Relational inductive biases, deep learning, and graph networks

Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, Razvan Pascanu

arXiv: 1806.01261

https://qdata.github.io/deep2Read

Presenter: Arshdeep Sekhon

Motivation

- Humans represent complex systems as compositions of entities and their interactions
- Combinatorial generalization: fit new knowledge into our existing structured representations, or adjust the structure itself to better accommodate

Motivation

- Humans represent complex systems as compositions of entities and their interactions
- Combinatorial generalization: fit new knowledge into our existing structured representations, or adjust the structure itself to better accommodate
- previous DNNs:
  - end-to-end design;
  - no hand engineering;
  - minimal a priori representational and computational assumptions

Motivation

- “reject the notion that structure and flexibility are somehow at odds or incompatible, and embrace both ”
- Graph NNs: intersection of deep learning and structured approaches
- strong relational inductive biases: specific architectural assumptions to learn about enitities and relations
- examine various deep learning methods through the lens of their relational inductive biases

Relational Reasoning

- structure : product of composing a set of known building blocks
- entity is an element with attributes
- relation is a property between entities. Relations between two objects might include same size as, heavier than, and distance from
- A rule is a function (like a non-binary logical predicate) that maps entities and relations to other entities and relations

Relational inductive Biases

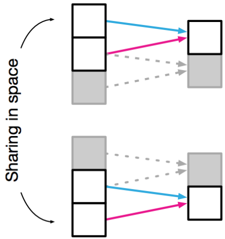
- An inductive bias allows a learning algorithm to prioritize one solution (or interpretation) over another, independent of the observed data
- prior distribution/regularization/architecture
- improve the search for solutions without substantially diminishing performance, as well as help find solutions which generalize in a desirable way

Relational inductive biases in standard deep learning building blocks

Fully connected layers

- entities are the units in the network,
- the relations are all-to-all (all units in layer i are connected to all units in layer j),
- the rules are specified by the weights and biases
- there is no reuse, and there is no isolation of information
- weak relational inductive bias

Convolutional layers



- INDUCTIVE BIAS: locality and translation invariance
- Locality : arguments to the relational rule are those entities in close proximity with one another in the input signals coordinate space
- Translation invariance reflects reuse of the same rule across localities in the input.

Relational inductive b

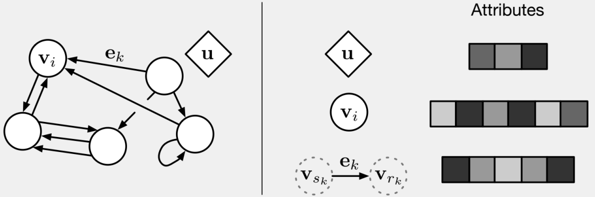
Recurrent Layers

- inputs and hidden states at each processing step as the entities
- Markov dependence of one steps hidden state on the previous hidden state and the current input, as the relations.
- RULE: combining the entities takes a steps inputs and hidden state as arguments to update the hidden state.
- INDUCTIVE BIAS: temporal invariance (rule reuse)
- INDUCTIVE BIAS: a bias for locality in the sequence via their Markovian structure

Computation over Sets and Graphs

- Desired inductive bias: permutation invariance
- Desired inductive bias: each object in a set may be affected by pairwise interactions with the other objects in the set

Graph Network Block



Here we use "graph" to mean a directed, attributed multi-graph with a global attribute. In our terminology, a node is denoted as $\mathbf{v}_i$, an edge as $\mathbf{e}_k$, and the global attributes as $\mathbf{u}$. We also use s_k and r_k to indicate the indices of the sender and receiver nodes (see below), respectively, for edge k. To be more precise, we define these terms as:

Directed : one-way edges, from a "sender" node to a "receiver" node.
Attribute : properties that can be encoded as a vector, set, or even another graph.
Attributed : edges and vertices have attributes associated with them.
Global attribute : a graph-level attribute.
Multi-graph : there can be more than one edge between vertices, including self-edges.

