Neural Module Networks

Sam & Nate
Neural Networks: Strengths and Weaknesses

Motivating questions/ discussion:

- What makes neural networks such a powerful class powerful learning algorithms? When do they approach or surpass human level?
Neural Networks: Strengths and Weaknesses

Motivating questions/ discussion:

- What makes neural networks such a powerful class powerful learning algorithms? When do they approach or surpass human level?
- What makes humans better than neural network approaches at some tasks? Which ones?
Neural Networks: Strengths and Weaknesses

Motivating questions/ discussion:

- What makes neural networks such a powerful class powerful learning algorithms? When do they approach or surpass human level?
- What makes humans better than neural network approaches at some tasks? Which ones? E.g. inferring rich semantics from visual scene, learning from very sparse data, etc...
  - Generalization/ transfer?
  - Structured or compositional thinking abilities?
  - Something else?
Neural Networks: Strengths and Weaknesses

Motivating questions/ discussion:

● What makes neural networks such a powerful class powerful learning algorithms? When do they approach or surpass human level?
● What makes humans better than neural network approaches at some tasks? Which ones? E.g. inferring rich semantics from visual scene, learning from very sparse data, etc...
  ○ Generalization/ transfer?
  ○ Structured or compositional thinking abilities?
  ○ Something else?
● For people interested in modeling human cognition: what makes neural networks useful as models of human thought, what makes them less helpful?
Neural Networks: Strengths and Weaknesses

Motivating questions/ discussion:

● What makes neural networks such a powerful class powerful learning algorithms? When do they approach or surpass human level?
● What makes humans better than neural network approaches at some tasks? Which ones? E.g. inferring rich semantics from visual scene, learning from very sparse data, etc...
   ○ Generalization/ transfer?
   ○ Structured or compositional thinking abilities?
   ○ Something else?
● For people interested in modeling human cognition: what makes neural networks useful as models of human thought, what makes them less helpful?
   ○ One idea: greater built-in modularity to neural networks might make them more tractable as “process models”.
Structured Probabilistic Inference

- A classic approach in AI and cognitive science, sometimes called “Good Old Fashioned Artificial Intelligence” (GOFAI), is based on finding rules that describe the structure of the world.
Structured Probabilistic Inference

- A classic approach in AI and cognitive science, sometimes called “Good Old Fashioned Artificial Intelligence” (GOFAI), is based on finding rules that describe the structure of the world.
- These approaches have segued into (usually) Bayesian accounts of inference over symbols.
Structured Probabilistic Inference

- A classic approach in AI and cognitive science, sometimes called “Good Old Fashioned Artificial Intelligence” (GOFAI), is based on finding rules that describe the structure of the world.
- These approaches have segued into (usually) Bayesian accounts of inference over symbols.
- A strong argument in favor is the ease of compositionality - useful for problems in language, reasoning, etc...
Structured Probabilistic Inference

- A classic approach in AI and cognitive science, sometimes called “Good Old Fashioned Artificial Intelligence” (GOFAI), is based on finding rules that describe the structure of the world.
- These approaches have segued into (usually) Bayesian accounts of inference over symbols.
- A strong argument in favor is the ease of compositionality - useful for problems in language, reasoning, etc....
Emergentist Approach (Deep Learning)

- Structured approaches are usually not as capable at pattern recognition as neural network approaches.
Emergentist Approach (Deep Learning)

- Structured approaches are usually not as capable at pattern recognition as neural network approaches.
- Generally, have capacity to represent “deep” structure to problems, not easily captured by symbols.
Emergentist Approach (Deep Learning)

- Structured approaches are usually not as capable at pattern recognition as neural network approaches.
- Generally, have capacity to represent “deep” structure to problems, not easily captured by symbols.
Visual Question Answering

- The common techniques fall into these two camps.
  - Structured symbolic: use semantic parsers to decompose questions into logical expression.
  - Deep learning: use bag of words (or more complicated) to represent question, train a classifier over the image and question simultaneously.
Neural Module Networks

- An attempt to get the best of both structured and emergentist approaches.
Neural Module Networks

- An attempt to get the best of both structured and emergentist approaches.

