Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network

Vikram Bhatt
Kapil Pathak

Department of Computational and Data Sciences
Department of Computer Science and Automation

November 15, 2018

Outline

Outline

Motivation

- Previous works[TWR+16] [YYH+14] [BGWB14] [BUGD+13] on link prediction focused on shallow,fast models which can scale to large knowledge graphs.
- These models learn less expressive features than deep multilayers models which potentially limits it's performance.

Motivation

- Previous works[TWR+16] [YYH+14] [BGWB14] [BUGD+13] on link prediction focused on shallow,fast models which can scale to large knowledge graphs.
- These models learn less expressive features than deep multilayers models which potentially limits it's performance.
- A novel embedding model ConvKB[NNNP17] employs Convolutional Neural networks for knowledge base completion task.
- Captures global relationships and transitional characteristics between entities and relations.

ConvKB Model

- Embedding of each triple (h, r, t) defined by (v_h, v_r, v_t) stacked in matrix $A = [v_h, v_r, v_t] \in \mathbb{R}^k$
- We use a filter $\omega \in \mathbb{R}^{1\times 3}$ operated on convolutional layer.
- Filter aims to capture global relationships + generalize transistional charectersitics as in other models.

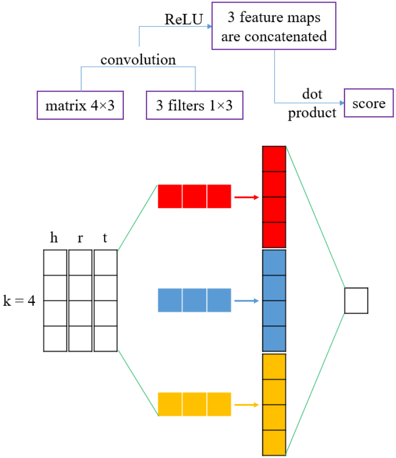


Figure 1: Process involved in ConvKB with embedding size $k = 4$ and number of filters $\tau = 3$

ConvKB Model

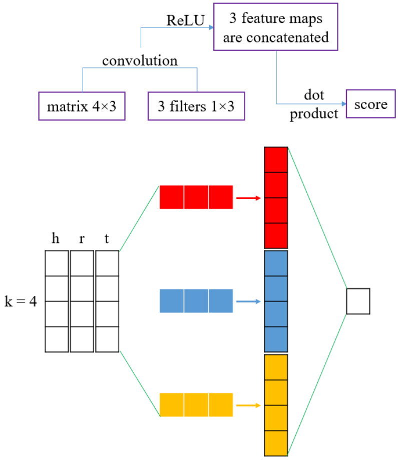


Figure 2: Process involved in ConvKB with embedding size $k = 4$ and number of filters $\tau = 3$

- Apply ω repeatedly over all rows and concatenate feature maps $\boldsymbol{v} = [v_1, v_2, \ldots, v_k] \in \mathbb{R}^k$
- $v_i = g(\omega.A_{i,:} + b)$ where b is bias and g is non-linear activation function.
- We use τ different filters and these τ feature maps concated into single vector $\chi \in \mathbb{R}^{\tau k \times 1}$

ConvKB Model

- Score function $f(h,r,t) = concat(g([v_h, v_r, v_t] * \omega)).w$
- Ω(set of all filters) and w are shared parameters and $*$ denotes convolution operation.

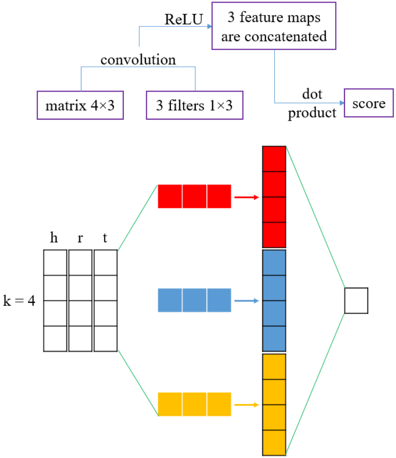


Figure 3: Process involved in ConvKB with embedding size $k = 4$ and number of filters $\tau = 3$

Loss function

$$L = \sum_{(h,r,t) \in \mathbb{G} \bigcup \mathbb{G}'} \log_e(1 + exp(l_{(h,r,t)}.f(h,r,t)))) + \frac{\lambda}{2}||w||^2$$

$$l_{(h,r,t)} = \begin{cases} 1 \text{ for } (h,r,t) \in \mathbb{G} \\ -1 \text{ for } (h,r,t) \in \mathbb{G}' \end{cases}$$

$\mathbb{G}$ is collection of valid triples in knowledge base.

$\mathbb{G}'$ is collection of invalid triples generated by corrupting triples in $\mathbb{G}$

Loss function

$$L = \sum_{(h,r,t) \in \mathbb{G} \bigcup \mathbb{G}'} \log_e(1 + exp(l_{(h,r,t)}.f(h,r,t)))) + \frac{\lambda}{2}||w||^2$$

$$l_{(h,r,t)} = \begin{cases} 1 \text{ for } (h,r,t) \in \mathbb{G} \\ -1 \text{ for } (h,r,t) \in \mathbb{G}' \end{cases}$$

$\mathbb{G}$ is collection of valid triples in knowledge base.

