Paper Presentation

Deep Compositional Question Answering with Neural Module Networks

Reporter: Xuan Chen

Student #82286956

List

- Problems being solved
- Why is it valuable?
- Contributions
- Model architecture
- Experiments
- Conclusions.

Problems being solved

- VQA (visual question answering) model: monolithic network & Neural modular network
- Neural Module Networks (NMNs), is a framework for modular, composable, jointly-trained neural networks. A network model is tailored to each question in the VQA dataset.
- Enhanced interpretability of the network

Why is it valuable?

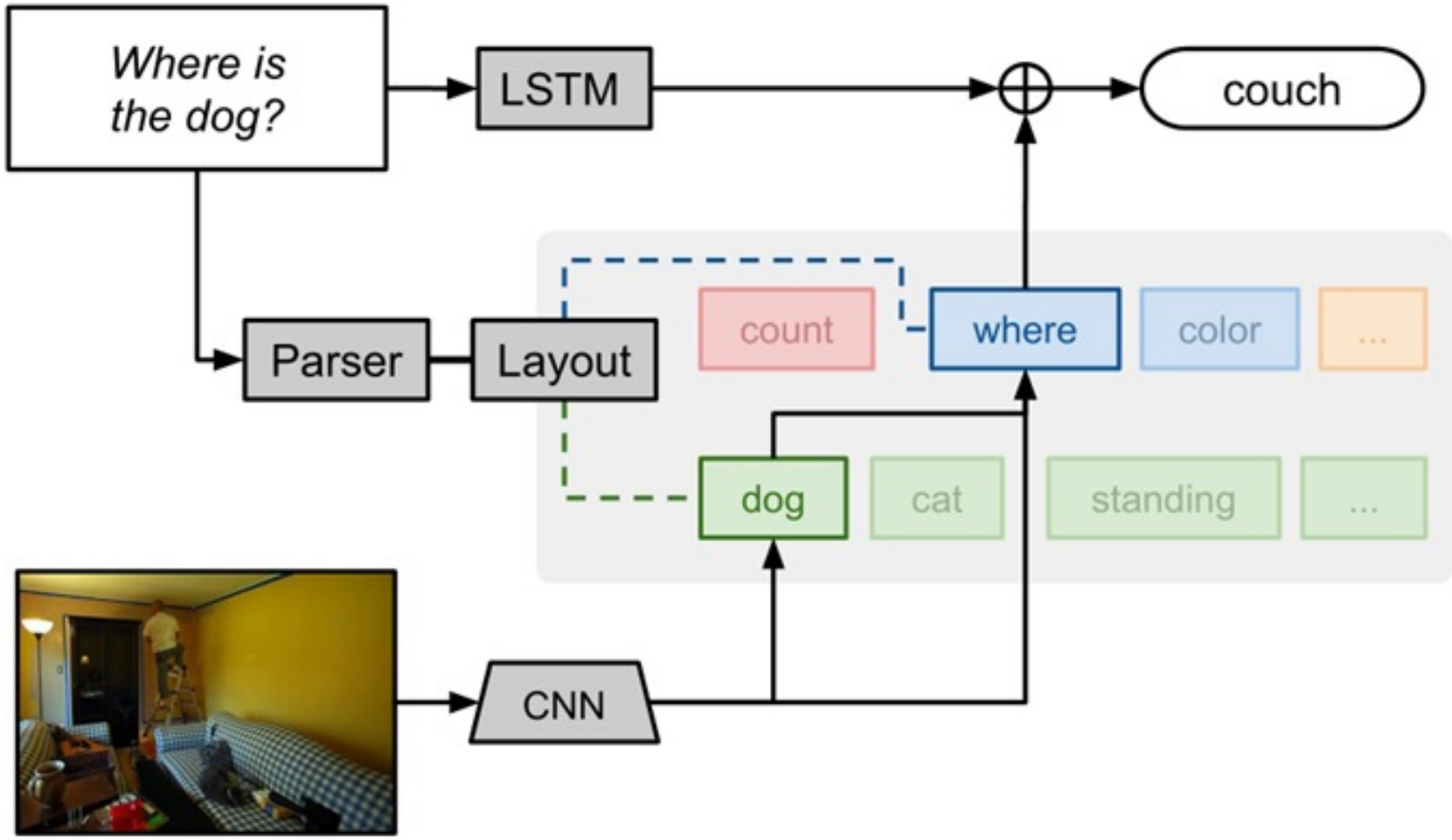
- There is no single “best network” for all computer vision tasks.
- Thinking of question answering as a highly-multitask learning setting, where each problem instance is associated with a novel task, and the identity of that task is expressed only noisily in language.

