10-707: Advanced Deep Learning **Spring 2020**

Lecture 20: April 13

Lecturer: Andrej Risteski *Scribes: Wen Wen,Gautam Jain, Ziyi Dou, Tianqing Zhang*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications, if as reader, you find an issue you are encouraged to clarify it on Piazza. They may be distributed outside this class only with the permission of the Instructor.*

This lecture's notes illustrate some uses of various LaTeX macros. Take a look at this and imitate.

20.1 Word Embeddings

- Word embeddings is a semantically meaning vector representation of words.
- Inner product could correlate the similarity among words
- Sentiment classification can be done by word embeddings through simple linear classification (e.g. from the comment of the customer, understand their feedback on products)
- Word embeddings can be used to train a network which can translate between languages

20.1.1 Word embeddings via predictive learning

There generally two kinds of tasks:

- Basic task: predict the next word, given a few previous ones
- Related task: predict middle word in a sentence, given surrounding ones.

Basic task:

The setting in basic task is to predict the next word based on previous words. Inspired by classical assumptions in NLP that the underlying distribution is Markov, which states that x_t only depends on the previous few words, the objective being optimized is:

$$\max_{\theta} \sum_{t} \log p_\theta \left(x_t | x_{t-1}, x_{t-2}, ..., x_{t-L}\right)$$

Note the situation may be violated if the target is long texts like paragraphs or books. The main problem of this setting is the parametrization of $p_\theta \left(x_t | x_{t-1}, x_{t-2}, ..., x_{t-L}\right)$. The trivial parametrization would involve a "lookup" table with V^L entries, where V denotes the number of vocabulary, L denotes the number of previous words. This table is unfeasible since it grows exponentially with the number of vocabulary.

A solution proposed by [Bengio-Ducharme-Vincent-Janvin '03] is to use neural parametrization of the above probabilities:

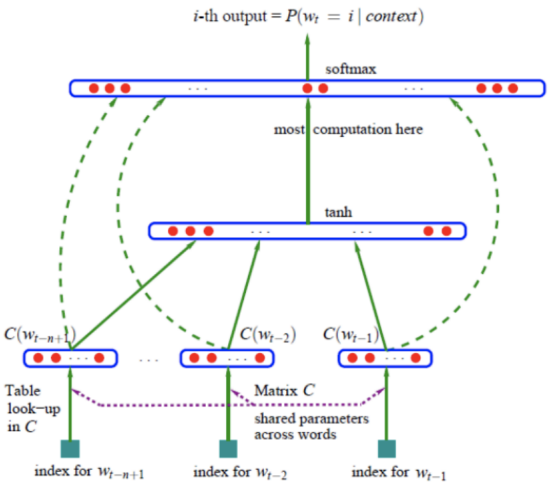

Fig 1. high-level architecture of neural parametrization

Figure above shows the high-level architecture. This model has an embedding matrix that assigns vector (word embeddings) for each of the words. Taking non-linear transformation of the embeddings as well as some positional index, it outputs a vector o. Finally, the softmax will produce a distribution by taking vector o as input.

Related task:

The setting here is to predict the middle word in a sentence, given surrounding ones:

$$\max_{\theta} \sum_{t} \log p_{\theta} \left(x_t | x_{t-L}, ..., x_{t-1}, x_{t+1}, ..., x_{t+L} \right)$$

A solution to this problem is CBOW (Continuous Bag of Words) proposed by Mikolov et al.'13:

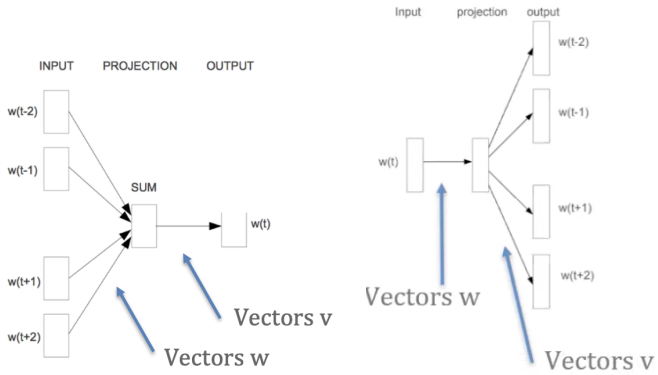

Fig.2. parametrization of CBOW (left) and Skip-Gram (right)

On the left shows CBOW. The problem is parametrizied by two vector, v and w. The conditional probability we are trying to optimized will be proportional to the exponential of the vector v of the word we try to predict, and the average of the vector w of the surrounding words:

$$p_\theta\left(x_t|x_{t-L},...,x_{t-1},x_{t+1},...,x_{t+L}\right) \propto \exp\left(v_{x_t}, \sum_{i=t-L}^{t+L} w_{t_i}\right)$$

