Learning to Compose Neural Networks for Question Answering

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, Dan Klein

*NAACL 2016 Best Paper*

Presented by: Aditya Gupta
Task Description

Learning to dynamically compose modular neural networks for grounded question answering.
“Grounding” the task description

Composing Modular Neural networks for Grounded Question Answering

Is there a red shape above a circle?

What rivers are in South Carolina?

What color is the necktie?
Neural Module Networks

Is there a red shape above a circle?

[describe] [and]
[find] [relate] [exists]

Is there a red shape above a circle?

exists
and
red
above
circle

yes
Contributions

Using structure of language to dynamically produce structure of model.

Combined advantages of both:

- **Representation Learning**: Learning lexical grounding by end to end deep neural nets (continuous)

- **Compositionality**: Just like in semantic parsers (discrete)
Sentence meaning are computations

Language of sets: sets encode meanings + set transformations encode meanings

Relaxed real valued vector functions rather than discrete output modules
Backpropagating through the tree structure.
Deciding the modules

- Deciding the specs of task specific submodules we need for the task at hand
  - Inputs
  - Outputs

- Modules outputs:
  - Distribution over input (Attention)
  - Distribution over output (Label)

- Fixing the module inventory!
[find] module (→ Attention)

Is there a red shape above a circle?

What color is the triangle?

Who is running in the grass?

What cities are south of San Diego?
Is there a red shape above a circle?

What color is the triangle?

Who is running in the grass?

What cities are south of San Diego?
[exists] module (Attention → Labels)

- [exists] → true

Is there a red shape above a circle?

What color is the triangle?

Who is running in the grass?

[find] [relate]

What cities are south of San Diego?
[describe] module

Is there a red shape above a circle?

What color is the triangle?

Who is running in the grass?

What cities are south of San Diego?
[and] module \[\text{(Attention}^* \rightarrow \text{Attention)}\]

- Is there a red shape above a circle?
- What color is the triangle?
- Who is running in the grass?
- What cities are south of San Diego?
Module Inventory

*Additional module [lookup] which is for hard attention.
Learning with fixed layout

Maximize \[ \sum_{(w, y, z)} \log p_z(y|w; \theta_e) \] with a parameter tying scheme based on the observed \( z \).
Properties

No supervision for individual modules

Context drives the module behavior

Mapping to MR is combinatorial optimization problem in standard semantic parsing. Mapping words in vocabulary to relatively small inventory of semantic symbols (discrete)

Here the lexicon learning becomes a continuous optimization problem
Building Layouts

Is there a red shape above a circle?

Rule based
Putting it together!

Is there a red shape above a circle?

...
Layout Probabilities

Which layout to choose? Assign probabilities to each candidate layout conditioned on the query.

Generate potentials:

\[ s(z_i|x) = a^\top \sigma(Bh_q(x) + Cf(z_i) + d) \]

where \( f(z_i) \) is the feature vector for the \( i^{th} \) layout.

Taking softmax over all candidate layouts:

\[ p(z_i|x; \theta) = \frac{e^{s(z_i|x)}}{\sum_{j=1}^{n} e^{s(z_j|x)}} \]
Putting it all together

Layout selection module + Execution module

Input Question → Layout Probabilities → Execution Module

Optimize $\log p(y|z, w; \theta_e)$ using ordinary backprop

Optimize $p(z|x; \theta_\ell)$ using policy gradient

$\nabla J(\theta_\ell) = \mathbb{E}[\nabla \log p(z|x; \theta_\ell) \cdot r]$

$\mathbb{E}[(\nabla \log p(z|x; \theta_\ell)) \cdot \log p(y|z, w; \theta_e)]$
Experiments

### VQA

<table>
<thead>
<tr>
<th></th>
<th>test-dev</th>
<th></th>
<th>test-std</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Yes/No</td>
<td>Number</td>
<td>Other</td>
<td>All</td>
</tr>
<tr>
<td>Zhou (2015)</td>
<td>76.6</td>
<td>35.0</td>
<td>42.6</td>
<td>55.7</td>
</tr>
<tr>
<td>Noh (2015)</td>
<td>80.7</td>
<td>37.2</td>
<td>41.7</td>
<td>57.2</td>
</tr>
<tr>
<td>Yang (2015)</td>
<td>79.3</td>
<td>36.6</td>
<td>46.1</td>
<td>58.7</td>
</tr>
<tr>
<td>NMN</td>
<td>81.2</td>
<td>38.0</td>
<td>44.0</td>
<td>58.6</td>
</tr>
<tr>
<td>D-NMN</td>
<td>81.1</td>
<td>38.6</td>
<td>45.5</td>
<td>59.4</td>
</tr>
</tbody>
</table>

### GeoQA

<table>
<thead>
<tr>
<th>Model</th>
<th>GeoQA</th>
<th>GeoQA+Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSP-F</td>
<td>48</td>
<td>–</td>
</tr>
<tr>
<td>LSP-W</td>
<td>51</td>
<td>–</td>
</tr>
<tr>
<td>NMN</td>
<td>51.7</td>
<td>35.7</td>
</tr>
<tr>
<td>D-NMN</td>
<td><strong>54.3</strong></td>
<td><strong>42.9</strong></td>
</tr>
</tbody>
</table>

- Improved results on both tasks
- Compared to previous work on module networks (NMN) where there’s a fixed layout for each question based on dependency parse
Critique
Parse -> Layouts

- Shallow description of actual mechanism of using the dependency parse to generate layout candidates.
  - Rule based mechanism over Stanford dependency parse output

- Doesn’t provide examples what the candidate networks look like

- Performance using the most complex layout

- Inference technique

- Statistics about #layouts

```prolog
(shape seat) (is shadow) (is multiple) (is modern);(is train);(is (and modern train)) (color stripe);(color train);(color (and stripe train)) (what other);(what side);(what (and other side)) (is bus);(is driver);(is (and bus driver)) (is bus);(is move);(is (and bus move)) (color bus) (is item);(is sale);(is (and item sale)) (what sale);(what tent);(what (and sale tent)) (is grocery);(is store);(is (and grocery store))
```
Featurizing layouts

Creates feature vector with binary indicators:
- An indicator for the sub-module type input (color, shape, above, etc.)
- An indicator for each sub-module
VQA: Not so modular

- VQA only uses 3 modules: [find] + [describe] + [and] + atmost 2 conjuncts

- Doesn’t perform well by simply using the proposed D-NMN.
  - Changed the model to generate an answer embedding rather than distribution over labels.
  - Extra hidden layer to combine input sent. representation

- Doesn’t talk about the performance of VQA by just using the proposed D-NMN architecture without the modification.

- NMN has a different set of modules.
Variance reduction for policy gradient

The next paper\textsuperscript{1} in this line of work uses two standard variance reduction techniques for the policy gradient:

- **Baseline techniques**
  - Exponential moving average over recent losses

- **Optimizing by initializing using expert policy**
  - Minimizing KL loss between the expert policy and layout policy

Questions?