Cognitive Module Networks for Grounded Reasoning

Alexey Potapov, Anatoly Belikov, Vitaly Bogdanov, Alexander Scherbatiy

AGI’19 @ Shenzhen
Motivating Example: VQA

- Simple questions about real-world scenes
- Complex questions about simple synthetic scenes
Motivating Example: VQA

- Simple questions about real-world scenes
  - Is the car yellow?

- Complex questions about simple synthetic scenes
Motivating Example: VQA

• Simple questions about real-world scenes
  • Difficult to assign all possible labels [child, boy, human, person, happy, jumping, tall, etc.] to all possible objects
  ⇒ Symbolic reasoning doesn’t work and is not needed
  ⇒ Subsymbolic DNN attention models are used*

*https://arxiv.org/abs/1803.07724
Motivating Example: VQA

- Simple questions about real-world scenes
- Complex questions about simple synthetic scenes*

Q: Are there an equal number of large things and metal spheres?
Q: What size is the cylinder that is left of the brown metal thing that is left of the big sphere?
Q: There is a sphere with the same size as the metal cube; is it made of the same material as the small red sphere?

*https://arxiv.org/abs/1612.06890
Motivating Example: VQA

- Simple questions about real-world scenes
- Complex questions about simple synthetic scenes
  - Feed-forward DNNs can reliably extract definite labels for objects
  - Questions are compositional that is badly modeled by DNNs
  \Rightarrow Symbolic reasoning disentangled from vision and language understanding works well (e.g. *)

*https://arxiv.org/abs/1810.02338
Motivating Example: VQA

- Simple questions about real-world scenes
- Complex questions about simple synthetic scenes
- Complex questions about complex scenes
Motivating Example: VQA

- Simple questions about real-world scenes
- Complex questions about simple synthetic scenes
- Complex questions about complex scenes
  - Are all boats on the same side of the river as bicycles and cars?
Motivating Example: VQA

• Simple questions about real-world scenes
• Complex questions about simple synthetic scenes
• Complex questions about complex scenes
  • How many trees with green leaves were there?
  • How many boats with humans in them were there?
  • ...

...
Motivating Example: VQA

• Simple questions about real-world scenes
• Complex questions about simple synthetic scenes
• Complex questions about complex scenes
  • How many trees with green leaves were there?
  • How many boats with humans in them were there?
  • ...
⇒ We need the images to answer these questions
Motivating Example: VQA

• Simple questions about real-world scenes
• Complex questions about simple synthetic scenes
• Complex questions about complex scenes
  ⇒ Reasoning over images themselves is needed!
• Can be studied on CLEVR also*

*https://arxiv.org/abs/1803.05268
Program Generation as Visual Reasoning

Question: Are there more cubes than yellow things?

Answer: Yes
Program Generation as Visual Reasoning

• Not really reasoning
• Direct, deterministic, specialized (yet learnt) mapping from questions to programs
• Imperative, not declarative
• Solving only one problem (cannot be reused for caption generation, etc.)
• No way to incorporate knowledge

⇒ Grounded reasoning is needed: full-fledged symbolic reasoning but over grounded symbols
Questions as Queries to Knowledge Base

• Is the pyramid green?
  ?– color(X, green), shape(X, pyramid).

• Are all green pyramids large?
  ?– not(shape(X, pyramid),
       color(X, green),
       not(size(X, large)))).

⇒ Powerful deductive reasoning with the use of knowledge (e.g. child(X):–boy(X), left(X,Y):–
     right(Y,X), etc.)
Queries in OpenCog

• What color is the large car?
• AndLink
  InheritanceLink
    VariableNode “X”
    ConceptNode “car”
  InheritanceLink
    VariableNode “Y”
    ConceptNode “color”
EvaluationLink
  PredicateNode “color_of”
  ListLink(VariableNode “X”, VariableNode “Y”)
EvaluationLink
  PredicateNode “size_of”
  ListLink(VariableNode “X”, ConceptNode “large”)
Grounded Reasoning in OpenCog

• But we don’t want just symbolic reasoning!
Grounded Reasoning in OpenCog

• But we don’t want just symbolic reasoning!

⇒ OpenCog readily provides GroundedPredicate and GroundedSchema nodes

EvaluationLink
    GroundedPredicateNode “py:filter_color”
    ListLink(ConceptNode “image”, ConceptNode “green”)

⇒ DNNs can be executed in the course of reasoning (Pattern Matcher, Unified Rule engine) as needed
Grounded Reasoning in OpenCog

⇒ OpenCog readily provides GroundedPredicate and GroundedSchema nodes
⇒ DNNs can be executed in the course of reasoning (Pattern Matcher, Unified Rule engine) as needed

Unfortunately,
• Tensors cannot be transmitted from one neural model to another through Atomspace as is
• Computation graphs of back-end DNN frameworks are broken and backpropagation doesn’t work
• DNNs created in run-time as objects cannot be addressed directly by GroundedPredicates or Schemas
CogNets

• A technical solution to the above problems
• PyTorch wrapper to OpenCog
• Main features:
  • Existing PyTorch models can be reused without modifications; inheritance of neural modules should only be changed from torch.nn.Module to CogModule
  • CogModule constructor accepts Atom to which a neural module should be bound
  • OpenCog’s BindLink can be used with small changes to syntax
CogNets

- `class InheritanceModule(CogModule)`
  - declaring PyTorch.nn module attachable to Atoms

- `h = InheritanceLink(ConceptNode("red"), ConceptNode("color"))`
  - creating PyTorch.nn module object and attaching it to the actual Atom in Atomspace initializing it with tensor truth value

⇒ A convenient way to specify DNN models and mix them up with symbolic knowledge in Atomese
CogNets

BindLink(
    VariableNode("$X"),
    AndLink(
        InheritanceLink(VariableNode("$X"),
                           ConceptNode("color")),
        AndNet.evaluate(
            execute(InheritanceLink(VariableNode("$X"),
                                    ConceptNode("color"))),
            execute(VariableNode("$X"), input.execute()))),
    VariableNode("$X"))

⇒ OpenCog’s BindLink (with Pattern Matcher or Probabilistic Logic Network) can readily be used
⇒ DNNs assembled in the course of variable grounding and reasoning remain trainable
Proof of Concept

• Transparency-by-design model is reproduced with the use of OpenCog instead of ad hoc imperative program executor

• MNIST example from DeepProbLog* is reproduced

\[2 + 5 = 8\]

⇒ OpenCog reasoning capabilities can be used to reason over symbols grounded in DNNs, or, equivalently, assemble arbitrary module networks on fly depending on the semantic content

⇒ Tightly integrative hybrid neural-symbolic system

* https://arxiv.org/abs/1805.10872
Thank you for attention!

Contact: alexey@singularitynet.io