Convolutional 2D Knowledge Graph Embeddings

Pasquale Minervini, Pontus Stenetorp, Sebastian Riedel
University College London - AAAI18

Tingyu Wei

Outline

1. Background
2. Model
3. Experiments

Background

- A knowledge graph $\mathcal{G} = \{(s, r, o)\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$
- The link prediction: learning a scoring function.
- Scoring principle:

  The score of a triple x = (s, r, o) is proportional to the likelihood that the fact encoded by x is true.

Neural Link Predictors

- Neural link prediction models can be seen as multi-layer neural networks consisting of an encoding component and a scoring component. Given an input triple (s, r, o):
  - **Encoding component:** maps entities s, o their distributed embedding representations e_s, e_o .
  - **Scoring component**: the two entity embeddings e_s and e_o are scored by a function φ_r .

Neural Link Predictors

Table 1: Scoring functions $\psi_r(\mathbf{e}_s, \mathbf{e}_o)$ from neural link predictors in the literature, their relation-dependent parameters and space complexity; n_e and n_r respectively denote the number of entities and relation types, i.e. $n_e = |\mathcal{E}|$ and $n_r = |\mathcal{R}|$.

Model	**Scoring Function** $\psi_r(\mathbf{e}_s, \mathbf{e}_o)$	**Relation Parameters**	**Space Complexity**
SE (Bordes et al. 2014)	$\left\|\mathbf{W}_r^L \mathbf{e}_s - \mathbf{W}_r^R \mathbf{e}_o\right\|_p$	$\mathbf{W}_r^L, \mathbf{W}_r^R \in \mathbb{R}^{k \times k}$	$\mathcal{O}(n_e k + n_r k^2)$
TransE (Bordes et al. 2013a)	$\left\|\mathbf{e}_s + \mathbf{r}_r - \mathbf{e}_o\right\|_p$	$\mathbf{r}_r \in \mathbb{R}^k$	$\mathcal{O}(n_e k + n_r k)$
DistMult (Yang et al. 2015)	$\langle \mathbf{e}_s, \mathbf{r}_r, \mathbf{e}_o \rangle$	$\mathbf{r}_r \in \mathbb{R}^k$	$\mathcal{O}(n_e k + n_r k)$
ComplEx (Trouillon et al. 2016)	$\langle \mathbf{e}_s, \mathbf{r}_r, \mathbf{e}_o \rangle$	$\mathbf{r}_r \in \mathbb{C}^k$	$\mathcal{O}(n_e k + n_r k)$
ConvE	$f(\mathrm{vec}(f([\overline{\mathbf{e}_s}; \overline{\mathbf{r}_r}] * \omega))\mathbf{W})\mathbf{e}_o$	$\mathbf{r}_r \in \mathbb{R}^{k'}$	$\mathcal{O}(n_e k + n_r k')$

1D vs 2D Convolutions

1. Concatenate two rows of 1D embeddings:

$$([a \quad a \quad a]; [b \quad b \quad b]) = [a \quad a \quad a \quad b \quad b \quad b]$$

2. Concatenate two rows of 2D embeddings with dimension, extend to an alternating pattern:

$$\left(\begin{bmatrix} a & a & a \\ a & a & a \end{bmatrix}; \begin{bmatrix} b & b & b \\ b & b & b \end{bmatrix}\right) = \begin{bmatrix} a & a & a \\ a & a & a \\ b & b & b \\ b & b & b \end{bmatrix} \quad \begin{bmatrix} a & a & a \\ b & b & b \\ a & a & a \\ b & b & b \end{bmatrix}$$