- Problem-dependent and dynamic.
- Intuitive, interpretable model, and transparent process.

Main contributions

- It first describes an approach to visual question answering based on neural module networks (NMNs). It answers natural language questions about images using collections of jointly-trained neural “modules”, dynamically composed into deep networks based on linguistic structure.
- The NMN performs well on answering object or attribute questions.
- Introduction of SHAPE dataset.

Model architecture



Part 1: Modules

- Three basic data types: **images, unnormalized attentions, and labels.**
- Form: *TYPE [INSTANCE] (ARG1,…)*

TYPE: a high-level module type (attention, classification, etc.) of the kind described in this section.

INSTANCE: the particular instance of the model under consideration.

- attend[red]: locates red things
- attend[dog]: locates dogs.

Part 1: Modules

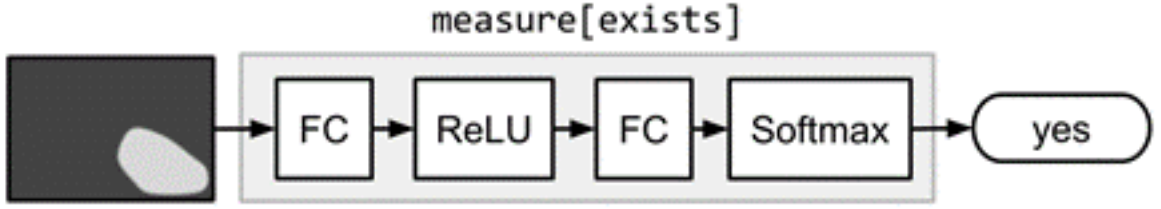
Attention

Re-attention

Measurement

measure : *Attention* → *Label*

measure[exists]



Combination

combine : *Attention* × *Attention* → *Attention*

Part 2: From strings to networks

- **Parsing:** Using Stanford Parser, extract grammatical relations between parts of a sentence, and generate abstraction; performs basic lemmatization.
- **Layout:** Based on specific tasks, converts symbolic representations into modular network structure.

Leaves: attend modules.

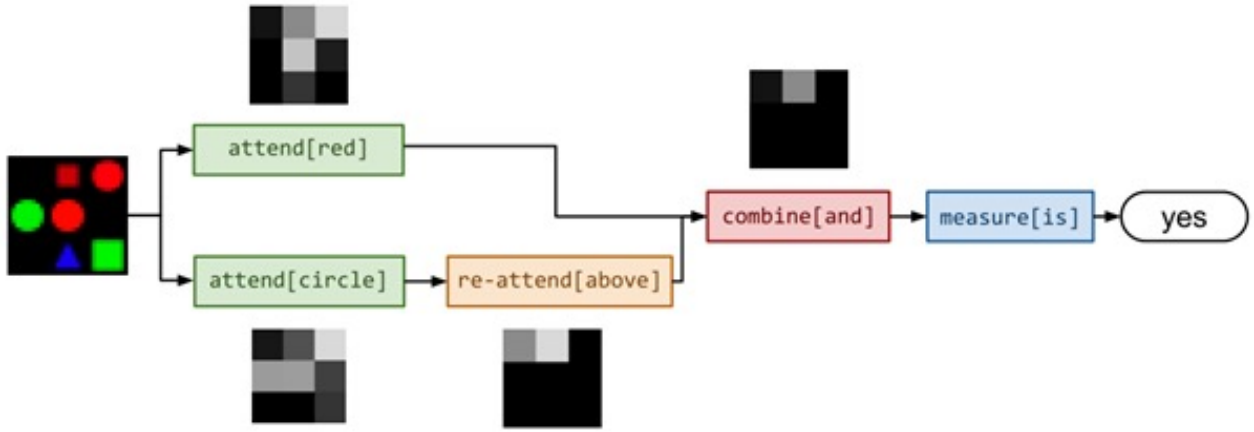
Internal nodes: re-attend modules/combine modules.

