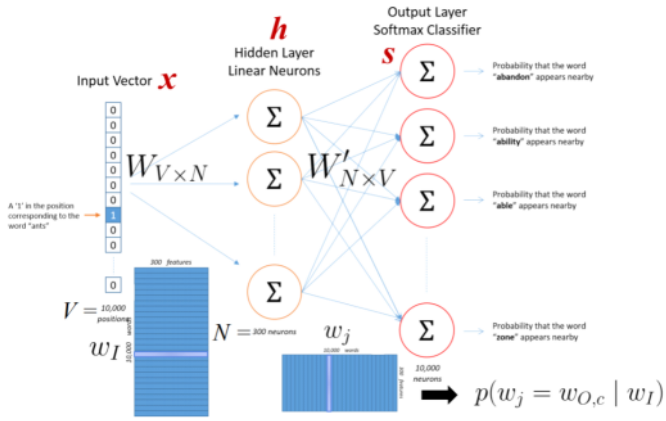
ADL—Word Embeddings

2019年3月26日　　上午 09:22

- Atmoic symbols: one-hot
- Neighbors: 相似性向量表示法
  - co-occurrence matrix
  - word2vec (直接去學)
  - GloVe (直接去學)

- word embedding
  - 空間直接找一個點代表這個word，遷入在space上面
  - 可以把target task接在一起，整個參數可以留到後面
  - 後面在update的時候可以讓前面的word embedding可以更新
  - 可以根據final task fine tune，隨著後面層的update做更新
- Word2Vec Skip-Gram
  - 比CBOW跟LM還要好
  - 想要利用neighbor，根據target word predict周圍的字詞們
  - 希望模型學到字跟字之間的關係，希望給定中間的字，context window中的word的probability最高
  - 學到所有字跟neighbor的關係，每個word都會被當成central ward，考慮所有可能的target word
  - 如果有新的句子進來的話只要更新一下就好了
  - 給定target word可以預測context word
  - output: 這個輸入字跟輸出字的對應機率是多少，希望輸出有出線的是1，其他是0
  - predict context word，中間hidden layer是linear的
  - 中間hidden layer的維度就是word embedding的維度
    - 每一個字都會對應300 維度的feature，所以可以拿第一個weight matrix出來當成一萬個字的vector
    - 看輸入的one-hot是地幾個維度，就等於是拿第一個weight matrix的第幾個row，當成vector，因為那一條row的參數可以很好表示出他的context word
  - output的vector不重要，hidded layer有能力表示target word才代表他是重要的
  - W' 的weight matrix 則可以算出分數、得到機率，最後透過softmax normalization，得到context word vector
  - W' 就是那個對應維度的字，當成context word時候的embedding
  - 每一個字都會有兩個embedding，一個是從前面matrix來的，另一個是從後面來的
  - 所以整個訓練好: 前面matrix是當成target word時候的embedding，後面matrix是當成context word所得到的embedding，最後輸出分數，透過softmax變成機率分布
  - given wI輸出是wj的機率是多少
  - 

  - 只會去update 在context window的那些字，才會對模型參數有影響
  - 當vocabulary (training data)很大的時候，會需要很久的時間才可以update好左邊的matrix
    - 每次不要update所有，每次training都只update部分
    - 解1: h-softmax=>前期再用，後期只會用negative sampling
      - 限制只去update root->node的路徑上機率 (罕用)
    - 解2: negative sampling

        - 原本要除上所有人的總和，只除上sample出來那些人的總和
        - 每個字都有同樣機率被sample過，假設每個字被當成negative的機率相同，且都會sample到，每次都只要算一部份的人就可以算出後面的weight進行update。
        - 只要計算被sample出來的那些字就好了，只算部分的字就可以算出數值
        - {context word}U{negative sampling}，sample當成negative的人，但是context word也要考慮進來
        - sampling method
            - random sample。有足夠水準了
            - distribution sampling: 依據不同分布得到不同被sample到的機率。ex: 第一個字被sample到的機率要是第三字的三倍
                - P不好選。less frequent word sampled often
                - 字比較多的比較容易被sample到，但因為他已經出現夠多次了被update出現次數比較多，所以比較爛的embedding是那些出現次數比較少次的字。希望少看到的字可以update多一點，字多的不用刻意去update了
                - empirical setting = 3/4。少出現的字要上升多一點次方。可以比random sample好一點
    - CBOW: given context predict target
    - LM: predict下一個字。原本是用RNN來做，後來發現skip-gram效果比較好

- Co-occurense Matrix(count-base) vs. Direct prediction (NN)
    - 前者train很快，統計概念去計算
    - 後者是找出vector希望有一些性質
    - 前者不一定有字跟字之間的關係，且不易增加新句子
    - 捕捉字跟字之間關係
    - data大的時候才用NN比較好
    - 統計性質可能少資料會比較好
    - 因此GloVe希望可以結合兩者

- GloVe
    - 想用embedding直接學，但又希望可以透過co-occurrence matrix方式當成objective
    - 希望學embedding，可以具有統計上的性質，兩者relation可以變成字的ratio
    - 利用square error update，學完的vector就會有統計上的性質
    - embdding learning學到有統計性質的representation
    - performance很好，效率很快，corpus小仍具有performance會利用統計性質，當有大data的時候可以用skip-gram

- Evaluation word embedding評估向量好不好
    - intrinsic evaluation直接衡量embedding本身好不好
        - 去看字跟字之間linear的relationship
        - A:B=C:X? 看對應的X是否正確，去找最接近的向量出來是否有跟預想的一樣對應
        - semantic，但是有些evaluation會有個問題，像是首都跟國家的關係，但如果首都更換動態更改的話GG
        - syntatic，形容詞跟最高級、過去式現在式
        - word correlation: 先請人來標註字跟字之間的關係，人覺得的跟機器覺得的是否相近，人給定一個相近程度分數。但有可能一字多義，或是詞性不同
    - Extrinsic Evaluation間接方式，看的是對task有否幫助，embedding好不一定performace好
        - 希望好的embedding在最終的performance可以比較好
        - 有進步代表embedding 有給一些概念
        - 可以把task資訊借用，例如給sentiment analysis
    - 大家會把各種evaluation都嘗試看看，希望各種結果都可以變好，如果只有一個遍高恐怕未必embedding比較好

結論:

- 希望學到low dimentional word vector: skip-gram, cbow, LM
- GloVe希望學到機率ratio代表一些meaning