$\mathbb{G}'$ is collection of invalid triples generated by corrupting triples in $\mathbb{G}$

Given two entities (h,t) log-odd probability that (h,r,t) is true $\sigma(f(h,r,t))$, where σ is sigmoid unit link function[TWR+16].

Outline

Datasets

DATASET	WN	FB15K-237
ENTITIES	40,943	14,951
RELATIONS	18	1,345
TRAIN EX.	141,442	483,142
TEST EX.	5,000	50,000
VALID EX.	5,000	59,071

Table 1: [YYH+14]Some statsitics of datasets we initally used for training our model.

- Freebase(N-Triples RDF,22Gzip,1.9 billion triples)
- Contains too many redudancies, inverse realtions and noisy also.
- Requires lot of pre-processing.
- Eventually we want to scale up our model to the dataset.

Outline

Training Protocol

- Initial embeddings obtained from STransE[NNNP17] which are fed to CNN layer, available at https://github.com/datquocnguyen/STransE.
- Corrupted triples are sampled using Bernoulli trick ([WZFC14][dai13])
- We train our model parameters including entity and relation embeddings, filters and weight vector W

Asynchronous SGD

Figure 4: Learning curve for asynchronous SGD with 5 workers and 1 parameter server

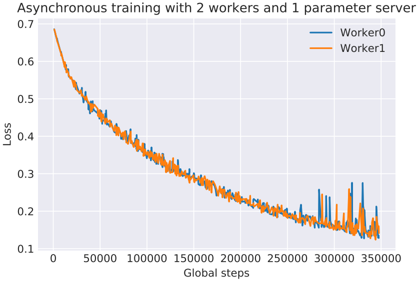


Figure 5: Learning curve for asynchronous SGD with 2 workers and 1 parameter server

Racing Conditions in Asynchronous Training

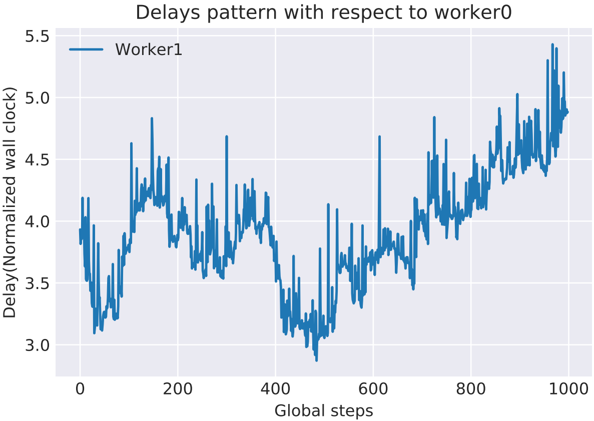


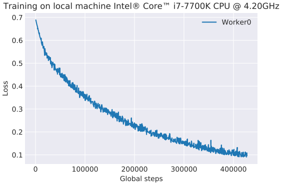
Figure 6: Intially both workers start at same point but eventually worker0 is straggling behind causing stale updates pushed at later steps during local gradient aggregation.Here I hand picked two workers to emphasize stark difference and increasing trend of lag after a certain point.

Bulk Synchronous Parallel Training

Figure 7: Learning curve for synchronous SGD with 5 workers and 1 parameter server

Figure 8: Learning curve for synchronous SGD with 2 workers and 1 parameter server

Just for comparison.....



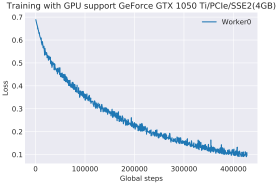


Figure 9: Learning curves in local machine(above Intel® Core™ i7-7700K CPU @ 4.20GHz) and below GPU GeForce GTX 1050 Ti/PCIe/SSE2

Train Setting	WallClock Time
GPU	37m
CPU	1hr 30min
1PS & 2W(Async)	4hr 23min
1PS & 5W(Async)	1hr 45min
1PS & 2W(Sync)	7hr 12min
1PS & 5W(Sync)	3hr

Table 2: Wallclock timings of our experiments took so far.

Outline

Plans Ahead

- We need to implement and scale up the inference part(Not done yet).
- Need to understand and pre-process the entire Freebase dataset(quite messy!).
- A possible direction to explore is stale synchronous parallel training for more stable and faster convergence.
- Identify bottlenecks in training and inference phase.
- A trade off between a possible improved link prediction scores by implementing deeper model and scalability issues due to increasing computational complexity

References I

- Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio, *A semantic matching energy function for learning with multi-relational data*, Mach. Learn. **94** (2014), no. 2, 233–259.

- Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko, *Translating embeddings for modeling multi-relational data*, Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2 (USA), NIPS'13, Curran Associates Inc., 2013, pp. 2787–2795.

- daiquocnguyen, *Project title*, `https://github.com/daiquocnguyen/ConvKB`, 2013.

References II

- Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung, *A novel embedding model for knowledge base completion based on convolutional neural network*, arXiv preprint arXiv:1712.02121 (2017).
- Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard, *Complex embeddings for simple link prediction*, International Conference on Machine Learning, 2016, pp. 2071–2080.
- Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen, *Knowledge graph embedding by translating on hyperplanes*, Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, AAAI'14, AAAI Press, 2014, pp. 1112–1119.

References III

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng, *Embedding entities and relations for learning and inference in knowledge bases*, arXiv preprint arXiv:1412.6575 (2014).