Relational Neural Networks

Material based on the paper: Neural Networks for Relational Data

- Deep networks (NNs with several layers) have some limitations:
  - lack of interpretability and explainability
  - somewhow limited to tabular representations (propositional)
  - graph networks can be limited to binary relations and also by data dimension
- Symbolic and structured representations are interpretable and allow for multiple levels of abstraction
- *Relational neural networks*: embed existing data relations to the network **structure** and learn parameters

Relational Neural Networks

Example:

Given: Set of instances $\mathcal{F}$, $\mathbf{Target}$ relation, relational data set $(\mathbf{x}, y) \in \mathcal{D}$;
Construct (structure learning): $\mathtt{R_j}$, relational random walk rules (relational feature describing the network structure of $\mathcal{N}$);
Train (parameter learning): w_j, rule weights via gradient descent with rule-based parameter tying to identify a sparse set of network weights of $\mathcal{N}$

Example. *The movie domain contains the entity types (variables)* $\mathtt{Person(P)}$, $\mathtt{Movie(M)}$ *and* $\mathtt{Genre(G)}$. *In addition there are relations (features):* $\mathtt{Directed(P,M)}$, $\mathtt{ActedIn(P,G)}$ *and* $\mathtt{InGenre(M,G)}$. *The domain also has relations for entity resolution:* $\mathtt{SamePerson(P_1,P_2)}$ *and* $\mathtt{SameGenre(G_1,G_2)}$. *The task is to predict if* $\mathtt{P_1}$ *worked under* $\mathtt{P_2}$, *with the target predicate (label):* $\mathtt{WorkedUnder(P_1,P_2)}$.

Relational Neural Networks

Using random walks:

Example (continued). *The body of the rule*

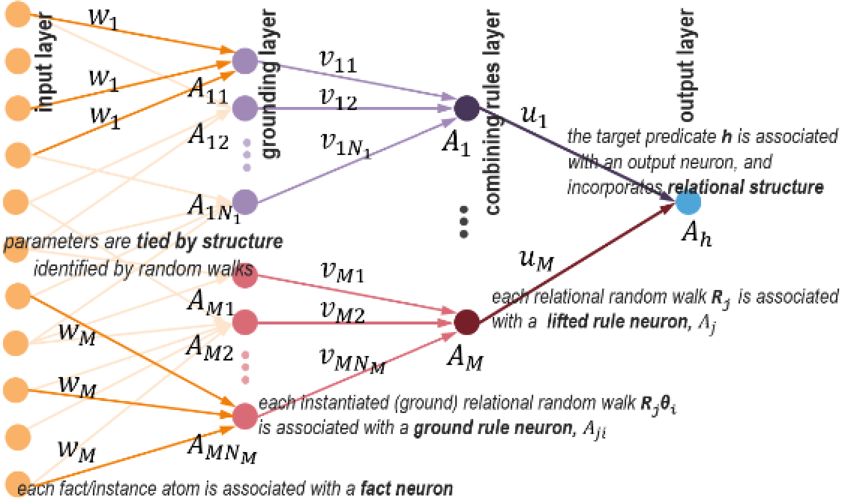
$$\mathtt{ActedIn}(\mathtt{P}_1, \mathtt{G}_1) \wedge \mathtt{SameGenre}(\mathtt{G}_1, \mathtt{G}_2) \wedge \mathtt{ActedIn}^{-1}(\mathtt{G}_2, \mathtt{P}_2) \wedge$$
$$\mathtt{SamePerson}(\mathtt{P}_2, \mathtt{P}_3) \wedge \mathtt{ActedIn}^{-1}(\mathtt{P}_3, \mathtt{M}) \wedge \mathtt{Directed}(\mathtt{M}, \mathtt{P}_4) \Rightarrow \mathtt{WorkedUnder}(\mathtt{P}_1, \mathtt{P}_4)$$

can be represented graphically as

$$\mathtt{P}_1 \xrightarrow{\mathtt{ActedIn}} \mathtt{G}_1 \xrightarrow{\mathtt{SameGenre}} \mathtt{G}_2 \xrightarrow{\mathtt{ActedIn}^{-1}} \mathtt{P}_2 \xrightarrow{\mathtt{SamePerson}} \mathtt{P}_3 \xrightarrow{\mathtt{ActedIn}^{-1}} \mathtt{M} \xrightarrow{\mathtt{Directed}} \mathtt{P}_4.$$

Relational Neural Networks

Network structure is built from output to input:



Relational Neural Networks

- Step 1: use path-constrained random walk to build $R_j, J = 1, 2, \ldots, M$ random walks $\rightarrow$ *relational features*
- paths are *lifted*: logical variable generalizes groundings

Relational Neural Networks

- Step 2: Output layer:
  - for the **Target**, which is also the head h (A_h in the net) in all R_j, introduce an output neuron called the target neuron $\rightarrow$ one-hot encoding can be used to code the target labels
  - The target neuron uses the softmax activation function (for multinomial probability distribution on the multiple classes)

Relational Neural Networks

- Step 3: *Combination Rules layer*: target is connected to M lifted rule neurons $A_j, j = 1, \dots, M \rightarrow$ lifted relational random walks (R_j, w_j)
  - each R_j is a conjunction of predicates:

$$Q_1^j(X, .) \wedge \cdots \wedge Q_L^j(., Z) \rightarrow Target(X, Z), j = 1, \dots, M$$

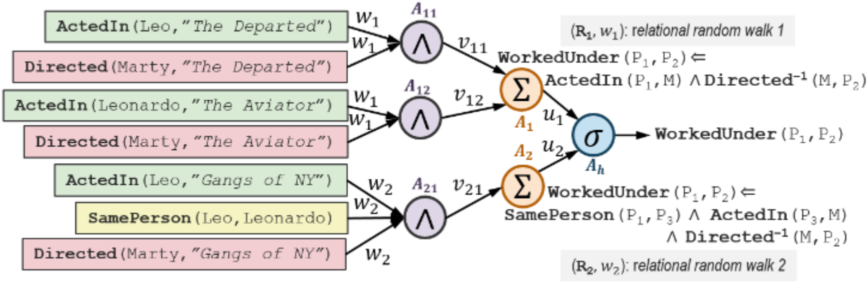
  - each lifted A_j combines the different groundings
- train the network to learn parameters

Relational Neural Networks

- Step 4: *Grounding layer*: for each instance of A_j introduce a *ground rule neuron* A_{ji}
- A_{ji} is only activated if all ground term of its conjunction is true
- these neurons are connected to the input layer which consists of the facts that compose each R_j

Relational Neural Networks

Example:



Relational Neural Networks

- special case of a *convolutional* neural network (CNN)
  - fact-grounding layer edges are equivalent of convolution (reduce dimensions)
  - combination rules layer: pooling
  - fully-connected layer (output): softmax

Relational Neural Networks

Limitations:

- same weights to distinct ground predicates
- binary predicates
- no connection between neurons in the same layer
- random walks may not produce representative paths
- why RNN if paths can be searched using ILP?
- why the RNN is built in layers?