Auto-Encoding Translated Embeddings with Variance on Knowledge Graph

Bingjun Guo

(bingjun3)

Undergraduate

Auto-Encoding Translated Embeddings with Variance on Knowledge Graph

Introduction

Problem Set

This project will focus on prediction tasks regarding **Knowledge Graph**, which is a formally structured type of data. Normally a knowledge graph consists of a number of "tripes" in following form:

$$(head, relation, tail)$$

in which *relation* represents how *head* is related to *tail*. For instance:

$$(grandma, is\_mother\_of, papa)$$

Both *head* and *tail* are considered as "entities". In this project we concern about representation learning on knowledge graph, which aims at learning meaningful **vector representations** for both relations and entites, and thus address further challenges on knowledge graphs such as verifying unseen triples and inferring new triples. For example:

$$\begin{aligned} &(grandma, is\_mother\_of, papa) \land (grandpa, is\_husband\_of, grandma) \\ &\qquad\qquad\qquad\qquad \Rightarrow (grandpa, is\_father\_of, papa) \end{aligned}$$

In this project we will build a set of neural networks to learn those representations for prediction task on both entities and relations (i.e. given any two, predict the left one).

Related Work

TransE (Translated Embeddings) This approach was inspired by the **parallelogram analogy** discovered in representations learned in former methods. As a typical example, for entities *king*, *queen*, *woman*, *man*, the followinig relation of their learned representations was found:

$$\mathbf{king} - \mathbf{man} = \mathbf{queen} - \mathbf{woman}$$

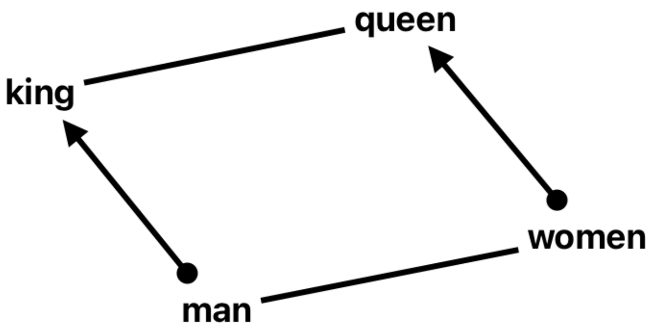

Figure 1. *Parallelogram Analogy*

Hence, TransE leverages this property directly and assigns the following objective for representations of all triples:

$$\mathbf{h} + \mathbf{r} = \mathbf{t}$$

in which **h**, **t** refer to representations of *head*, *tail* and **r** refers to representation of *relation*. Under the assumption of parallelogram analogy, representations of entities are affinely "translated" by those of relations. All representations are learned/optimized regarding this objective with gradient descent or other optimization methods. No neural network is involved

One of the limit of this model can be its disadvantage on representing 1-to-n or n-to-n relations. In this project we will try to resolve this through introducing neural networks.

VAE (Variational Auto-Encoder) This deep encode-decode method is improved upon auto-encoder. Basically, VAE introduces ramdom disturbance on encoded vectors, which is observed to force the smoothness of the latent representations. This property should help address entangled representations as were observed in Project 4.

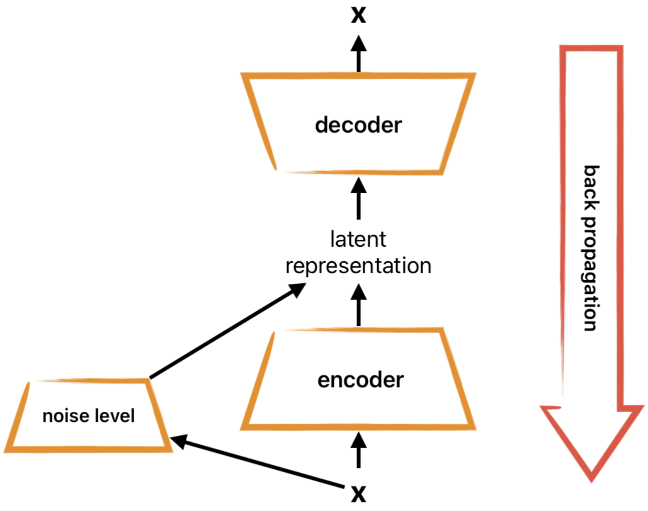

Figure 2. *VAE*

Just like auto-encoder, VAE is also self-supervised, aiming to decode original inputs from encoded latent representations, depicted by the objective:

