Neural Word Embedding as Implicit Matrix Factorization

Team 2: Clem, Violet, Marie, Sayed

Content

1) What is matrix factorization? What is it good for (especially in the context of Machine Learning)?
2) What did the researchers discovered?
3) Are they suggesting any improvements to the model due to their discovery?
4) What about personal opinion about it?

Introduction

- Embedding: class of technique where indiv. words are represented as vectors in predefined vector space. Word mapped to one vector / Vector values learned in a way resembling neural network.
- Application? Vector words embeddings that can predict words appearance around these words
- Most popular word embedding model: Word2Vec (repr. as neural network) or SGNS
- BUT authors proved that the SGNS vectors are implicitly better trained factorizing a matrix containing corpus information and stats

A simple explanation

Matrix factorization can be summarised as: $M = UV^T, U \in \mathbb{R}^{n \times d}, V \in \mathbb{R}^{m \times d}$

D = text corpus
M = number of documents in corpus
N = number of unique words in document

N x M word = Matrix[u,v]
U = word
V = document

For word embedding u / document embedding v, their product detain the info = how many times u occurs in v?
D = embedding dimensionality (large = better approximation)

→ Decompose the word-document matrix into a product of matrices. Matrix factorization is a way to generate latent features when multiplying two different kinds of entities.

What's matrix factorisation?

Matrix of a global corpus is defined as the shifted point-wise mutual information matrix M:

$$\mathbf{M}_{ij} := \mathbf{w}_j \cdot \mathbf{c}_i = PMI(i,j) - \log(k)$$

Word2vec would implicitly do matrix factorization of this big matrix filled up by these PMI statistics computed from the corpus (telling us how likely it is to see another particular word if you see one word appearing)

To implement SGNS with matrix factorization, following steps:

- Precompute the correct shifted-PMI matrix from the corpus; and,
- Determine the correct corresponding loss function for the factorization.

What is matrix factorization? EXAMPLE

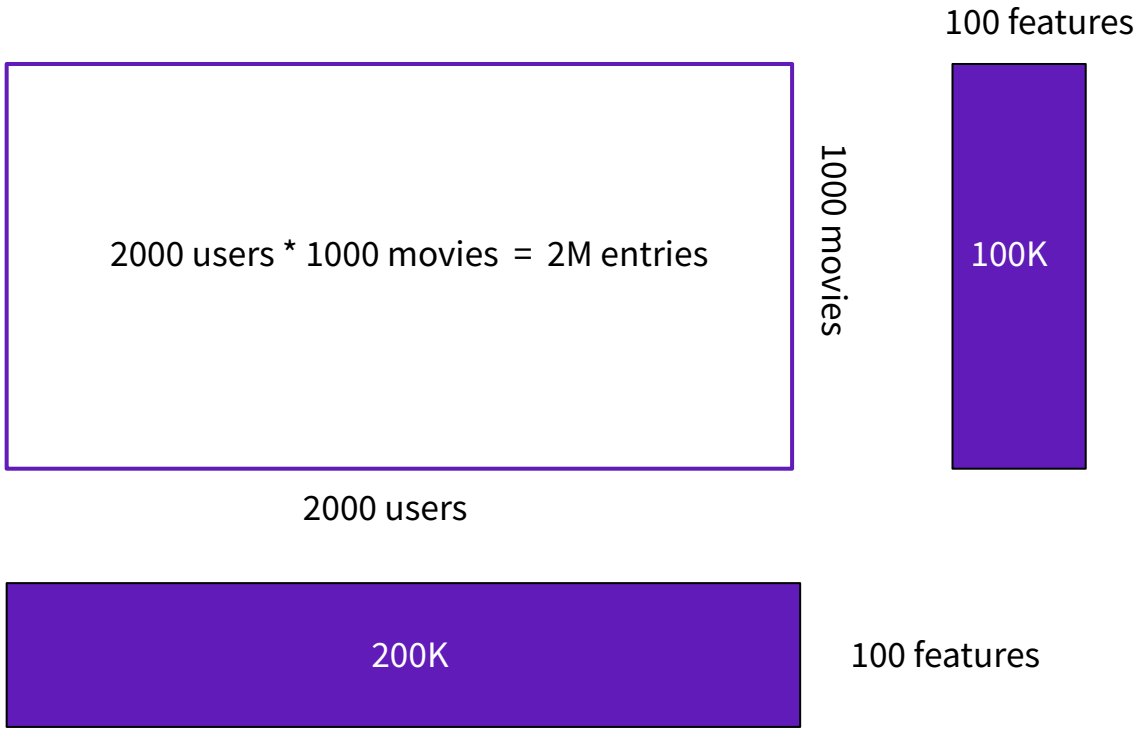
- Math
- Cinema example

$$N \begin{bmatrix} & M & \\ 1 & 2 & 3 \\ 5 & 10 & 15 \end{bmatrix} = \begin{bmatrix} 1 \\ 5 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 10 \\ 50 \end{bmatrix} \begin{bmatrix} 0.1 & 0.2 & 0.3 \end{bmatrix}$$

