Distributed word embeddings

A szoftmax függvény

- Bináris osztályozás szigmoid függvénnyel $\sigma(z)=\frac{1}{1+\exp^{-z}}$
  - $z=w^T x$ mint normalizálatlan valószínűség
  - Alternatívan legyen $z_1=w_1^T x$ és $z_2=w_2^T x$ a pozitív és negatív osztályba tartozás normalizálatlan valószínűségei
  - $softmax(z)=\left[\frac{\exp^{z_1}}{\exp^{z_1}+\exp^{z_2}}, \frac{\exp^{z_2}}{\exp^{z_1}+\exp^{z_2}}\right]$
  - $z=z_1-z_2$ választással a logiszikus regresszió is ezt csinálja

Szoftmax példa

- A szoftmax függvény segítségével (-∞, ∞) intervallumból jövő értékeket szekvenciájából tudunk eloszlást gyártani
  - Osztályozást végző neurális hálók gyakori összetevője

[-2, 3, 0] → [0.14 20.09 1.00]

[-1, 4, 1] → [0.37 54.60 2.72]

[1, 2, 2.3] → [2.72 7.39 9.97]

Szoftmax példa

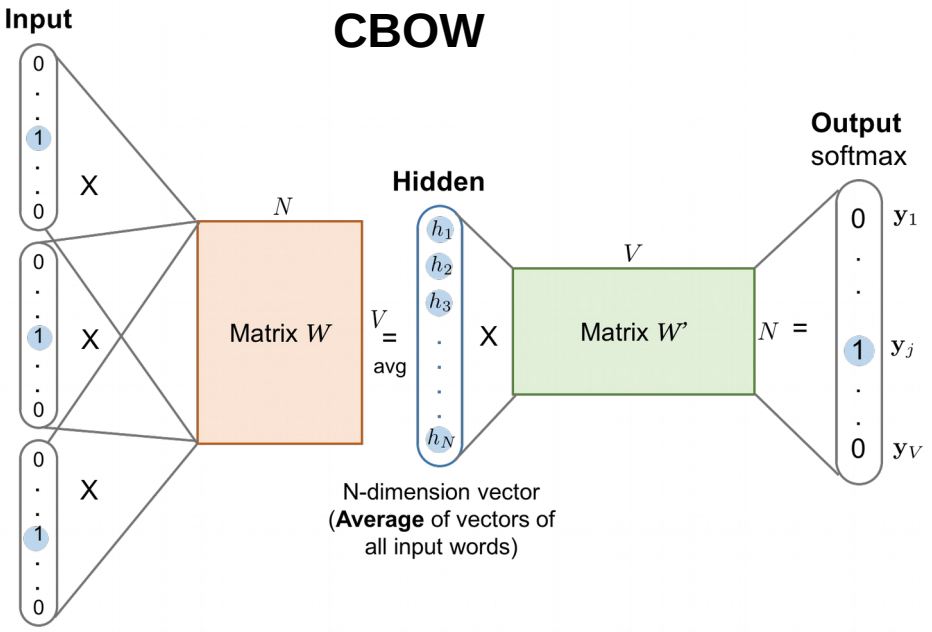
- A szoftmax függvény segítségével (-∞, ∞) intervallumból jövő értékeket szekvenciájából tudunk eloszlást gyártani
  - Osztályozást végző neurális hálók gyakori összetevője

[-2, 3, 0] → [0.14 20.09 1.00] → [0.006 0.946 0.048]
[-1, 4, 1] → [0.37 54.60 2.72] → [0.006 0.946 0.048]
[1, 2, 2.3] → [2.72 7.39 9.97] → [0.189 0.377 0.434]

word2vec (Mikolov et al., 2013)

