Deep Compositional Question Answering with Neural Module Networks

Andreas, J. & Rohrbach, M. & Darrell, T. & Klein, D.
University of California, Berkeley
Visual Question Answering

- How do we answer these questions?
- What breed is the dog?
  - 2 steps:
    - Look for dog
    - Infer breed
How many shapes are to the left of a red circle?

- 4 steps:
  - Look for circles
  - Identify red ones
  - Look to their left
  - Count shapes
Visual Question Answering

• It’s compositional by nature
  – `classify[breed](attend[dog])`
  – `count[shapes](re-attend[left](combine[and] (attend[red],attend[circle])))`

• Upper bound?
  – Potentially none

• There’s no “single best network”
  – Function mapping questions & contexts to answers
Neural Module Networks (NMN)
Neural Module Networks (NMN)

- General architecture for composing neural modules into deep networks
- Neural modules
  - Composable
  - Jointly-trained
  - Heterogeneous
    - Perform different computations depending on the “messages” exchanged among them
Each **training datum** is a 3-tuple \((w, x, y)\):
- \(w\) is a natural-language question
- \(x\) is an image
- \(y\) is an answer

A **Model** consists of 2 parts:
- Collection of modules \(\{m\}\)
- Predictor \(P\) maps *string* \(\rightarrow\) *network*
  - \(P(w) \rightarrow\) network layout
  - \(p(y \mid w, x; \theta)\) \(\theta\) – modules parameters
• **Task**: identify minimal set of modules needed
  - Attention, Re-attention, Combination, Classification, and Measurement
• Modules operate on 3 data types
  - Images
  - Attentions (unnormalized)
  - Labels (classification predictions)
• For visual question answering task:
  - Interesting compositional phenomena occur in attention space
  - Other types can be added later
Modules

- **TYPE[INSTANCE](ARG1, . . .)**
  - Type – high-level module type (Ex. attend, classify,...)
  - Instance – particular instance under consideration (Ex. dog, car, ...)
  - Modules with no arguments implicitly take images as input.

\[
\text{attend} : \text{Image} \rightarrow \text{Attention}
\]
Modules

\[
\text{re-attend} : \text{Attention} \rightarrow \text{Attention}
\]

\[
\text{combine} : \text{Attention} \times \text{Attention} \rightarrow \text{Attention}
\]
**Modules**

classify: \[ \text{Image} \times \text{Attention} \rightarrow \text{Label} \]

measure: \[ \text{Attention} \rightarrow \text{Label} \]
Strings to Networks

- **Predictor $P$**
  - Takes in question $w$ outputs a layout
  - Has 3 main stages: parsing, filtering, building

- **Parsing**
  - Stanford Parser
    - Dependency representation
    - Express grammatical relations
      - Objects $\rightarrow$ attributes
      - Events $\rightarrow$ participants
      - Lemmatization (helps with sparsity of instances)
Strings to Networks

• Filtering
  – Keep dependencies connected to *wh-word*
    • *What is standing in the field?*
      – what(stand)
    • *What color is the truck?*
      – color(truck)
    • *Is there a circle next to a square?*
      – is(circle, next-to(square))
  – *Remove determiners and modals*
    • *What type of cakes were they? What type of cake is it?*
      – type(cake)
Strings to Networks

- Final Layout construction
  - Leaves become “attend” modules
  - Internal nodes “re-attend” or “combine”
    • Depending on arity
  - Root nodes become “classify” or “measure”
    • Depending on question
      - Measure → yes/no
      - Classify → all others
Final Model

- Combination of NMN predictions and simple LSTM question encoder.
  - Deals with the simplification caused by the parser
  - Captures *semantic* regularities
    - *What color is the bear?*
      - *Green* – not likely
      - *Brown* – more likely
- Standard single-layer LSTM 1024 hidden units
- Final prediction is the geometric average of the two probability distributions.
Neural Module Networks (NMN)
Training

- Find module parameters maximizing likelihood of data.
- For each training example
  - A different network structure
  - An input image
  - An output label
- Networks with same high-level structure (layout) but different instances
  - What color is the cat?  
    - classify[color](attend[cat])
  - Where is the truck?  
    - classify[where](attend[truck])

Can be processed in the same batch
Training

- AdaDelta optimizer
  - Some weights are updated more frequently than others.
  - Adaptative per-weight learning rates

- Labels such as “dog” or “red” - attend[truck]
  - Just a notational convenience
  - No manual specification of behaviour
  - Behaviour is acquired as a byproduct of training.
Datasets

• VQA Dataset
  – Based on the COCO dataset
  – 200,000 natural images
    • Each with 3 questions and 10 answers per question.
  – Generated by human annotators
    • Larger and more natural than CocoQA
  – Training only with answers marked as “high confidence”
Datasets

- **SHAPES**
  - Synthetic images of simple arrangements of shapes
  - Good for compositionality
    - Questions in other datasets are too simple
      - Only one or two pieces of information needed
      - No evaluation of robustness
    - Questions in SHAPES
      - Between 2 and 4 attributes, object types, relations
  - 244 questions and 15,616 images
Experiments and Results