U V

O(N*M) O(N+M) $X = UV^T$

What is matrix factorization? EXAMPLE



What is matrix factorization? EXAMPLE

AxB

$$\begin{pmatrix} x & u \\ v & y \end{pmatrix}\begin{pmatrix} y & -u \\ -v & x \end{pmatrix}$$

$$= \begin{pmatrix} xy-uv & 0 \\ 0 & xy-uv \end{pmatrix}$$

$$= xy-uv \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} x & u \\ v & y \end{pmatrix} = A \qquad \begin{pmatrix} y & -u \\ -v & x \end{pmatrix} = B$$

$$\begin{pmatrix} 0 & A \\ B & 0 \end{pmatrix}\begin{pmatrix} 0 & A \\ B & 0 \end{pmatrix} = \begin{pmatrix} 0 & A \\ B & 0 \end{pmatrix}^2$$

$CxC = C^2$

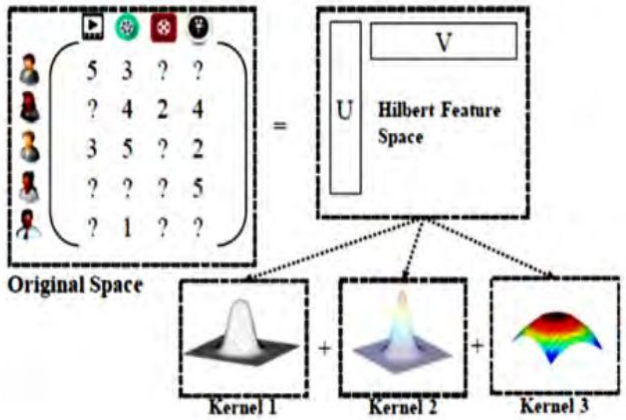
Benefits

computational benefits! (Storage of the matrix)

Links two important bodies of research: neural networks and distributional semantics

2 What did the researchers discovered?

The traditional MF algorithms method assumes that the data of matrix R are distributed on a linear hyperplane, but this assumption is not always true in reality. If the data of matrix R is distributed on a nonlinear hyperplane, the kernel-based matrix factorization method works, as shown in Figure. Firstly, two potential factor matrices U and V are embedded into a high- dimensional Hilbert space by a linear combination of kernel functions. Then, the nonlinear reconstruction of the score matrix in the original space is realized through the product of two latent factor matrices.



unknown score prediction based on matrix factorization of kernel function

- low-rank primitive eigenvalue matrix by singular value decomposition
- not possible for all data set in real life to obtain complete matrix effectively
- embed data into high-dimensional feature space to solve this problem.

KMF & MKLMF algorithms

Are proposed for collaborative filtering. Both algorithms can simultaneously utilize the nonlinear underlying association between rows (users) and columns (items) of the scoring matrix.

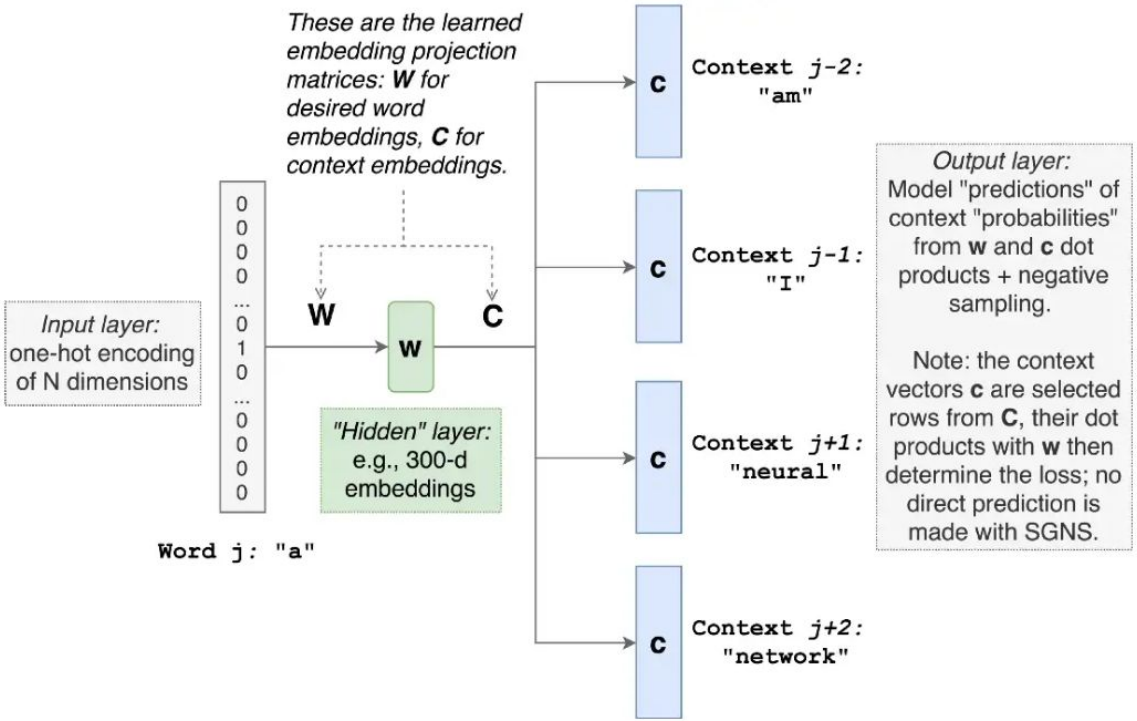
The KMF introduces the accounting method into the matrix factorization problem, embeds the low rank characteristic matrix into the higher dimensional space, and uses the score data of the original space to learn the nonlinear association.