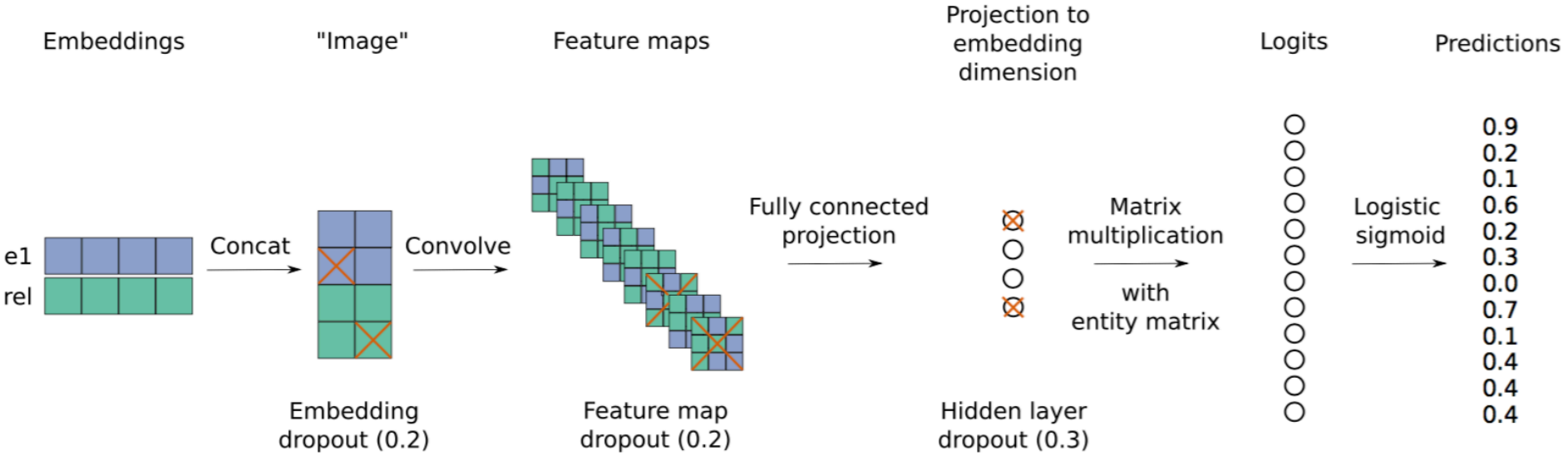
2D convolution vs 1D convolution : more feature interactions

Outline

1. Background
2. Model
3. Experiments

Model

Figure 1: In the ConvE model, the entity and relation embeddings are first reshaped and concatenated (steps 1, 2); the resulting matrix is then used as input to a convolutional layer (step 3); the resulting feature map tensor is vectorised and projected into a k-dimensional space (step 4) and matched with all candidate object embeddings (step 5).



Model

- Scoring Function

$$\psi_r(\mathbf{e}_s, \mathbf{e}_o) = \quad f(\mathrm{vec}(f([\overline{\mathbf{e}_s}; \overline{\mathbf{r}_r}] * \omega))\mathbf{W})\mathbf{e}_o$$

- Loss Function

$$\mathcal{L}(p, t) = -\frac{1}{N}\sum_i (t_i \cdot \log(p_i) + (1 - t_i) \cdot \log(1 - p_i))$$

1-N scoring

- **Why?**

Convolution consumes about 75-90% of the total computation time.

- **How?**
  - A triple (s, r, o), score it (1-1 scoring) .
  - One (s, r) pair, score it against all entities simultaneously (1-N scoring).

- **What?**

A ten-fold increase in the number of entities only increases the computation time by 25%.

Outline

1. Background
2. Model
3. Experiments

Knowledge Graph Datasets

- **WN18** : WordNet, tends to follow a strictly hierarchical structure.
- **FB15k** : Freebase, describes facts.
- **WN18RR and FB15k-237** : WN18 and FB15k where inverse relations are removed.
- **YAGO3-10** : YAGO3, deal with descriptive attributes of people.
- **Countries** : a benchmark dataset ,evaluate a model's ability to learn long-range dependencies.

Inverse Model

- **(feline, hyponym, cat)** to **(cat, hypernym, feline)**
- WN18 and FB15k have 94% and 81% test leakage as inverse relations.
- Rule-based model :
  - At test time, check if the test triple has inverse matches outside the test set.
  - If k matches are found, sample a permutation of the top k ranks for these matches.
  - If no match is found, we select a random rank for the test triple.

