Learning to Reason

PhD Thesis Proposal

Wojciech Zaremba WOJ.ZAREMBA@GMAIL.COM

Advisors:
Rob Fergus FERGUS@CS.NYU.EDU
Yann LeCun YANN@CS.NYU.EDU
New York University

Abstract

Neural networks proven to be a very powerful models for object recognition (Krizhevsky et al., 2012), natural language processing (Mikolov, 2012), speech recognition (Graves et al., 2013), and many others (Sutskever et al., 2014). However, there is still a huge gap between them, and an intelligent systems. I identify several potential unaddressed skills, which intelligent systems should possess: (1) reasoning abilities, (2) capability to integrate with external interfaces and (3) small sample complexity. My research focuses on tackling this problems.

1. Introduction

It's clear that by improving performance of current statistical learning systems, we won't be able to make them intelligent. Even if our object recognition system would yield 0% of prediction error, they wouldn't be intelligent. Same applies to speech recognition systems, machine translation and others. This work asks what skills are necessary for statistical learning system to become "intelligent". Moreover, it attempts to address this remaining unaddressed skills.

I believe, that crucial, poorly addressed skills that intelligent system has to poses are (1) reasoning abilities, (2) capability to integrate with external interfaces and (3) small sample complexity. I would like to address all this problems within a seamless system. The same system should be used across different tasks, and should be able to emulate simpler models.

I have partially addressed some of proposed problems. I will make clear over the further part of this proposal, which parts have been addressed, and which are future goals.

2. Reasoning abilities

Reasoning - "the process of forming conclusions, judgments, or inferences from facts or premises"*.

System that can reason should be able to understand high level concepts like

- understand scope
- being able to memorize
- branching
- repetitions
- and many more

Each of this skills could be solved in particular domain. However, such approach limits use of such system, and assumes that we know a priori all possible scenarios. Moreover, I think that, intelligent reasoning system cannot be based only on predefined rules. Intelligent system has to be based on pattern matching, and application of learnt heuristic algorithms.

There are many domains where we can test if our system can reason, and if it's able to learn postulated concepts. Eventually, I would like to use the same tools for all domains. Domains of my interest are learning about computer programs, and proving mathematical theorems. This domains are rich in scoping, branching, pattern repetition, and so on. Drawing high level conclusions in such domains requires sophisticated reasoning skills.

*Definition from http://dictionary.reference.com/browse/reasoning

Input:
```
f=(8794 if 8887<9713 else (3*8334))
print((f+574))
```
Target: 9368.
Model prediction: 9368.

Figure 1. An exemplary program that we take as an input. Task is to predict program evaluation. We have achieved high performance on this task (Zaremba & Sutskever, 2014).

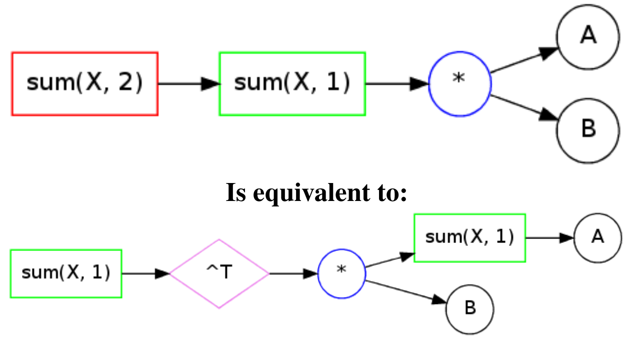

Figure 2. (Zaremba et al., 2014)

2.1. Reasoning in computer programs

The first reasoning task that I am interest in is to train statistical models meaning of computer programs. I propose one instance of program understanding. Task is to take a program as an input (in our case character-by-character), and predict program evaluation. Prediction of program evaluation requires understanding every single operand of a program. For instance, in case of addition it involves bit shifts, and memorization of operations on digits. Moreover, programs contain variable assignment, if-statements, and so on. We were able partially to address postulated problem (Zaremba & Sutskever, 2014). Figure 1 shows an exemplary program, target for such program, and prediction that we obtained with recurrent neural network.

Current results are promising, however they are very limited. We are able to deal with programs that can be evaluated by reading them once from left-to-right. Generic programs are not like that. Our reasoning system has to be able to evaluate for arbitrary long time, if task requires it. Moreover, it shouldn't be limited by finite memory size. Memory should be available as an interface 3.

