Deep Compositional Question Answering with Neural Module Networks

Anonymous CVPR submission

Paper ID ****

Abstract

We propose a modular approach to question-answering tasks grounded in images. Complex questions are broken into their linguistic substructures, each of which is operationally grounded with an appropriately modularized neural network fragment. For example the network for black cat *might intersect a cat detector with a color detector. The resulting compound networks are then jointly trained, yielding composable neural "modules" that can be dynamically remixed to interpret novel questions. We apply our approach to both new and established datasets for visual question answering, achieving several state-of-the-art results.*

1. Introduction

We often want to answer complex questions about unstructured data like images (Fig XXX). While semantic parsers from the natural language processing literature can reliably convert questions into logical expressions, such expressions must typically be evaluated against structured knowledge bases, and are unhelpful for visual question answering. Neural-net classifiers, meanwhile, are designed to recognize simple entities and attributes in continuous signals, but not to compose them in structured ways.

This paper presents a general-purpose technique for integrating the representational power of neural networks with the flexible compositional structure afforded by symbolic approaches to semantics. Where previous work has treated both the image and the question as inputs to a monolithic classification model, we instead take the perspective that a question is a noisy specification of a hidden computation that must be performed on the image to produce an answer. Crucially, this computation may be different for each problem instance, and is never observed observed during training.

Our approach bears a superficial resemblance to a classical semantic parser. However, instead of mapping from questions to logical forms, our model maps from questions to neural network structures. These networks are assembled on the fly (possibly into novel topologies) from a collection of jointly-learned neural "modules". Finally, they are evaluated against the input image to produce an answer.

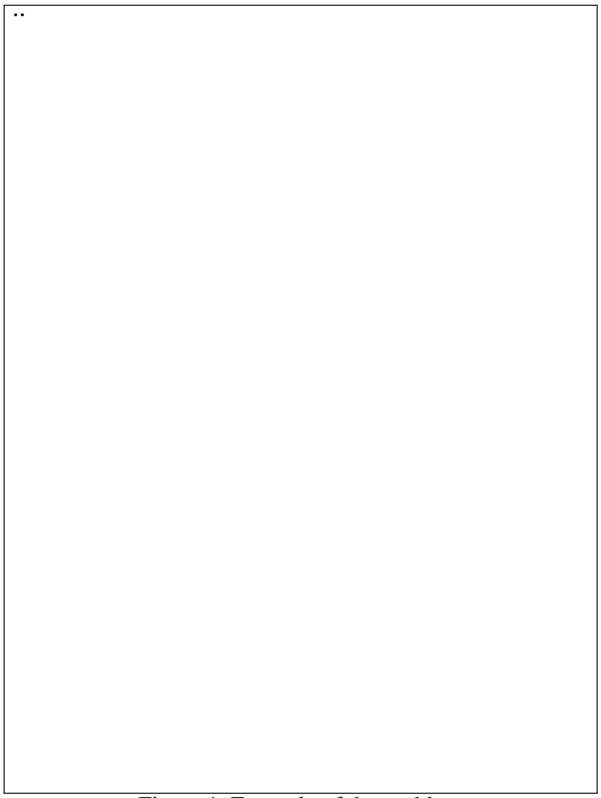


Figure 1. Example of the problem

We evaluate our approach on three visual question answering tasks. On the recently-released CocoQA and VQA datasets, we achieve results comparable to [better than] existing approaches, and show that it specifically outperforms previous work on questions with compositional structure [e.g. requiring that an object be located and one of its attributes described]. However, most of the questions in both datasets are quite simple, involving little or no composition. To test our approach's ability to handle highly structured questions, we introduce a new dataset of synthetic images paired with complex questions involving spatial relations, logical operators, and shape and attribute recognition. On

this dataset we outperform the previous state of the art by XXX.

The contributions of this work are twofold. First, we demonstrate that off-the-shelf tools for linguistic structure prediction can be used to inform decisions about the structure of neural networks. Second, and more generally, we demonstrate that it is possible to train a collection of neural modules in such a way that they can be assmbled into novel topologies at evaluation time.