- Two observations:
  - There is no one best neural network architecture or learning algorithm for all tasks (that we know of).
  - It is often helpful to use pre-trained network and then “fine-tune”.
Neural Module Networks

- An attempt to get the best of both structured and emergentist approaches.
- Two observations:
  - There is no one best neural network architecture or learning algorithm for all tasks (that we know of).
  - It is often helpful to use pre-trained network and then “fine-tune”.
  - **Conclusion**: neural networks are empirically modular. Intermediate representations are useful for different purposes.
Neural Module Networks

● An attempt to get the best of both structured and emergentist approaches.

● Two observations:
  ○ There is no one best neural network architecture or learning algorithm for all tasks (that we know of).
  ○ It is often helpful to use pre-trained network and then “fine-tune”.
  ○ **Conclusion**: neural networks are empirically modular. Intermediate representations are useful for different purposes.

● Different kinds of processing might be involved.
  ○ Example: convolutions might be useful for object identification, but recurrence might be useful for counting.
Neural Module Networks

- Neural network architecture built on “modules”, which are:
  - Independent
  - Composable
  - Well-typed
Neural Module Networks

- Neural network architecture built on “modules”, which are:
  - Independent
  - Composable
  - Well-typed
- It makes sense to not have a fixed architecture to solve every problem. Best structure, components, might vary between problems.
- Consider: “is there a television?” versus “how many objects are resting on top of the television?”
General Approach

Steps

- First analyze each question with a semantic parser.
- The output of the semantic parser is then used to determine which “modules” to use.
- Modules are assembled and then jointly-trained.
Types of Modules

- Three input/output types: images, attentions, and labels.
Types of Modules

Find

Image $\rightarrow$ Attention
Types of Modules

Find

Image → Attention

Combine

Attention × Attention → Attention
Types of Modules

Find

\[ \text{Image} \rightarrow \text{Attention} \]

Combine

\[ \text{Attention} \times \text{Attention} \rightarrow \text{Attention} \]

Transform

\[ \text{Attention} \rightarrow \text{Attention} \]
Types of Modules

Find

Transform

Image $\rightarrow$ Attention

Combine

Attention $\times$ Attention $\rightarrow$ Attention

Describe

Image $\times$ Attention $\rightarrow$ Label
Types of Modules

Find

\[ \text{Image} \rightarrow \text{Attention} \]

- find[red]
  - Convolution

Combine

\[ \text{Attention} \times \text{Attention} \rightarrow \text{Attention} \]

- combine[or]
  - Stack
  - Conv
  - ReLU

Transform

\[ \text{Attention} \rightarrow \text{Attention} \]

- transform[above]
  - FC
  - ReLU

Describe

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

- describe[color]
  - Attend
  - FC

Measure

\[ \text{Attention} \rightarrow \text{Label} \]

- measure[be]
  - FC
  - ReLU
  - FC
  - Softmax

- yes
From strings to networks

- Need to assemble the layout based on the input question
From strings to networks

● Need to assemble the layout based on the input question
● Uses the Stanford Parser with basic lemmatization
  ○ “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))
From strings to networks

- Need to assemble the layout based on the input question
- Uses the Stanford Parser with basic lemmatization
  - “What is standing in the field?” \(\rightarrow\) what(stand)
  - “What color is the truck?” \(\rightarrow\) color(truck)
  - “Is there a circle next to a square?” \(\rightarrow\) is(circle, next-to(square))
- Create tree based on parsing
  - “What color is the tie?” \(\rightarrow\) describe[color](find[tie])
Answering Natural Language Questions

- Utilizes a simple LSTM question encoder
  - Simplifying the question discards important information. E.g., what is versus what are.
  - Allows for reasonable guesses based purely on the question
Answering Natural Language Questions