- **SHAPES**
  - LeNet for features
  - NMN easy
    - No size-6 questions

<table>
<thead>
<tr>
<th></th>
<th>size 4</th>
<th>size 5</th>
<th>size 6</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Majority</td>
<td>64.4</td>
<td>62.5</td>
<td>61.7</td>
<td>63.0</td>
</tr>
<tr>
<td>VIS+LSTM</td>
<td>71.9</td>
<td>62.5</td>
<td>61.7</td>
<td>65.3</td>
</tr>
<tr>
<td>NMN</td>
<td>89.7</td>
<td>92.4</td>
<td>85.2</td>
<td>90.6</td>
</tr>
<tr>
<td>NMN (easy)</td>
<td>97.7</td>
<td>91.1</td>
<td>89.7</td>
<td>90.8</td>
</tr>
</tbody>
</table>

- *Is there a red shape above a circle?*
Experiments and Results

- VQA
  - 16-layer VGGNet
  - conv5 layer

- **What color is his tie?**

<table>
<thead>
<tr>
<th></th>
<th>test-dev</th>
<th>test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Yes/No</td>
<td>Number</td>
</tr>
<tr>
<td>LSTM [2]</td>
<td>78.20</td>
<td>35.7</td>
</tr>
<tr>
<td>VIS+LSTM [2]</td>
<td>78.9</td>
<td>35.2</td>
</tr>
<tr>
<td>NMN</td>
<td>69.38</td>
<td>30.7</td>
</tr>
<tr>
<td>NMN+LSTM</td>
<td>77.7</td>
<td>37.2</td>
</tr>
</tbody>
</table>
## Experiments Results

<table>
<thead>
<tr>
<th></th>
<th>types</th>
<th># instances</th>
<th># layouts</th>
<th>max depth</th>
<th>max size</th>
</tr>
</thead>
<tbody>
<tr>
<td>VQA</td>
<td>attend, combine, classify, measure</td>
<td>1995</td>
<td>66549</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>SHAPES</td>
<td>attend, re-attend, combine, measure</td>
<td>8</td>
<td>164</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Image</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Image" /></td>
<td>how many different lights in various different shapes and sizes?</td>
</tr>
<tr>
<td><img src="image2" alt="Image" /></td>
<td>what is the color of the horse?</td>
</tr>
<tr>
<td><img src="image3" alt="Image" /></td>
<td>what color is the vase?</td>
</tr>
<tr>
<td><img src="image4" alt="Image" /></td>
<td>is the bus full of passengers?</td>
</tr>
<tr>
<td><img src="image5" alt="Image" /></td>
<td>is there a red shape above a circle?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Image</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image6" alt="Image" /></td>
<td>measure<a href="attend%5Blight%5D">count</a></td>
</tr>
<tr>
<td><img src="image7" alt="Image" /></td>
<td>classify<a href="attend%5Bhorse%5D">color</a></td>
</tr>
<tr>
<td><img src="image8" alt="Image" /></td>
<td>classify<a href="attend%5Bvase%5D">color</a></td>
</tr>
<tr>
<td><img src="image9" alt="Image" /></td>
<td>measure[is](combine[and](attend[bus], attend[full]))</td>
</tr>
<tr>
<td><img src="image10" alt="Image" /></td>
<td>measure[is](combine[and](attend[red], re-attend<a href="attend%5Bcircle%5D">above</a>))</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Image</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image11" alt="Image" /></td>
<td>four (four)</td>
</tr>
<tr>
<td><img src="image12" alt="Image" /></td>
<td>brown (brown)</td>
</tr>
<tr>
<td><img src="image13" alt="Image" /></td>
<td>green (green)</td>
</tr>
<tr>
<td><img src="image14" alt="Image" /></td>
<td>yes (yes)</td>
</tr>
<tr>
<td><img src="image15" alt="Image" /></td>
<td>yes (yes)</td>
</tr>
</tbody>
</table>
Issues

• Yes/No category performance
  – Model overfitting
    • Proposal: ensemble with sequence-only system
• Parser
  – Lots of room for improvement
  – Complex questions not parsed correctly
    • *Are these people most likely experiencing a work day?*
      – Parsed as be(people,likely)
      – Should be be(people,work)
    • Proposal: joint learning
## Issues

<table>
<thead>
<tr>
<th>Question</th>
<th>Scenario</th>
<th>Material</th>
<th>Is Clock</th>
<th>Is Red is Blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is stuffed with toothbrushes wrapped in plastic?</td>
<td>where does the tabby cat watch a horse eating hay?</td>
<td>what material are the boxes made of?</td>
<td>Is this a clock?</td>
<td>Is a red shape blue?</td>
</tr>
<tr>
<td>classify<a href="attend%5Bstuff%5D">what</a></td>
<td>classify<a href="attend%5Bwatch%5D">where</a></td>
<td>classify<a href="attend%5Bbox%5D">material</a></td>
<td>measure<a href="attend%5Bclock%5D">is</a></td>
<td>measure[is](combine[and](attend[red], attend[blue]))</td>
</tr>
<tr>
<td>container (cup)</td>
<td>pen (barn)</td>
<td>leather (cardboard)</td>
<td>yes (no)</td>
<td>yes (no)</td>
</tr>
</tbody>
</table>

- Plausible semantic confusions
- Normal Lexical Variations
- Irrelevant *a priori* plausible knowledge
Comments

• Appreciate the high-level style
  – Pleasant & easy read
  – Good comprehension of intuition and proposal
• Implementation remains unknown
  – Devil is in the details
• Clearly not the final paper
  – Several redaction errors
Thank you