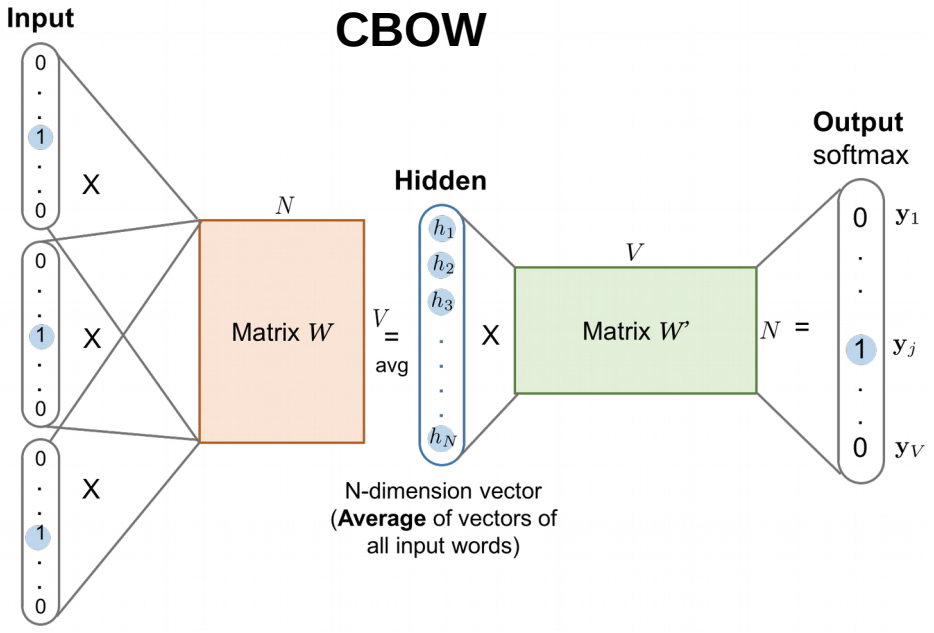
- Algoritmuscsalád több (kritikus) hiperparaméterrel
  - Alapcélja: olyan prediktív modellt tanulni, ami képes minél pontosabban megbecsülni, hogy ha egy szövegrészből kitakarunk egy/több szót, akkor mi/mik volt/voltak az/azok
- Minden szóhoz rendeljünk egy kontextus és output reprezentációt (egy-egy N dimenziós vektort)
  - Predikcióinkat a kontextus és output vektorok pontszorzatain alkalmazott szoftmax függvénnyel hozzuk meg

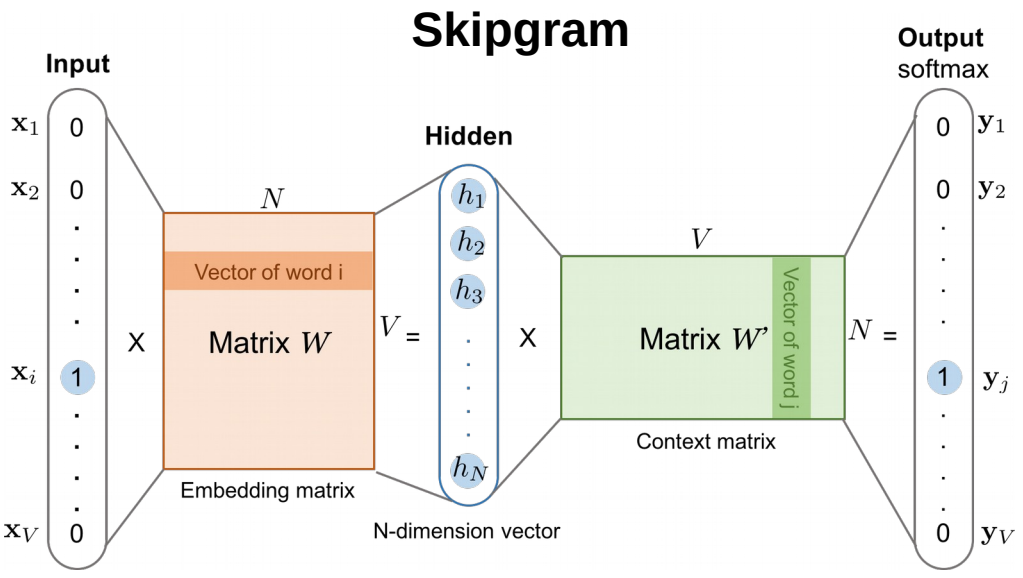
Continuous bag of words (CBOW) vs. Skipgram



Képek forrása: lilianweng.github.io

Continuous bag of words (CBOW) vs. Skipgram





Képek forrása: lilianweng.github.io

A predikciós mechanizmus

- Egy-egy kontextusablak viszonyában regisztráljuk a predikció kapcsán jelentkező hibát, és frissítsük a szóreprezentációkat

$$p(o_i|c_j) = softmax(\mathbf{1}_j^T W W')$$

  - $\mathbf{1}_j$ = [0 0 0 … 0 0 1 0 0 … 0] alakú ún. one-hot vektor

  - W és W’ paraméterek függetlenek egymástól (Miért?)
  - Kezdetben random értékekeket tartalmaznak, SGD-vel frissítjük őket a tanulás során

A frissítési szabály

- Szükségünk van a predikció hibájának gradiensére
  - A predikciós hiba az elvárt szó előrejelzésének negált log valószínűsége

$$\ell = -\log\left(p\left(o_i|c_j\right)\right) = -\log\frac{e^{w_j^T w_i^{'}}}{\sum_{k=1}^{|V|} e^{w_j^T w_k^{'}}} = ?$$

  - Mi lesz a hibatag gradiense? Hogy lehet értelmezni?

