Bengio, Ducharme, Vincent: A Neural Probabilistic Language Model

Arto Teräs ⟨ajt@iki.fi⟩

31.10.2001

Tavoite

- Laskea aiemman materiaalin perusteella seuraavan sanan todennäköisyys

Pääidea

- Käytetään neuroverkkoa, jolle opetetaan yhtä aikaa sekä sanasekvenssit että kunkin sanan ominaisuudet

⇒ Pystytään arvioimaan todennäköisyyttä myös silloin, kun sanasekvenssiä ei ole aiemmin esiintynyt

Yleinen tilastollinen kielimalli

- Ilmaistaan sanan todennäköisyys kaikkien aiemmin havaittujen sanojen perusteella
- $P(w_1^T) = \prod_{t=1}^{T} P(w_t \mid w_1^{t-1})$, jossa w_t on t:s sana ja w_1^j sanasekvenssi $(w_1, w_2, ..., w_j)$

Ongelmat

- Sanojen yhteisjakauma (joint distribution) kasvaa eksponentiaalisesti huomioitaessa useampia peräkkäisiä sanoja
  - Kompleksisuus $O(V^n)$, jossa V sanaston koko ja n peräkkäisten sanojen määrä
    Esimerkiksi 100000 sanan sanasto ja 10 peräkkäistä sanaa: enimmillään $100000^{10} - 1 = 10^{50} - 1$ vapaata parametria
- Mitä useampia sanoja huomioidaan, sitä harvemmin aiemmin on havaittu täsmälleen sama sekvenssi

Perinteiset ratkaisut

- N-gram -mallit: Tilastoidaan n peräkkäisen sanan yhdistelmät, jossa n on pieni
- Lasketaan todennäköisyys perustuen aiemmin kohdattujen samojen yhdistelmien lukumäärään
- Kohdattaessa outo sanasekvenssi muodostetaan todennäköisyys lyhyemmän kontekstin perusteella

Samankaltaiset sanat

- Samankaltaiset sanat esiintyvät lauseissa samoissa rooleissa

⇒ Lauseen `The cat is walking in the bedroom` esiintyminen opetusmateriaalissa pitäisi lisätä lauseen `A dog was running in a room` todennäköisyyttä, vaikka jälkimmäistä ei olisikaan aiemmin havaittu

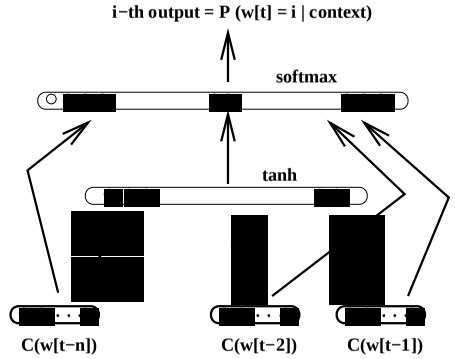
Sanojen piirrevektorit

- Liitetään jokaiseen sanaan piirrevektori $C(w_n)$, joka kertoo sen ominaisuudet
- Ominaisuuksien lukumäärä huomattavasti pienempi kuin sanaston koko, esimerkiksi 30-100 kpl
- Samankaltaisilla sanoilla lähekkäiset piirrevektorit
- Vektorit voidaan alustaa satunnaisiksi ja antaa neuroverkon hakea arvot oppimisprosessin aikana

Opetusvaihe

- Tavoitteena oppia hyvä malli $f(w_t, ..., w_{t-n}) = \hat{P}(w_t \mid w_1^{t-1})$
- Annetaan syötteeksi kerrallaan n sanan piirrevektorit
- Määritellään sanasekvenssien todennäköisyysfunktio, jonka muuttujina ovat nämä piirrevektorit
- Verkko sovittaa samanaikaisesti vektorien arvoja ja todennäköisyysfunktiota
- Kaksi arkkitehtuuria, jotka poikkeavat toisistaan hieman sekvenssin viimeisimmän sanan käsittelyn osalta

Mallin kaaviokuva (suora arkkitehtuuri)



Toiminnan pääperiaate

- Kun todennäköisyysfunktiota (log-likelihood) maksimoidaan, samankaltaisissa yhteyksissä esiintyvät sanat saavat lähellä toisiaan olevat piirrevektorit
- Piirteiden ollessa lähellä toisiaan uusi havaittu lause X ei lisää pelkästään saman sanasekvenssin todennäköisyyttä vaan myös kaikkien niiden, jotka ovat lähellä X:ää lauseiden piirreavaruudessa
- Esimerkki: `Kissa käveli katolla`
  `Koira käveli kadulla` (++)
  `Ulkona oli jänis` (+)
  `Kauris siili leijona` (−)

Tuloksia

- Mallia testattiin noin miljoonan sanan (Brown) ja 34 miljoonan sanan (Hansard) aineistoilla
- Noin 20-30 % parempia tuloksia smoothed trigram -malliin verrattuna perplexity-arvoilla mitattuna
- Parhaat tulokset saavutettiin yhdistämällä malli trigram-mallin kanssa

Havaintoja

- Varottava ylisovitusta (over-fitting) pienillä opetusaineistoilla
- Laskentatarkkuus tärkeää suurilla aineistoilla
- Laajemman kontekstin huomiointi parantaa tuloksia noin 10 sanaan asti
- Piirrevektorien alustuksella ei suurta merkitystä vertailtaessa täysin satunnaista alustusta ja erään sanojen yleisyyteen perustuvan algoritmin antamia arvoja
- Piirrevektorien arvojen samanaikainen sovitus yhtä aikaa muiden parametrien kanssa tärkeää

Nopeutuskeinoja

- Listat kaikkein todennäköisimmistä sanoista
- Useimmin esiintyvien sekvenssien esilaskenta
- Puheentunnistuksessa riittää, että tutkitaan akustisesti samankaltaisia sanoja
- Tutkittavan tekstin kappaleiden permutointi voi nopeuttaa sovitusta

Avoimia kysymyksiä

- Ominaisuusvektorien määrittely eri kokoisille yksiköille (parin sanan ryhmät, sanojen osat)
- Erityyppisten neuroverkkojen ja oppimisalgoritmin vaikutus
- Mitä voidaan päätellä tutkimalla millaiset piirrevektorit ovat muodostuneet oppimisprosessin seurauksena
- Muulla tavoin hankitun semanttisen, syntaktisen ja morfologisen informaation hyödyntäminen
- Suorituskyky käytännön sovelluksissa

Kotitehtäväkysymykset

1. Esitetyssä mallissa huomioidaan tavallista suurempi määrä edeltäviä sanoja uuden sanan todennäköisyyttä laskettaessa. Mitä tekniikoita käytetään, jotta vapaiden parametrien määrä ja laskenta-aika pysyvät kohtuullisina?
2. Miten sanoihin liittyvät piirrevektorit (feature vectors) muodostetaan? Miten vektorien valintaa voitaisiin mahdollisesti parantaa?