000 \cdot 100 \approx 3 \cdot 10^6$ liczb.