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

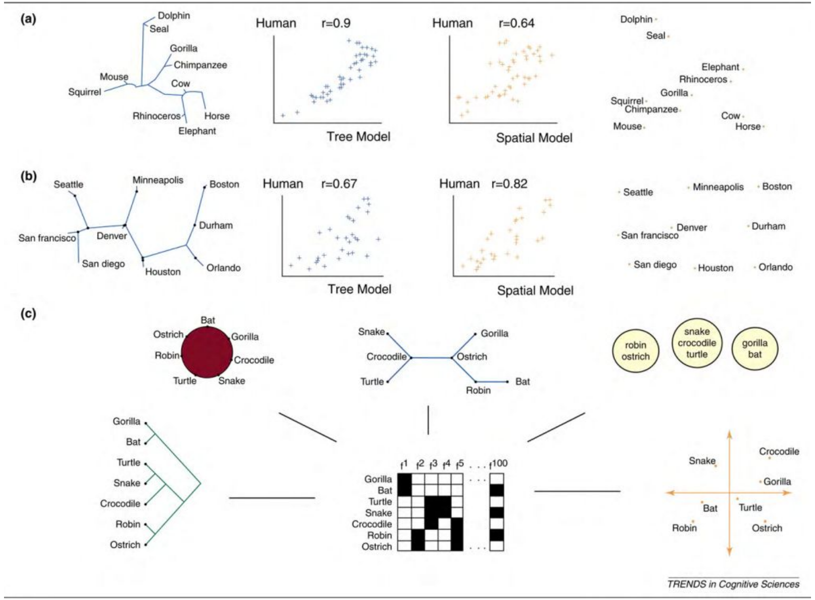
- A classic approach in AI and cognitive science, sometimes called “Good Old Fashioned Artificial Intelligence” (GOFAI), is based on finding rules that describe the structure of the world.
- These approaches have segued into (usually) Bayesian accounts of inference over symbols.

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)

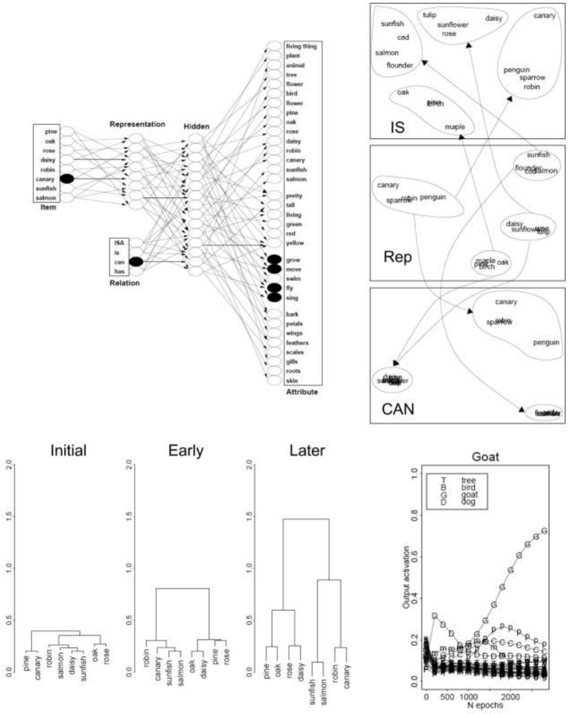
- Structured approaches are usually not as capable at pattern recognition as neural network approaches.

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

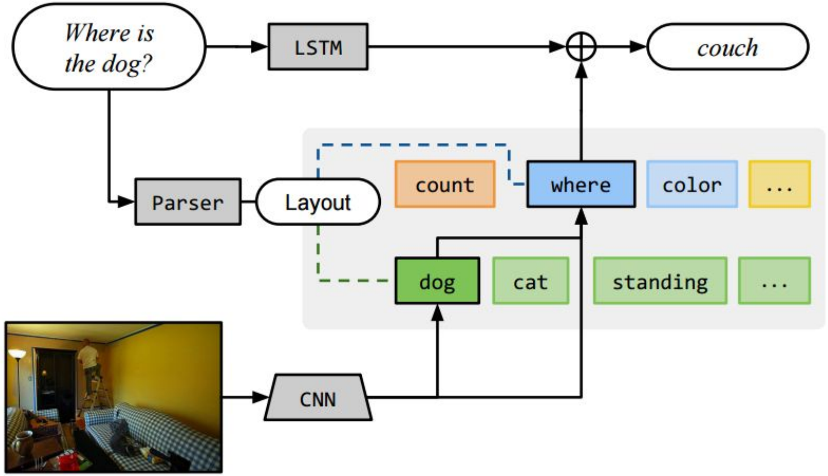
Neural Module Networks

- 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.