Another example of word-to-vector is Skip-Gram, in which the problem becomes predicting the surrounding words given the middle word:

$$\max_\theta \sum_t \sum_{i=t-L, i\neq t}^{t+L} \log p_\theta\left(x_i|x_t\right)$$

in which $p_\theta\left(x_i|x_t\right) \propto \exp\left(v_{x_i}, w_{x_t}\right)$. In practice, the bottleneck is the softmax distribution, where the partition function sums over the entire vocabulary. Common ways to solve this are negative sampling, hierarchical softmax. In general, given the two distinct parametrization, CBOW and Skip-Gram, the advantages of each kind is not clear.

20.2 Semantic Similarity

- Similar words tend to have larger (renormalized) inner products (also called cosine similarity).
- Precisely, if we look at the word embeddings for words i, j

  $$\langle \frac{w_i}{||w_i||}, \frac{w_j}{||w_j||} \rangle = \cos(w_i, w_j)$$

  tends to be larger for similar words i, j.
- To solve semantic similarity query like "which is the most similar word to", output the word with the highest cosine similarity.

20.2.1 Explaining Similarities

Part 1: The distributional hypothesis of language: This is the notion that the word is defined by the company it keeps, basically the words it co-occurs with. We can use this idea to define word embeddings. Let

$$M_{w,c} = Pr(\text{words w, c} \mid \text{co-occur in a window of size 5})$$

Each word w can be represented using the corresponding row in M [HAL; Lund & Burgess'96]. The word embedding derived using the method follows similarity if checked empirically. But the word embedding can be huge.

Follow up idea: *Find a "low-dimensional" approximation* In order to do this we find a matrix $\tilde{M} \in \mathbb{R}^{N\times k}$ s.t $MM^T \approx \tilde{M}\tilde{M}^T$ where N is the size of the vocabulary and k is the word vector size. This ensures that the distance betweent the word embeddings remains the same.
An intuitive way to achieve this is by doing SVD:
Rank-k SVD of a matrix M gives matrices $U_k, V_k \in \mathbb{R}^{N\times k}, \Sigma_k \in \mathbb{R}^{k\times k}$ s.t. $||M - U_k\Sigma_k V_k^T||_F$ is minimized.
You can also take the square root of each entry of co-occurence matrix ("range-squishing").

Part 2: SkipGram/CBOW : Rewriting the SkipGram objective, we have:

$$\max_{\theta} \sum_{t} \sum_{i=t-L, i\neq t}^{t+L} \log p_\theta\left(x_i | x_t\right)$$

$$= \max_{\theta} \sum_{i,j} f_{i,j} \log(\frac{\exp(\langle v_i, w_j \rangle)}{\sum_k \exp(\langle v_i, w_k \rangle)})$$

where $f_{i,j}$ is number of times word i appears within distance 5 (or any arbitrary distance) of word j in corpus. In a way we can see that the expression is trying to maximize inner products for v_i, w_j or words i, j that tend to co-occur more!

Levy-Goldberg showed that a natural modification of the SkipGram objective (SGNS) results in word vectors satisfying.

$$\langle v_i, w_j \rangle \approx PMI(i,j) = \frac{\log P(i,j)}{\log P(i) \log P(j)}$$

The numerator represents the (normalized) frequency of seeing i,j in a local window. The denominator represents pointwise mututal information (individual frequencies). They also show above correspondence holds provably, if word vectors are allowed to be very high dimensional (dimension of vocabulary or more). The word embeddings learned today are only of size approximately equal to 300. The next section talks about how to find word embeddings that work well and are not order of N.

Generative model for language with semantics
The corpus is generated sequentially with each the t^{th} word being produced at time t. The process is driven by a "discourse vector" doing a random walk in a "discourse space" of size d ¡¡ N (N is vocabulary size). Each word has a (time invariant) latent vector v_w and if discourse vector at time t is c_t. The probability of emitting word w at time t is,

$$Pr(w) \propto \exp\left(v_w \cdot c_t\right)$$

Setup of the parameters and the assumptions :

- Discourse Space: It is a unit sphere and the random walk is slow i.e

  $$||c_t - c_{t-1}||_1 << \frac{1}{\sqrt{d}}$$

  Moreover, stationary distribution of random walk is uniform over unit sphere (denoted as C) , and $c_t \sim C$ for all t (i.e. walk is started from stationary distribution)

- Latent Vectors: v_w are spatially isotropic and are drawn from scaled normal distribution with zero mean and identity variance. The scalar which contributes to the scaling is a random variable with expectation constant and with probability 1 it is bounded by some constant. This helps in keeping $||v_w|| = \sqrt{d}$ with high probability.