The MKLMF algorithm further expands KMF, uses the observation data in the score matrix to learn the weight of each kernel function, and then synthesizes multiple kernel functions for collaborative filtering.

The simulation results show that the prediction accuracy of this algorithm is better than other current algorithms.

3 Are the researchers suggesting any improvements to the model due to their discovery?

SGNS(Skip-Gram with Negative Sampling)



An example of the typical SGNS diagram for sentence “am I a neural network”, with my annotations.

From SGNS to Matrix Factorization

$$\mathcal{L} = \sum_{(t,c)\in\Omega} \left[\log \sigma(\mathbf{w}_t \cdot \mathbf{c}_c) + \sum_{n\sim U}^{k} \log[1 - \sigma(\mathbf{w}_t \cdot \mathbf{c}_n)] \right]$$

Loss function for SGNS over the entire corpus. σ is the logistic sigmoid function.

$$\frac{\partial \mathcal{L}_{ij}}{\partial \mathbf{w}_i \cdot \mathbf{c}_j} = \left(N_{ij} + \frac{k N_i N_j}{N} \right) \left[\sigma(\mathbf{w}_i \cdot \mathbf{c}_j) - \sigma\left(\log \frac{N N_{ij}}{k N_i N_j} \right) \right]$$

Partial derivative of MF-SGNS loss function, with respect to the dot product.

$$\log \frac{N N_{ij}}{k N_i N_j} = PMI(i,j) - \log k$$

```
--- Word pairs with the notable PMIs ---

|          |              (PMI = 9.3956)
puerto     rico         (PMI = 8.9212)
rico       puerto       (PMI = 8.9182)
las        vegas        (PMI = 8.7001)
vegas      las          (PMI = 8.6940)
bin        laden        (PMI = 8.3084)
optional   trim         (PMI = 8.3080)
trim       optional     (PMI = 8.3073)
laden      bin          (PMI = 8.2967)
sri        lanka        (PMI = 8.1917)
lanka      sri          (PMI = 8.1655)
jacques    chirac       (PMI = 8.1147)
chirac     jacques      (PMI = 8.0814)
makeup     racial       (PMI = 8.0538)
racial     makeup       (PMI = 8.0533)
composite  nasdaq        (PMI = 8.0391)
nasdaq     composite    (PMI = 8.0328)
```

Improvements: MF-SGNS

PMI Matrix Factorization instead of Loss optimization of SGNS

Input:
Large corpus of text —-> Smaller file containing pre-extracted corpus statistics
such statistic extraction only ever needs to be run once

Training:
a fast, parallelized implementation can do so in less than 3 hours for a 10GB corpus of compressed .gz text files.

Sources

https://cla2019.github.io/embedmatrix.pdf
https://machinelearningmastery.com/what-are-word-embeddings/
https://medium.com/radix-ai-blog/unifying-word-embeddings-and-matrix-factorization-part-1-cb3984e95141
https://proceedings.neurips.cc/paper/2014/file/feab05aa91085b7a8012516bc3533958-Paper.pdf
https://towardsdatascience.com/recommendation-system-matrix-factorization-d61978660b4b
https://towardsdatascience.com/what-does-word2vec-actually-learn-489f3f950388
http://ruder.io/word-embeddings-1/index.html
https://www.youtube.com/watch?v=8LRT2gjHNVc
https://www.youtube.com/watch?v=4g8IVU9w3bg&list=WL&index=2
https://www.enjoyalgorithms.com/blog/text-data-pre-processing-techniques-in-ml
https://www.enjoyalgorithms.com/blog/word-vector-encoding-in-nlp
https://ronxin.github.io/wevi/
https://medium.com/radix-ai-blog/unifying-word-embeddings-and-matrix-factorization-part-1-cb3984e95141
http://jalammar.github.io/illustrated-word2vec/

Thank you!