- Utilizes a simple LSTM question encoder
  - Simplifying the question discards important information. E.g., what *is* versus what *are*.
  - Allows for reasonable guesses based purely on the question

- Output of the encoded is then added to the NMN
  - Elementwise ReLU
Answering Natural Language Questions

● Utilizes a simple LSTM question encoder
  ○ Simplifying the question discards important information. E.g., what is versus what are.
  ○ Allows for reasonable guesses based purely on the question

● Output of the encoded is then added to the NMN
  ○ Elementwise ReLU

● Final output is a softmax over the set of answers seen during training
Testing Compositionality

- Created a dataset called **SHAPES** to test on synthetic data
  - 64 images
  - 244 unique questions
  - All answers are yes-or-no*
Testing Compositionality

- Created a dataset called SHAPES to test on synthetic data
  - 64 images
  - 244 unique questions
  - All answers are yes-or-no*
- Necessary but not sufficient for robust visual QA
Testing Compositionality

describe(count)(find[light]))
describe(color)(find[horse]))
describe(color)(find[wave]))
describe(is)(combine[and](find[bus], find[full]))
measure(is)(combine[and](find[red], transform[above](find[circle])))

four (four)
brown (brown)
green (green)
yes (yes)
yes (yes)

describe(what)(find[stuff]))
describe(where)(find[watch]))
describe(material)(find[box]))
describe(is)(find[clock]))
measure(is)(combine[and](find[red], find[blue])))

container (cup)
pen (pens)
leather (cardboard)
yes (no)
yes (no)

<table>
<thead>
<tr>
<th>% of test set</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 (train size ≤ 5)</td>
<td>97.7</td>
<td>91.1</td>
<td>89.7</td>
<td>90.8</td>
</tr>
</tbody>
</table>
Testing On Natural Images

- Used the VQA dataset
  - More than 200,000 images from MSCOCO
  - Each paired with three questions and ten answers per question
Testing On Natural Images

- Used the VQA dataset
  - More than 200,000 images from MSCOCO
  - Each paired with three questions and ten answers per question
- Input layer to the NMN was the conv5 layer of VGG16
  - Additionally tried fine-tuning VGG16 to MSCOCO
Testing On Natural Images

- Used the VQA dataset
  - More than 200,000 images from MSCOCO
  - Each paired with three questions and ten answers per question
- Input layer to the NMN was the conv5 layer of VGG16
  - Additionally tried fine-tuning VGG16 to MSCOCO

<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</td>
<td>78.7</td>
<td>36.6</td>
</tr>
<tr>
<td>VIS+LSTM [3]</td>
<td>78.9</td>
<td>35.2</td>
</tr>
<tr>
<td>ATT+LSTM</td>
<td>80.6</td>
<td>36.4</td>
</tr>
<tr>
<td>NMN</td>
<td>70.7</td>
<td>36.8</td>
</tr>
<tr>
<td>NMN+LSTM</td>
<td>81.2</td>
<td>35.2</td>
</tr>
<tr>
<td>NMN+LSTM+FT</td>
<td>81.2</td>
<td>38.0</td>
</tr>
</tbody>
</table>
Conclusions

- The parser has room for improvement
  
  - “Are these people most likely experiencing a work day?”
    - Should be: be(people, work)
    - Was: be(people, likely)
  
  - Hand inspection suggests 80-90% of questions parsed correctly for **simple** questions
Conclusions

- The parser has room for improvement
  - “Are these people most likely experiencing a work day?”
    - Should be: be(people, work)
    - Was: be(people, likely)
  - Hand inspection suggests 80-90% of questions parsed correctly for *simple* questions

- The system works
  - Points to a paradigm of “programs” built from neural networks
Limitations

- No need to do inference over architecture, weights separately.
- Still uses supervised learning.
- Types pretty restricted.