Applying Memory Networks to Question-Answering (QA)

Background and motivations

Most machine learning models lack an easy way to read and write to part of a (potentially very large) long-term memory component, and to combine this seamlessly with inference. Hence, they do not take advantage of one of the great assets of a modern day computer.

The central idea is to combine the successful learning strategies developed in the machine learning literature for inference with a memory component that can be read and written to. Recently there has been a resurgence in models of computation using explicit storage and a notion of attention [1, 2, 3, 4] The central idea is to combine the successful learning strategies developed in the machine learning literature for inference with a memory component that can be read and written to. In [1, 2, 3, 4], the storage is endowed with a continuous representation; reads from and writes to the storage, as well as other processing steps, are modeled by the actions of neural networks. During the last two years, both Google Deepmind and Facebook AI research are actively working on this area.

Question-Answering (QA)

In question-answering (QA) task, the system is told a set of facts or a story, and then it has to answer questions on that subject. The typical setting and examples of QA is illustrated in the following figure. To infer an answer of a question, the system need to access a set of statements (context). Only some subset of the statements contain information needed for the answer, and the others are essentially irrelevant distractors.

```
Sam walks into the kitchen.
Sam picks up an apple.
Sam walks into the bedroom.
Sam drops the apple.
Q: Where is the apple?
A. Bedroom
```

```
Brian is a lion.
Julius is a lion.
Julius is white.
Bernhard is green.
Q: What color is Brian?
A. White
```

```
Mary journeyed to the den.
Mary went back to the kitchen.
John journeyed to the bedroom.
Mary discarded the milk.
Q: Where was the milk before the den?
A. Hallway
```

In principle QA task can be modeled using a language modeler such as a recurrent neural network (RNN). However, RNNs are known to have difficulty in performing memorization. It recently shown that various kinds of proposed memory networks can be used to handle such problems that require long-term memory.

Your task

In this project, you are going to start with applying one of the proposed memory networks to the problem of QA task. The implementations for different versions

of memory network are open on Github. A good starting point is the methods proposed by [4].

The first part of the project will be replicating results in one of the paper we choose and understand various memory networks.

After you gain some experiences running memory network on the QA task, you need to apply it to one other domain where long-term dependency is of great concern. For example, one interesting application is moving from simple QA to more complex dialogues or conversations. Working with a new application requires you to make necessary network structure modification and this will also be the fun part.

Besides retaining information for longer term, memory network can also be use to quickly encode and retrieve new information so as to adapt to new trend in data. So we could also study some nice properties of memory networks while applying it to other domains.

Depending on the quality of the results we might attempt to publish it as a scientific publication.

Ideal candidate

- Experienced with Python and a good coder in general.
- Familiar with machine learning, especially artificial neural network.
- Passionate about the topic.

Contact

Fei Mi (fei.mi@epfl.ch)

References

[1] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. arXiv preprint arXiv:1410.5401, 2014.

[2] Ankit Kumar, Ozan Irsoy, Jonathan Su, James Bradbury, Robert English, Brian Pierce, Peter Ondruska, Ishaan Gulrajani, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. arXiv preprint arXiv:1506.07285, 2015.

[3] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. One-shot learning with memory-augmented neural networks. arXiv preprint arXiv:1605.06065, 2016.

[4] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In Advances in neural information processing systems, pages 2440–2448, 2015.