Co-occurrences capture PMI (Mutual Information) : In the following theorem, we will show that such a model can approximate PMI using low dimensional word embeddings. Further we will see that the norm of word vector represents the frequency and the spatial orientation contains the meaning. The presence of scaling allows the probability of word to vary as it would in different corpus.

Theorem 20.1 *Let $p(w, w')$ be the probability that w, w' appear consecutively. We can show,*

$$\log(p(w, w')) = \frac{1}{2d}||v_w + v_{w'}||^2 - 2\log Z \pm \epsilon$$

$$\log(p(w)) = \frac{1}{2d}||v_w||_2^2 - \log Z \pm \epsilon$$
$$\log(PMI(w, w')) = \frac{1}{2d}\langle v_w, v_{w'}\rangle \pm 3 \times \epsilon$$

Proof: First, by marginalizing, we get

$$\begin{aligned} p(w, w') &= \int_{c,c'} p(w|c)p(w'|c')p(c, c')dcdc' \\ &= \int_{c,c'} \frac{\exp(< v_w, c >)}{Z_c} \frac{\exp(< v_{w'}, c' >)}{Z_{c'}} p(c)p(c'|c)dcdc', \end{aligned}$$

where c, c' are successive context vectors.

Recall that Z_c is constant within $(1 \pm o(1))$ for almost all c with high probability, which means $(1 - o(1))Z \leq Z_c \leq (1 + o(1))Z$.

If context drifts slowly, we can treat $c \approx c'$, *i.e.*

$$\int_{c'} \frac{\exp(< v_{w'}, c' >)}{Z_{c'}} p(c)p(c'|c)dc' = (1 \pm o(1)) \exp(< v_{w'}, c >).$$

Therefore,

$$\begin{aligned} p(w, w') &= \int_{c,c'} \frac{\exp(< v_w, c >)}{Z_c} \frac{\exp(< v_{w'}, c' >)}{Z_{c'}} p(c)p(c'|c)dcdc' \\ &= \frac{(1 \pm o(1))^2}{Z^2} \int_c \exp(< v_w + v'_w, c >)p(c).dc \end{aligned}$$

Because $p(c)$ is a product distribution, the above expression factors:

$$\int_c \exp(< v_w + v'_w, c >)p(c)dc = \mathbb{E}_c \exp(< v_w + v'_w, c >) = \prod_{i=1}^{d} \mathbb{E}_{c_i} \exp(v_w + v'_w)_i c_i.$$

Distribution of c_i is approximately Gaussian with variance $1/d$, and for X a Gaussian with variance $1/d$, we have

$$\mathbb{E}[\exp(aX)] = \frac{\sqrt{d}}{\sqrt{2\pi}} \int_x e^{ax} e^{-dx^2} dx = e^{\frac{a^2}{4d}}$$

Therefore,

$$\int_c \exp(< v_w + v'_w, c >)p(c)dc = \prod_{i=1}^{d} \mathbb{E}_{c_i} \exp(v_w + v'_w)_i c_i \approx \exp(\frac{1}{4d}||v_w + v'_w||^2).$$

And thus

$$\log(p(w, w')) = \frac{1}{2d}||v_w + v_{w'}||^2 - 2\log Z \pm \epsilon.$$

■

Making a connection to CBOW:

Consider a simplified version of CBOW, where we first sample a discourse $c \sim C$, then sample the window $(x_{t-L}, \cdots, x_{t+L}) \sim \prod_{i=t-L}^{t+L} \exp(< v_{x_i}, c >)/Z_c = \frac{\exp(<\sum_i v_{x_i}, c>}{Z_c^L}$.

Assuming $Z_c = Z$ for simplicity as well, then the maximum likelihood estimate for c is:

$$\begin{aligned}\max_c p(c|x_{t-L}, \cdots, x_{t+L}) &= \max_c p(c, x_{t-L}, \cdots, x_{t+L}) \text{ (Bayes rule)} \\ &= \max_c p(x_{t-L}, \cdots, x_{t+L}|c) \text{ } (p(c) \text{ is uniform}) \\ &= \max_c \exp(< \sum_i v_{x_i}, c >) \\ &= \frac{\sum_i v_{x_i}}{||\sum_i v_{x_i}||}\end{aligned}$$

Hence, CBOW effectively tries to estimate the discourse, and maximize the probability of v_{x_t} given this estimate.

20.2.2 Analogies

Observations: We can solve *analogy* queries by linear algebra. Precisely, $w =$ queen will be the solution to:

$$\text{argmin}_w ||v_w - v_{king} - (v_{woman} - v_{man})||^2.$$

Levy & Goldberg'14 propose that the solution w should satisfy

$$\frac{P(\mathcal{X}|man)}{P(\mathcal{X}|woman)} \approx \frac{P(\mathcal{X}|king)}{P(\mathcal{X}|W)},$$

for most words c.