$$d(e(\mathbf{x}) + \boldsymbol{\epsilon}_{\mathbf{x}}) = \mathbf{x}$$

in which $d(\cdot)$ stands for the decoder, $e(\cdot)$ stands for the encoder, $\mathbf{x}$ stands for the input, and $\boldsymbol{\epsilon}_{\mathbf{x}}$ stands for the random disturbance to the latent representation whose scale relies on $\mathbf{x}$. Decoder, encoder, and the noise generator are all learnable deep neural networks.

Method

In this project we propose a novel model, **"Variational Auto-Transcoder (VAT)"**, fusing the strength of TransE and VAE, that is, meaningful parallelogram analogy and less entangled deep representation, in order to alleviate issues of both models. In specific, instead of directly assigning vector representations to entites or relations, we decide to map entities and relations into and from the latent vector space with variational auto-encoder. Ideally, in this way, our model will be capable of learning both smooth and meaningful representations for entities and relations, enabling effective "translations" in the latent representation space. Basically, the smoothness guaranteed by variational

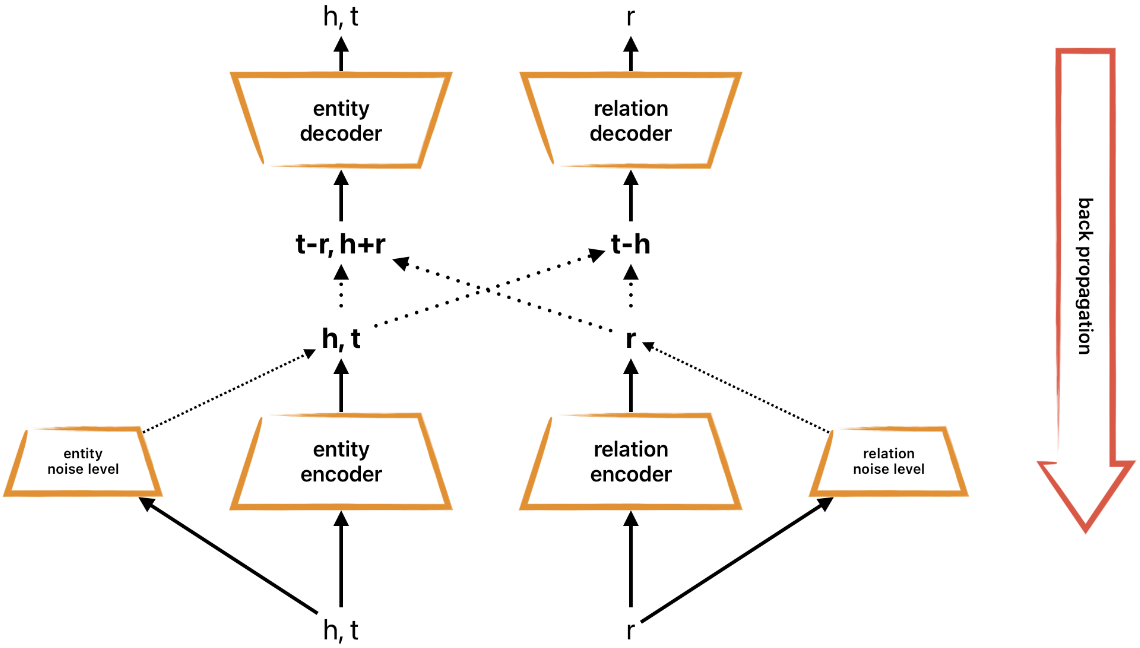

Figure 3. *Variational Auto-Transcoder (VAT)*

mapping is expected to reduce the entanglement of learned representations, which results in VAT's better understanding of translation. Meanwhile, the introduce of deep decoder is expected with better capability of understanding translated representations for inference.