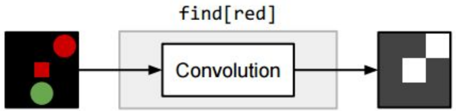


Types of Modules

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

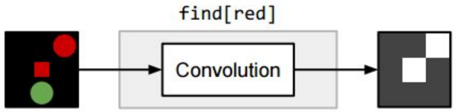
Types of Modules

Find $Image \rightarrow Attention$

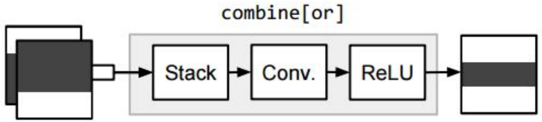


Types of Modules

Find $Image \rightarrow Attention$

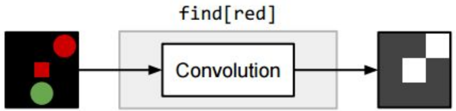


Combine $Attention \times Attention \rightarrow Attention$

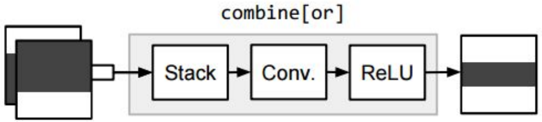


Types of Modules

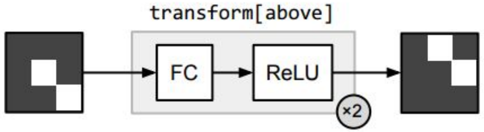
Find $\quad Image \rightarrow Attention$



Combine $\quad Attention \times Attention \rightarrow Attention$

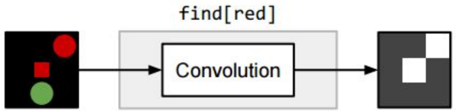


Transform $\quad Attention \rightarrow Attention$

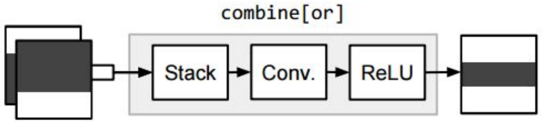


Types of Modules

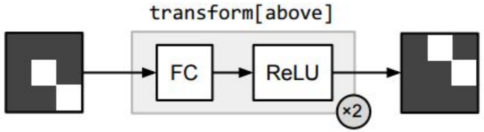
Find $Image \rightarrow Attention$



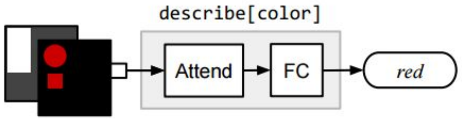
Combine $Attention \times Attention \rightarrow Attention$



Transform $Attention \rightarrow Attention$

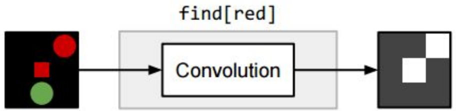


Describe $Image \times Attention \rightarrow Label$

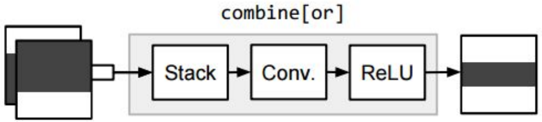


Types of Modules

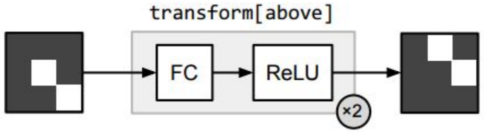
Find $Image \rightarrow Attention$



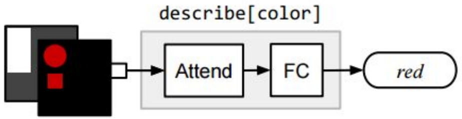
Combine $Attention \times Attention \rightarrow Attention$



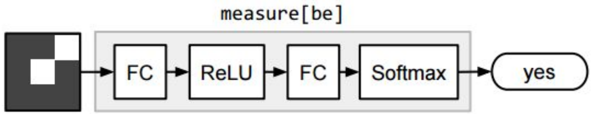
Transform $Attention \rightarrow Attention$



