DeepWhat?!

End-to-end models for task-oriented visual dialogue with reinforcement

Charles Lu (clu8) and Eli Wu (eliwu)
CS231N Spring 2017

Introduction

Systems with the ability to communicate naturally with humans have been one of the top goals of artificial intelligence research. Most research thus far generally fails to fully take into account past dialogue in the conversation as well as the context in which the conversation is happening. This limits the context in which these methods can respond.

DeepWhat?! is a system that attempts to tackle these issues by playing the GuessWhat?! game, trained on a dataset introduced in 2016 and fine-tuned with reinforcement learning.

Problem

This project attempts to complete several tasks simultaneously— improving goal-driven dialogue generation, question answering, as well as object detection. These problems are encapsulated in the GuessWhat?! game, requiring the system to generate questions from images, generate answers from questions and images, and ultimately find objects within the images from questions, answers, and images.

Questioner:	Oracle:
Is it an orange?	Yes
Is it the one on top?	No
Is it the one in the center?	No
Is it the one on bottom?	Yes

Guesser:
<Choose bottom orange>

An example of a GuessWhat?! game being played by our model(s).

GuessWhat?! Is an interactive two player game in which players focus on an object within a rich natural image.

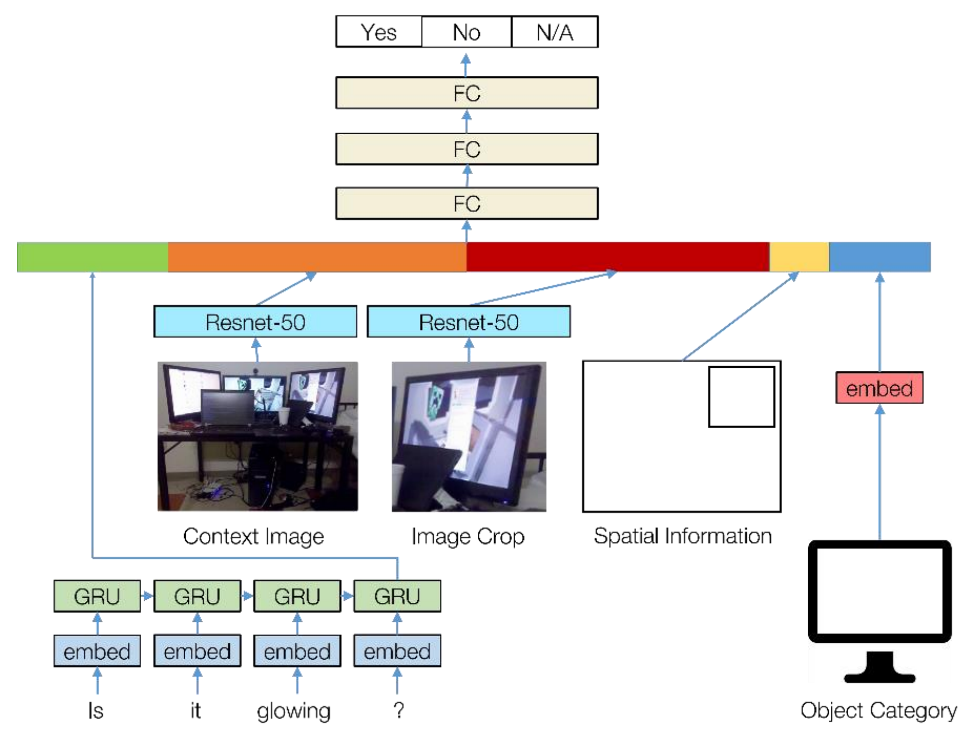
The "oracle" is assigned a random object in the image, and is tasked with answering the yes or no questions of the "questioner". The questioner is tasked with finding the object by asking the oracle yes or no questions about the image.

At a glance:

- 150K games
- 800K question-answer pairs
- 66K images

Methodology

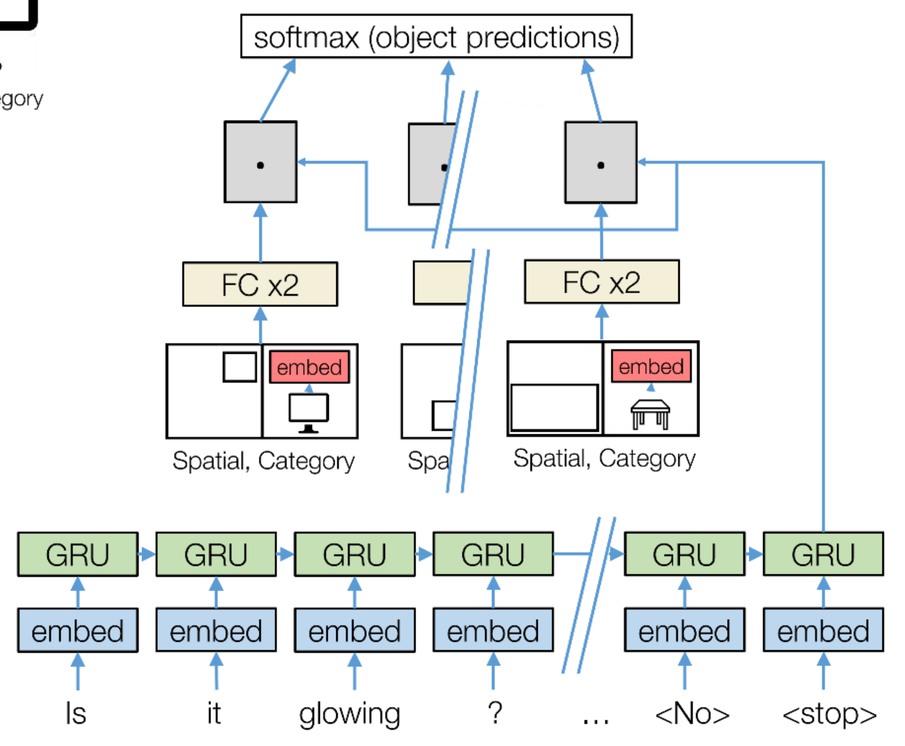
In order to play GuessWhat?!, we split the problem into three parts, and utilize a different model for each component.