Furthermore, as TransE being in form as a look-up table of vector representations for entities and relations, the process to make an inference on entities/relations with TransE would typically be:

1. Get the translated (predicted) representation. For example: $(\mathbf{h} + \mathbf{r})$ (predicting $\mathbf{t}$)

2. Check all the entities/relations and propose the closest one

The process can be rather time-consuming as number of entities/relations grows enormous. Regarding this issue, a key advantage of VAT is to directly map latent representations towards probability vectors which specifies confidence for each entity/relation with decoder. Once the vectors are prepared, predicted entities/relations will simply be told by the largest elements in the vectors, which would be much more efficient.

Experiments and Results

As experiments on our model, we trained and tested on three datasets, namely FB15K, FB15K237, and WN18RR, drawn from large scale knowledge bases Freebase and WordNet. The original scales of these datasets are too enormous for our networks which were implemented from scratch thus without hardware accelerations. Hence, we filtered for a limited number of entities in each dataset accordingly and kept only relations and samples that were associated. Sizes of the filtered datasets are as follow:

database	version (filtered)	entities	relations	samples
Freebase	FB15K	200	226	1408
Freebase	FB15K237	250	122	1392
Wordnet	WN18RR	500	10	400

All neural networks in VAT were in depth 4. The learning rates for both models were set as 0.006. Both models were trained for 50 epochs and then tested on predicting entities or relations in unseen triples, for example, $(\mathbf{h}^*, \mathbf{r}^*, \mathbf{?})$, on each dataset. Comparisons of their performances are as below:

database	Freebase				WordNet	
version	FB15K		FB15K237		WN18RR	
metrics	*mean rank (%)*	*top 10% (%)*	*mean rank (%)*	*top 10% (%)*	*mean rank (%)*	*top 10% (%)*
TransE	(**34**, 24, **22**)	(**28, 49, 51**)	(**43, 16**, 22)	(**16, 63, 57**)	(65, 30, 39)	(0, 36, 9)
VAT	(35, **20**, 23)	(21, 47, 47)	(45, 18, **19**)	(10, 55, **57**)	(**56, 21, 36**)	(**18, 64, 45**)

Three percentages in triples represent prediction metrics on *head*, *relation*, and *tail* respectively. *Mean rank* gives the mean probability ranking of target entity/relation in percentage, while *top 10%* states the partition of predictions in which such rankings are within top 10%.

Discussion

Overall, the two models perform closely on both databases. TransE performs slightly better than VAT on Freebase, while VAT shows more advantages on WordNet.

As for analysis of such result, tracing back to features of datasets, filtered Freebase has more relations, more number of training samples, and less entities. Since TransE didn't introduce artificial neural networks, its training and convergence was faster, and such difference would be enlarged in datasets which are with more samples while being complicated in relations. With more developed deep learning settings, VAT should have the potential to reach the same or even better performance as TransE on more complicated data, which is supported by VAT's better capability on WordNet. WordNet is much sparser in relations, which makes them more challenging to predict, since a single relation will represent translations between a larger number of different *head-tail* pairs.

Meanwhile, as was stated before, inference process of VAT can be more efficient since predictions are directly made by mapping instead of retrieving. Furthermore, the latent representations learned by VAT which are capable for mapping brings broader probability to the model, such as to transfer the representations for more downstream tasks involving neural networks.

In conclusion, for this project, we propose Variational Auto-Transcoder (VAT) which shows close or better performance as the classical baseline, TransE, while being more efficient and flexible for inference, on modeling knowledge graph under our computational limitations. As improvements, introduction of more deep learning techniques and frameworks is expected to result in a potentially better performance through helping resolve the convergence difficulty of artificial neural networks.