Az elvi modell problémái

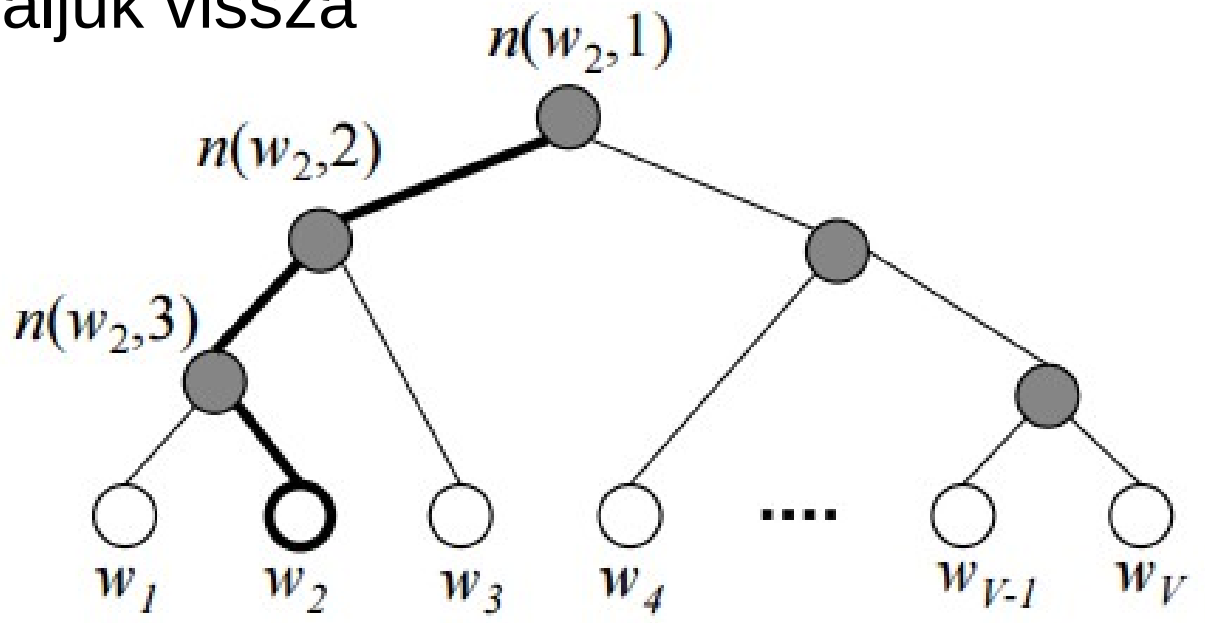
- A célfüggvény gradiensét nagyon költséges kiszámolni
  - Egyetlen frissítés alkalmával a teljes szótár (V) fölötti összegzést igényel
- A probléma javítható hierarchikus szoftmax vagy negatív mintavételezés alkalmazásával

Hierarchikus szoftmax

- A predikciós mechanizmust (W') cseréljük le egy bináris fára
  - N csúcsú bináris fa várható magassága log(N) 👍
  - A bináris fa minden csúcsa egy-egy döntést hoz
  - Egy outputra vonatkozó predikció legyen a fában hozzá való eljutás során hozott döntések valószínűségeinek szorzata
- 1 db |V| kimenetelű multinomiális eloszlás helyett hozzunk log(|V|) bináris döntést

Hierarchikus szoftmax illusztrációja

- Minden csúcs egy-egy mini osztályozó
  - Balra vagy jobbra tovább?
- A hibát így propabáljuk vissza



Negatív mintavételezés

- A teljes szótár feletti predikció helyett hozzunk *néhány* egyszerű bináris döntést (valid/invalid kontextus)
  - Bináris döntést hozni sokkal olcsóbb, mint egy |V| kimenetelest

$$\ell = -\log\left(p\left(Y=1|o_i,c_j\right)\right) - \sum_{k=1,o_k\sim Q}^{K} \log\left(p\left(Y=0|o_k,c_j\right)\right)$$

  - Q a szótár elemei feletti gyakorisági eloszlás

Negatív mintavételezés

- A teljes szótár feletti predikció helyett hozzunk *néhány* egyszerű bináris döntést (valid/invalid kontextus)
  - Bináris döntést hozni sokkal olcsóbb, mint egy |V| kimenetelest

$$\ell = -\log\left(p\left(Y=1|o_i,c_j\right)\right) - \sum_{k=1,o_k\sim Q}^{K} \log\left(p\left(Y=0|o_k,c_j\right)\right)$$

  - Q a szótár elemei feletti gyakorisági eloszlás
    - Q meghatározása során a szavak gyakoriságát emeljük egy 1-nél kisebb hatványra (pl. 0.75) → Miért jó ötlet ez?

További trükkök

- Adaptív ablakméret (távolabbi szomszédok alulsúlyozása)
- Gyakori szavak alulmintavételezése 1-sqrt(t/f(w)) valószínűséggel
  - t egy hiperparaméter
  - f(w) a szó gyakorisága
- Gyakori mintázatok (pl. *New_York*) beazonosítása
  - Történhet pl. PPMI segítségével
- Stb, stb…