2. Motivations

Many tasks in computer vision, including recognition, detection, and captioning, share common substructure. For example, we might schematically express the sequence of computations performed by a recognizer as

```
classify(pickMostRelevant(detectObjects))
```

or a detector as

```
drawBoundaries(detectObjects)
```

In practice the picture is not this clean—classification or detection is performed end-to-end by a single neural network, and the boundaries between these "phases" are not clearly defined. Nevertheless we might expect *a priori* that a network used for classification might expose intermediate representations useful for building a detector. Indeed, it is now commonplace to initialize systems for a variety of vision tasks with a prefix of a network trained for classification [4]. This has been shown to substantially reduce training time and improve accuracy. So while network structures are not *universal* (in the sense that the same network is appropriate for all problems), they are at least empirically *modular* (in the sense that intermediate representations for one task are useful for many others).

Can we generalize this idea in a way that is useful for question answering? Rather than thinking of question answering as a problem of learning a single function to map from questions and contexts to answers, it's perhaps useful to think of it 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. If we consider a few examples of questions:

how many black cats?	`count(and(cat, black))`
what color is the cat?	`color(cat)`
what color is the dog?	`color(dog)`
is there a dog?	`exists(dog)`

we again see that there is common computational substructure involved in solving the associated tasks. With sub-networks for computing `cat`, `classify`, `count`, etc., we can in principle answer questions with novel structure like—e.g. *is there a black dog?*—without any additional training data.

Note in particular that we expect these modules to differ not only in their parameters, but more fundamentally in their topologies. Intuitively, `cat` should take an image as input, perform some fully-convolutional operation, and output an attention (understood as a distribution over positions in the image), while `color` should take both the input image and such an attention, and map to a distribution over labels.

3. Model

For the experiments in this paper, we assume access to a pre-trained semantic which converts each natural language sentence into a simple structured representation of meaning, as in Section XXX. Thus each training datum for this task can be thought of as a 4-tuple (w, p, x, y), where

- w is a natural-language question
- p is a semantic parse (produced by the parser)
- x is an image
- y is an answer

Our goal is to learn parameters θ for a probability distribution $p(y|w, p, x; \theta)$ which maps from an input question and image to a distribution over answers. In the remainder of this section, we describe preprocessing steps applied to images and queries, and then describe how these are used to compute the final dsitribution $p(y|w, p, x)$.

3.1. Queries

As noted above, we assume access to a pre-trained semantic parser. Such parsers can easily be learned from existing datasets for semantic or even syntactic parsing; details of the parser for each experiment are given in the relevant portion of the experiments section.

XXX something about the type system.

XXX something about combinator logics / leaves of the tree.

Thus, given a question like *how many cats are black?* with logical form `count(and(cat, black))`, we wish to convert it into a generic network layout of the form `count(and(detect[cat], detect[black]))`, where brackets indicate XXX. The mapping from the functions `detect` and `black` can be inferred directly from the types assigned to each of these functions, so the only human-specified knowledge required by any part of the system is the extremely compact mapping from types to modules, specified fully here:

entity $\rightarrow$ truth	detect
truth $\rightarrow$ truth	redetect
truth $\rightarrow$ property	classify

3.2. Images

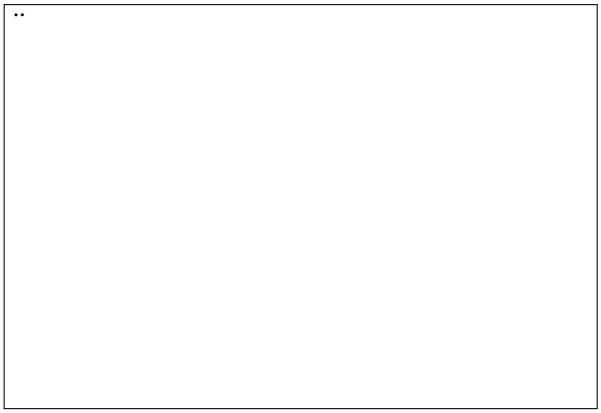
We preprocess each image in the dataset with the first K layers of VGGNet [CITE].