Oracle Model

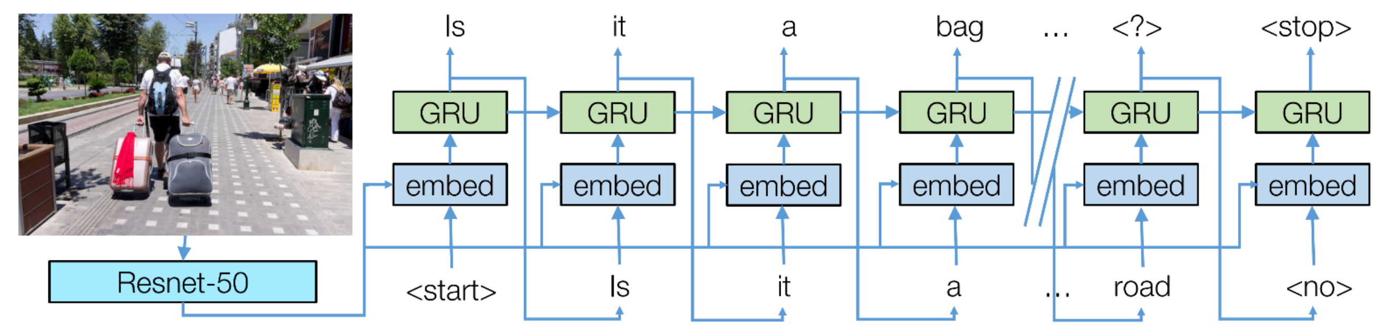
The **oracle** model takes as input the image, and correct object crop and spatial information. It returns, then, the answer to the question – yes, no, or N/A.

The **guesser** model takes as input the dialogue (question/answer pairs), as well as the spatial and category information of the objects in the image. It returns the predictions for the correct object.



Guesser Model

The **questioner** model attempts to ask questions to lead to the correct guess. It takes as input an image, and produces a dialogue (some questions) as output. During training, we feed the ground truth questions as input; during evaluation, we implement sampling and beam search.



Questioner Model

Reinforcement learning fine-tuning. The GuessWhat?! game can be framed as a Markov decision process - the state at some point in the game is a tuple containing the image, all previous question-answer pairs, and all words already generated in the current question utterance. The question generation model therefore represents a stochastic policy.

We fix the trained weights for the oracle and guesser to fine-tune the questioner model. Each state-action pair is given a zero-one reward, which is only 1 when $t = T$ and the guesser is correct.
We use policy gradient with the REINFORCE algorithm due to the large action space (2000+ possible words).

Results

Oracle	Train	Val	Test
1 GRU, 2 FC	78.94%	73.63%	73.62%
1 GRU, 2 FC+Drop	65.28%	64.55%	64.65%
1 GRU layer, 3 FC	79.95%	75.5%	75.23%
2 GRU layers, 3 FC	80.02%	75.78%	75.44%
Dominant Class	52.6%	53.8%	49.1%
Guesser	**Train**	**Val**	**Test**
1 GRU, 2 FC	64.21%	51.44%	51.58%
2 GRU, 2 FC	64.04%	56.69%	56.84%
2 GRU, 2 FC+Drop	58.31%	46.72%	50.26%
3 GRU, 2 FC	52.47%	48.03%	49.47%
1 LSTM, 2 FC	68.54%	56.43%	51.32%
2 LSTM, 2 FC	60.28%	51.71%	45.53%
Random	17.1%	17.1%	17.1%
Questioner	**Train**	**Val**	**Test**
1 LSTM, random sample			<Will Update>
2 LSTM, random sample			<Will Update>
2 LSTM, beam search			<Will Update>
2 LSTM w/ reinforcement, beam search			<Will Update>

Discussion and Future Work

The models performed well; they were able to get within 5% of state of the art accuracy for all tasks or better

➢In practice, we found that guesser performance increased when the image crop was not included as a feature; perhaps consider leaving out or using different features as input

➢Consider using different features from pretrained networks such as VGG or SqueezeNet

➢Consider using conversation history for the oracle