Root nodes: measure modules for yes/no questions and classify modules for all other questions.

- **Generalizations:** provide sentences, or even direct query statements like sql.

Sample NMNs for question answering

(a) NMN for answering the question *What color is his tie?*



(b) NMN for answering the question *Is there a red shape above a circle?*

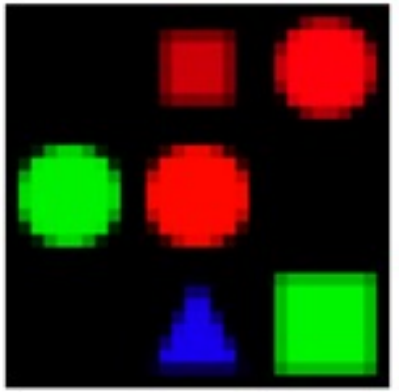
Experiments

- Results on different datasets: SHAPE dataset(up), VQA dataset(down).

	size 4	size 5	size 6	All
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 (easy)	97.7	91.1	89.7	**90.8**

	test-dev				test
	Yes/No	Number	Other	All	All
LSTM [2]	78.20	35.7	26.6	48.8	–
VIS+LSTM [2]	78.9	35.2	36.4	53.7	54.1
NMN	69.38	30.7	22.7	42.7	–
NMN+LSTM	77.7	37.2	39.3	54.8	**55.1**

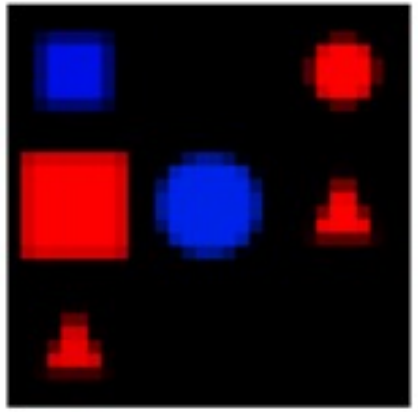
Correct cases:

				
how many different lights in various different shapes and sizes?	*what is the color of the horse?*	*what color is the vase?*	*is the bus full of passengers?*	*is there a red shape above a circle?*

`measure[count](`
`  attend[light])`
	`classify[color](`
	`  attend[horse])`
		`classify[color](`
		`  attend[vase])`
			`measure[is](`
			`  combine[and](`
			`  attend[bus],`
			`  attend[full])`
				`measure[is](`
				`  combine[and](`
				`  attend[red],`
				`  re-attend[above](`
				`  attend[circle])))`

four (four)	brown (brown)	green (green)	yes (yes)	no (no)

Wrong cases:

				
what is stuffed with toothbrushes wrapped in plastic?	*where does the tabby cat watch a horse eating hay?*	*what material are the boxes made of?*	*is this a clock?*	*is a red shape blue?*

`classify[what](`
`  attend[stuff])`
	`classify[where](`
	`  attend[watch])`
		`classify[material](`
		`  attend[box])`
			`measure[is](`
			`attend[clock])`
				`measure[is](`
				`  combine[and](`
				`  attend[red],`
				`  attend[blue]))`

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

Conclusion

- Previous work: focus on attention, but the process of question answering can not be explained properly.
- This paper: introduces Neural Module Networks, which is made up with several modular networks. It is customized by every question in VQA dataset (dynamically).
- Generate a more standard neural network → construct models → complex reasoning tasks