After both of these preprocessing steps, each training datum consists of (a string, a query, an image, and an answer).

3.3. Modules

Our goal in this section is to identify a small set of modules that can be assembled into all the configurations necessary for our tasks. This corresponds to identifying a minimal set of composable vision primitives. While others may need to be invented in the future, we use the following for the tasks described in this paper.

ATTEND



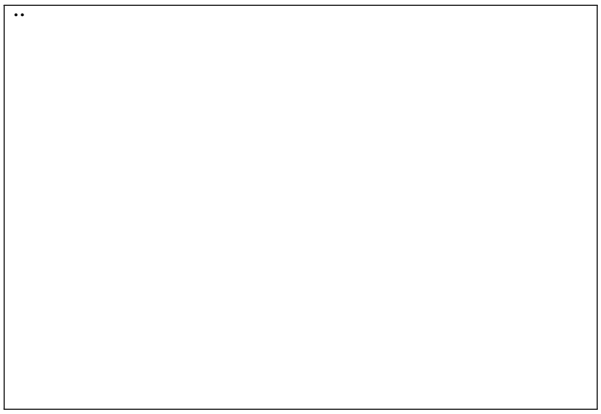
RE-ATTEND

COMBINE

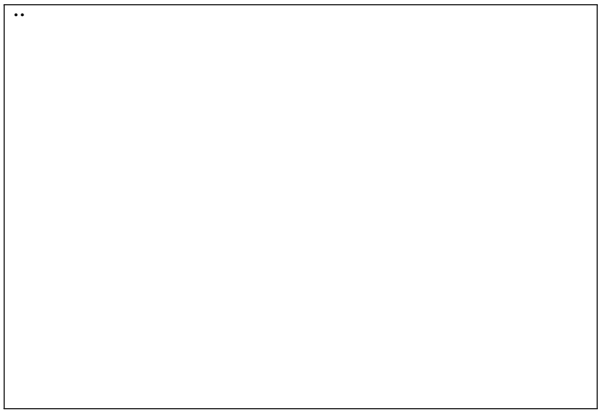
System	All	Object	Location	Color	Count
NMN					
Others					

System	All	Object	Location	Color	Count
NMN					
Others					

System	All	Depth2	Depth3	Depth4
NMN				
Others				



CLASSIFY



3.4. Networks

3.5. Strings

4. Learning

5. Experiments: VQA

6. Experiments: COCO-QA

7. Experiments: Synthetic data

8. Related work

The neural module networks we have described have a number of close cousins in the neural network literature. Standard recurrent neural networks [1] can be viewed as a special case where all modules are of the same type, arranged in a sequence. Similarly, this approach may be thought of as a generalization of recursive neural networks [5] with heterogenous network fragments at each node. As noted above, there is a large literature on learning to answer questions about structured knowledge representations from question–answer pairs, both with and without learning of meanings for individual predicates [3, 2].

9. Conclusions and future work

In this paper we have maintained a strict separation between predicting network structures and learning network parameters. It is easy to imagine that these two problems might be solved jointly, with uncertainty maintained over network structures throughout training and decoding.

The fact that deep neural modules can be trained to produce predictable outputs—even when freely composed—points toward a more general paradigm of "programs" built from neural networks. In this paradigm, network designers (human or automated) have access to a standard kit of neural parts from which to construct models for performing complex reasoning tasks. While visual question answering provides a natural testbed for this approach, its usefulness is potentially much broader, extending from queries about documents and structured knowledge bases to general signal processing and function approximation.

References

[1] J. L. Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990. 4

[2] J. Krishnamurthy and T. Kollar. Jointly learning to parse and perceive: connecting natural language to the physical world. *Transactions of the Association for Computational Linguistics*, 2013. 4

[3] P. Liang, M. I. Jordan, and D. Klein. Learning dependency-based compositional semantics. *Computational Linguistics*, 39(2):389–446, 2013. 4

[4] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *arXiv preprint arXiv:1411.4038*, 2014. 2

[5] R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng. Parsing with compositional vector grammars. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2013. 4