Therefore, to solve analogy, we can find word w that minimizes

$$\sum_{\mathcal{X}} (\log \frac{P(\mathcal{X}|man)}{P(\mathcal{X}|woman)} - \log \frac{P(\mathcal{X}|king)}{P(\mathcal{X}|W)})^2.$$

Implicitly, the ratio $\frac{P(\mathcal{X}|a)}{P(\mathcal{X}|b)}$ is a function of the semantic relation R that,

$$\frac{P(\mathcal{X}|a)}{P(\mathcal{X}|b)} = \nu_R(\mathcal{X}) \xi_{a,b,R}(\mathcal{X})$$

We take the log of the equation above and get,

$$\log(P(\mathcal{X}|a)) - \log(P(\mathcal{X}|b)) = \log(\nu_R(\mathcal{X})) + \tilde{n}_{a,b,R}(\mathcal{X})$$

.

The second term on the right hand side is the noise to the true semantic relation. In order to reduce the noise, we want to go to a lower dimension for the vectors.

Let V be matrix with rows v_x for all words $\mathcal{X}$. Let $\epsilon_{a,b}$ be the error in PMI approximation from the main theorem for generative model. By main Theorem, we have,

$$\log(P(\mathcal{X}|a)) - \log(P(\mathcal{X}|b)) = \frac{1}{d} < v_{\mathcal{X}, v_a - v_b} > + \epsilon_{a,b}(\mathcal{X}) = \frac{1}{d} V(v_a - v_b) + \epsilon_{a,b}(\mathcal{X})$$

Rewriting, we will get,

$$V(v_a - v_b) = d\tilde{\mu_R} + d(\tilde{n}_{a,b,R}(\mathcal{X}) - \epsilon_{a,b}(\mathcal{X}))$$

.

Then we can apply a linear regression with design matrix V and solve for $v_a - v_b$. We will get $v_a - v_b = V^+(d\tilde{\mu_R} + d(\tilde{n}_{a,b,R}(\mathcal{X}) - \epsilon_{a,b}(\mathcal{X})))$. If we have V that is well-conditioned, we can denoise a lot. For example, if V has entries that are random Gaussian with variance $\frac{1}{N}$, $\sigma_{min}(V) = \Omega(\sqrt{\frac{N}{d}})$, then ,

$$||V^+ d(\tilde{n}_{a,b,R}(\mathcal{X}) - \epsilon_{a,b}(\mathcal{X}))||_2 \leq d\sqrt{\frac{d}{N}}||\tilde{n}_{a,b,R}(\mathcal{X}) - \epsilon_{a,b}(\mathcal{X})||_2$$

.

If $d \ll N$, this will drive the error to be very small. In other word, the small dimension and reduce the noise in the semantic relation. By denoting $\mu_R = V^+ d\tilde{\mu_R}$ and $n_{a,b,R} = n_a, \tilde{b}, R - \epsilon_{a,b}$, we get

$$v_a - v_b = \mu_R + n_{a,b,R}$$

with $||n_{a,b,R}|| \ll ||\tilde{n}_{a,b,R}||$.

20.2.3 Experimental results

The main theorem suggest that we can simply fit algorithms to minimize $\Sigma_j(< v_i, v_j > -PMI(i,j))^2$. Although this is extremely simple, it is still competitive against other algorithms such as CBOW, SkipGram or GloVe.

One other reasonable verification one can perform is to check the partition function Z_c are relatively the same for the different random vectors. In our experiment, we compute the partition function for 1000 random unit vectors and verify that the $Z_c = \Sigma \exp < v_w c >$ are relatively close.

A different application the word embedding can perform is the Polysemy. By performing sparse coding on the word vector in a dictionary of 2000 'atom' of discourse, one can separate the words by their 'meaning group'. For example, 'install', 'booting', 'linux' etc. would be grouped into one group because they are related to computer software, and 'sphere', 'geometry', 'dimension' etc. will be group into another because they are related to mathematics.

References

[HAL; Lund & Burgess'96] Lund, K.,Burgess, C. "Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28, 203–208 (1996).

[Levy & Goldberg'14] Levy, O.,Goldberg, Y. "Neural word embedding as implicit matrix factorization. *Advances in neural information processing systems*, 2177-2185 (2014).

[CW87] D. Coppersmith and S. Winograd, "Matrix multiplication via arithmetic progressions," *Proceedings of the 19th ACM Symposium on Theory of Computing*, 1987, pp. 1–6.