Key Design Principles

- Flexible representations
- Configurable within-block structure
- Composable multi-block architectures

GNN Algorithm

Algorithm 1 Steps of computation in a full GN block.

function GRAPHNETWORK(E, V, $\mathbf{u}$)
    for $k \in \{1 \ldots N^e\}$ **do**
        $\mathbf{e}'_k \leftarrow \phi^e\left(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}, \mathbf{u}\right)$ ▷ 1. Compute updated edge attributes
    end for
    for $i \in \{1 \ldots N^n\}$ **do**
        let $E'_i = \left\{\left(\mathbf{e}'_k, r_k, s_k\right)\right\}_{r_k = i,\; k=1:N^e}$
        $\bar{\mathbf{e}}'_i \leftarrow \rho^{e \rightarrow v}\left(E'_i\right)$ ▷ 2. Aggregate edge attributes per node
        $\mathbf{v}'_i \leftarrow \phi^v\left(\bar{\mathbf{e}}'_i, \mathbf{v}_i, \mathbf{u}\right)$ ▷ 3. Compute updated node attributes
    end for
    let $V' = \left\{\mathbf{v}'\right\}_{i=1:N^v}$
    let $E' = \left\{\left(\mathbf{e}'_k, r_k, s_k\right)\right\}_{k=1:N^e}$
    $\bar{\mathbf{e}}' \leftarrow \rho^{e \rightarrow u}\left(E'\right)$ ▷ 4. Aggregate edge attributes globally
    $\bar{\mathbf{v}}' \leftarrow \rho^{v \rightarrow u}\left(V'\right)$ ▷ 5. Aggregate node attributes globally
    $\mathbf{u}' \leftarrow \phi^u\left(\bar{\mathbf{e}}', \bar{\mathbf{v}}', \mathbf{u}\right)$ ▷ 6. Compute updated global attribute
    return $(E', V', \mathbf{u}')$
end function

Relational inductive biases in graph networks

- graphs can express arbitrary relationships among entities,
- graphs represent entities and their relations as sets, which are invariant to permutations.
- a GNs per-edge and per-node functions are reused across all edges and nodes, respectively.

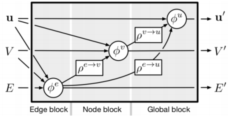
Relational inductive

Design Principle: Flexible Representations

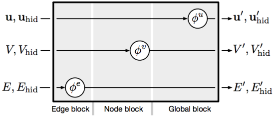
- Attributes
  - edge-focused GN : the edges as output, for example to make decisions about interactions among entities
  - node-focused GN uses the nodes as output
  - graph-focused GN uses the globals as output
- Graph Structure: known vs not known

Design Principle: Flexible Representation

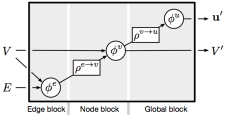
- Configurable within-block structure



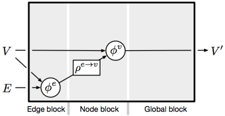
(a) Full GN block



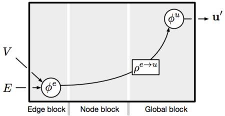
(b) Independent recurrent block



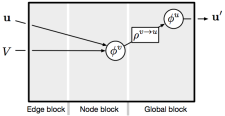
(c) Message-passing neural network



(d) Non-local neural network



(e) Relation network



(f) Deep set

Design Principles: Composable multi-block architectures

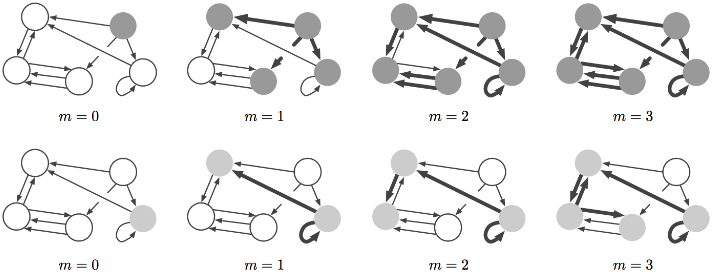


Figure 7: Example of message passing. Each row highlights the information that diffuses through the graph starting from a particular node. In the top row, the node of interest is in the upper right; in the bottom row, the node of interest is in the bottom right. Shaded nodes indicate how far information from the original node can travel in m steps of message passing; bolded edges indicate which edges that information has the potential to travel across. Note that during the full message passing procedure, this propagation of information happens simultaneously for all nodes and edges in the graph (not just the two shown here).

Limitations

- cannot be guaranteed to solve some classes of problems, such as discriminating between certain non-isomorphic graphs.
- notions like recursion, control flow, and conditional iteration are not straightforward to represent with graphs
- covariance rather than invariance to permutations of the nodes and edges is preferable

Open Questions

- inferring graph structure
- adaptively modify graph structures during the course of computation.
- interpretability of the behavior of graph networks.