Describe $Image \times Attention \rightarrow Label$



Measure $Attention \rightarrow Label$



From strings to networks

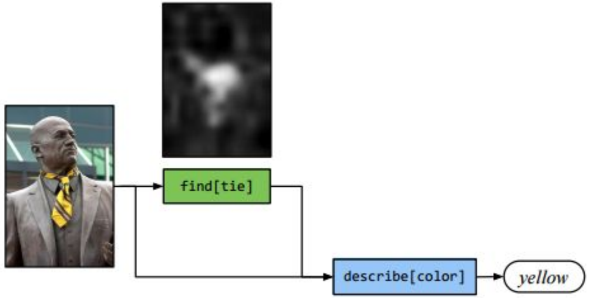
- Need to assemble the layout based on the input question

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?” ➔ `what(stand)`
  - “What color is the truck?” ➔ `color(truck)`
  - “Is there a circle next to a square?” ➔ `is(circle, next-to(square))`
- Create tree based on parsing
  - “What color is the tie?” ➔ `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

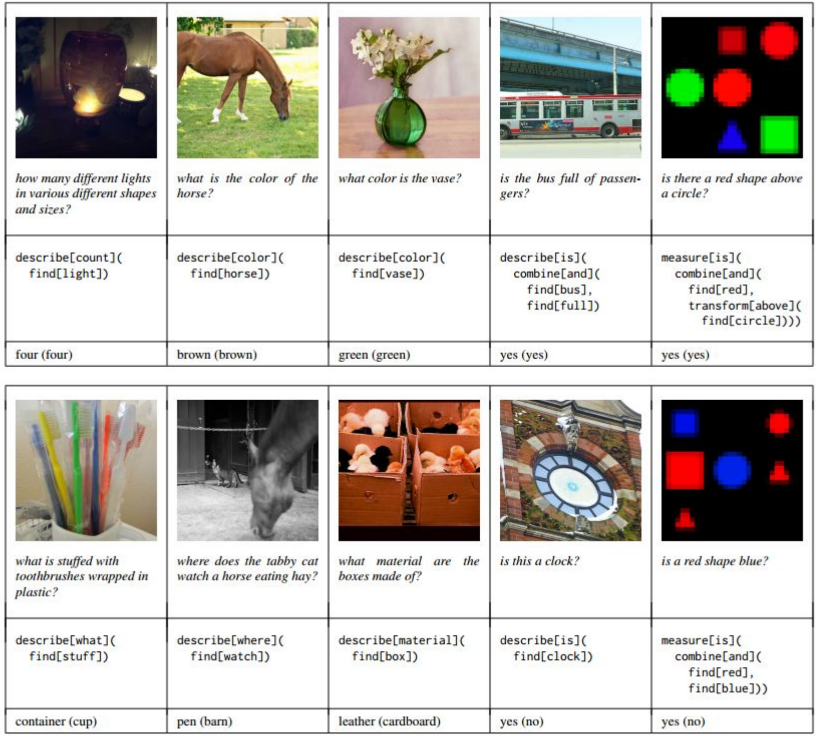
Testing Compositionality

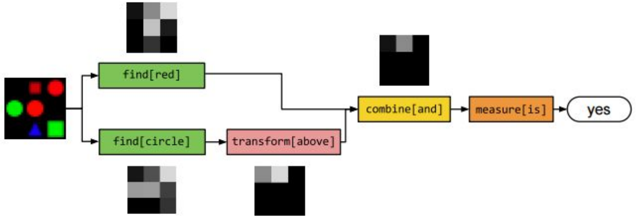
- Created a dataset called `SHAPES` to test on synthetic data
  - 64 images
  - 244 unique questions
  - All answers are yes-or-no*

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





	size 4	size 5	size 6	All
% of test set	31	56	13	
Majority	64.4	62.5	61.7	63.0
VIS+LSTM	71.9	62.5	61.7	65.3
NMN	89.7	92.4	85.2	**90.6**
NMN (train size ≤ 5)	97.7	91.1	89.7	**90.8**

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

	test-dev				test
	Yes/No	Number	Other	All	All
LSTM	78.7	36.6	28.1	49.8	–
VIS+LSTM [3] [2]	78.9	35.2	36.4	53.7	54.1
ATT+LSTM	80.6	36.4	42.0	57.2	–
NMN	70.7	36.8	39.2	54.8	–
NMN+LSTM	81.2	35.2	43.3	58.0	–
NMN+LSTM+FT	81.2	38.0	44.0	58.6	**58.7**

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.