2.2. Reasoning in mathematics

It's known that theorem proving is an intractable task in computational sense. However, humans are able to prove theorems. They have to employ a prior knowledge. They fit known mathematical "tricks" to the new problem.

More formally, we can think about proofs as a multiple axioms application that starts with hypothesis that we want to prove. This way mathematical skills can be perceived as learning prior over trees of axioms. This prior "suggests" us which sequence of axioms is more likely to lead to the theorem proof.

My interest lies in learning such prior in some constrained mathematical domain. Our recent work (Zaremba et al., 2014), concerns identities over polynomials over matrices. Figure 2 shows 2

3. Interface learning

Contemporary machine learning systems are closed in a box. They have limited access to the training dataset,

, and cannot interact with external interfaces.

Initial advances in machine learning, where lead by engineering features, e.g histogram of colors, SIFT (Scale-invariant feature transform) features. This approach has its limitations, and its fragile. Moreover, it requires a human expert to train the system for the every new domain. Ideal system should have access to external interfaces that might give access to information, or simplify it processing. Interfaces that I have in mind are following (1) an external memory (Weston et al., 2014; Graves et al., 2014), (2) grid that helps to process images, (2) search engine, (4) Linux syscalls, or (5) execution environment like python interpreter. External interfaces are not-differentiable, and their state space is massive. This obstacles could potentially be addressed by learning a differentiable model that describes them (e.g. neural network). Neural network would simulate such external interface for purpose of being differentiable in a model-based reinforcement learning approach. This work is under progress.

4. Small sample complexity

Current deep learning systems suffer of large sample complexity. Such high sample complexity hinders potential use of the systems in online learning systems (e.g. robots). Its expected that during the initial phase of learning any system without prior knowledge would need to consume a large number of samples. We hope that over the time of training, sample complexity should drop. However, this is not observed in current systems. I propose several approaches how sample complexity can be decreased. Meta-optimizer is a target solution. However, I propose some intermediate solutions, that can potentially help.

4.1. Augmentation marginalization

4.2. One-shot learning objective

This work is under progress.

4.3. Meta-optimizer

I would like to build a meta-optimizer that could overcome this limitation. Such optimizer would consume gradients of a neural network, and would decide on the next update step. Optimizer itself could be parameterized with a neural network. Proper weights could simulate any first order, gradient-based, learning algorithm like SGD, momentum, LBFGs etc. This implies that meta-optimizer subsumes all first order, gradient-based optimization techniques. Trained meta-optimizer could update the network in a much more clever way, and a single sample could provide enough knowledge. This work is under progress.

5. Discussion

Tackling aforementioned problems would take us much closer to the real intelligent systems, and defines for me the three main pillars of artificial intelligence. However, there are many other problems, which would need to be solved / integrated within such system to make it fully intelligent, e.g. navigation, learning by imitation, cooperation, and many others. I hope that, all other skills can be integrated by means of external interface, and don't have to be modelled in any special way. For instance, navigation skill could emerge as an use two interfaces (1) GPS location interface, and (2) an external memory.

6. Disclaimer

This is my personal opinion, and it shouldn't be judged in scientific way.

I strongly believe that creation of artificial intelligence is potentially dangerous. However, I think, that more dangerous is avoiding to create it. We exhaust resources of our planet in rapid fashion, and lack of resources leads to wars. The only way to have abundance of resources is to make them free. Artificial intelligence could make all resources virtually free. The only remaining question is if human can accept world where everything is given.

References

Graves, Alex, Mohamed, Abdel-rahman, and Hinton, Geoffrey. Speech recognition with deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 6645–6649. IEEE, 2013.

Graves, Alex, Wayne, Greg, and Danihelka, Ivo. Neural turing machines, 2014.

Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Mikolov, Tomáš. *Statistical language models based on neural networks*. PhD thesis, Ph. D. thesis, Brno University of Technology, 2012.

Sutskever, Ilya, Vinyals, Oriol, and Le, Quoc V. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*, 2014.

Weston, Jason, Chopra, Sumit, and Bordes, Antoine. Memory networks, 2014.

Zaremba, Wojciech and Sutskever, Ilya. Learning to execute, 2014.

Zaremba, Wojciech, Kurach, Karol, and Fergus, Rob. Learning to discover efficient mathematical identities. *arXiv preprint arXiv:1406.1584*, 2014.