Results-1

Table 3: Link prediction results for WN18 and FB15k

	WN18					FB15k				
	MR	MRR	Hits @10	Hits @3	Hits @1	MR	MRR	Hits @10	Hits @3	Hits @1
DistMult (Yang et al. 2015)	902	.822	.936	.914	.728	97	.654	.824	.733	.546
ComplEx (Trouillon et al. 2016)	–	.941	.947	.936	.936	–	.692	.840	.759	.599
Gaifman (Niepert 2016)	**352**	–	.939	–	.761	75	–	.842	–	**.692**
ANALOGY (Liu, Wu, and Yang 2017)	–	**.942**	.947	.944	**.939**	–	.725	.854	.785	.646
R-GCN (Schlichtkrull et al. 2017)	–	.814	.964	.929	.697	–	.696	.842	.760	.601
ConvE	504	**.942**	.955	.947	.935	**64**	**.745**	**.873**	**.801**	.670
Inverse Model	567	.861	**.969**	**.968**	.764	1897	.706	.737	.718	.689

Results-2

Table 4: Link prediction results for WN18RR and FB15k-237

	WN18RR					FB15k-237				
			Hits					Hits		
	MR	MRR	@10	@3	@1	MR	MRR	@10	@3	@1
DistMult (Yang et al. 2015)	**5110**	.43	.49	.44	.39	254	.241	.419	.263	.155
ComplEx (Trouillon et al. 2016)	5261	.44	**.51**	**.46**	**.41**	339	.247	.428	.275	.158
R-GCN (Schlichtkrull et al. 2017)	–	–	–	–	–	–	.248	.417	.258	.153
ConvE	5277	**.46**	.48	.43	.39	**246**	**.316**	**.491**	**.350**	**.239**
Inverse Model	13219	.36	.36	.36	.36	7148	.009	.012	.010	.006

Results-3

Table 5: Link prediction results for YAGO3-10 and Countries

	YAGO3-10					**Countries**		
			Hits			AUC-PR		
	MR	MRR	@10	@3	@1	S1	S2	S3
DistMult (Yang et al. 2015)	5926	.34	.54	.38	.24	**1.00±0.00**	0.72±0.12	0.52±0.07
ComplEx (Trouillon et al. 2016)	6351	.36	.55	.40	.26	0.97±0.02	0.57±0.10	0.43±0.07
ConvE	**2792**	**.52**	**.66**	**.56**	**.45**	**1.00±0.00**	**0.99±0.01**	**0.86 ±0.05**
Inverse Model	60251	.02	.02	.02	.01	–	–	–

Parameter efficiency of ConvE

Table 2: Parameter scaling of DistMult vs ConvE.

Model	Param. count	Emb. size	MRR	Hits @10	Hits @3	Hits @1
DistMult	1.89M	128	.23	.41	.25	.15
DistMult	0.95M	64	.22	.39	.25	.14
DistMult	0.23M	16	.16	.31	.17	.09
ConvE	5.05M	200	.32	.49	.35	.23
ConvE	1.89M	96	.32	.49	.35	.23
ConvE	0.95M	54	.30	.46	.33	.22
ConvE	0.46M	28	.28	.43	.30	.20
ConvE	0.23M	14	.26	.40	.28	.19

Ablation Analysis

Table 7: Ablation study for FB15k-237.

Ablation	Hits@10
Full ConvE	0.491
Hidden dropout	-0.044 $\pm$ 0.003
Input dropout	-0.022 $\pm$ 0.000
1-N scoring	-0.019
Feature map dropout	-0.013 $\pm$ 0.001
Label smoothing	-0.008 $\pm$ 0.000

Conclusion

- Introducing a 2D convolutional link prediction model, ConvE.
  - Highly **parameter efficient**.
  - Fast through **1-N scoring**.
  - Expressive through **multiple layers of non-linear features**.
  - Robust to **overfitting** due to batch normalisation and dropout.
  - Achieves **state-of-the-art results** on several datasets.

- Investigate the severity of **test leakage**, introducing robust versions of datasets.

Thanks