Augmentation of Block-World

Eyvind Niklasson Cornell Tech
New York, New York
een7@cornell.edu

Abstract

This paper augments the work done by Misra et al. (2017) by modifying various parts of the architecture in order to try and determine any bottlenecks in performance. The modifications investigated are - various CNN architectures, Bi-LSTM instead of LSTM and attention based mechanisms on the input image. The results show improvement on the train set but little-to-no improvement on the test-set, demonstrating the augmentations' propensities to increase overfitting in this architecture.

1 Introduction

The work by Misra et al. (2017) on Bisk et al. (2016)'s Blockworld dataset presented a novel architecture achieving the first (and best) benchmark on performance on this task. The original dataset was released as a set of sentences describing the movements of blocks with various identifying marks (logos or digits) on a virtual board.

2 Background

Problem Definition: Block-World

The original environment by Bisk et al. (2016) consisted of individual cubes placed on a large square board. Each cube was labeled with an identifying bitmap graphic - a logo, number, etc. The blocks were then moved from their original (random) positions to form different patterns on the board (either another random placement, or placed to form a digit from MNIST). To achieve this final state, the blocks were moved either one by one or in sets of several blocks at a time - where "moved" is defined as "teleporting" from the initial location to the final location. At this point, mechanical turk workers were shown "before" and "after" movement images, and asked to describe, as if they were explaining to another human being, what changed between these states. The resulting dataset contains a number of boards undergoing transformations from an initial state to a final state (say an MNIST digit), with accompanying mechanical turk instructions along the way developed by asking mechanical turk workers to describe the changes. For an illustration of this mechanism please see Figure 1.

Figure 1: An example of the natural language instruction from Blockworld, along with the before and after states.

Motivation

The problem of having robots understand human-generated natural language is one that has been a research goal for decades. From the first automated telephone services to Siri, computer have slowly evolved to understand what humans want. However, lately, big advances have been shown with models and systems that can be trained end-to-end - that is both learn language and learn to interact with the world via one training process. Due to the strictly limited set of "well-supervised" or "well-annotated" data we have, solutions like this have great potential to improve the state of the art in human-robot interaction.

Reinforcement Learning Approach

Architecture

Our work here is entirely based off of the existing model by Misra et al. (2017). They produced the first reinforcement learning approach to the task,

with the goal of parsing the natural language instructions and having the correct actions be performed within the "block-world". The architecture is described in more detail in the paper but the overall design can be seen in Figure 2.

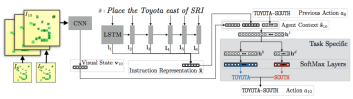

Figure 2: Architecture used by Misra et. al, pictureby Misra et. al

Previous Results

As the only paper to approach this "block-world"-modelling problem, the results by Dipendra et. al are also the current state of the art results. The best results were achieved with a reinforcement learning approach - making use of a policy network with two augmentations: contextual-bandit rewards - i.e. rewards are computed and applied at each time-step instead of a total reward, and in turn reward-shaping implemented via this contextual bandit setting. The reward-shaping is implemented via two reward-shaping functions - referred to as "F1" and "F2" in the paper - the first of which is a function of the distance to the target position, while the second attempts to encourage the agent to follow the shortest distance path (as algorithmically pre-computed).

The performance of the algorithm itself is reflected via the "Bisk Metric" - the average distance (in units of "block-width") from the center of target position at the final position of the block.

Known Issues/Areas of Improvement

A few empirically described areas of improvement were brought forth in the original paper, among them

- Blocks "failing" to stop correctly. When empirically observed, quite often a block being controlled by the network approaches the correct position, but does not output the "STOP" instruction. Instead, it proceeds to do a random walk around the target location, often ending up farther from the target than it initially finds itself when it first approaches the target.
- Blocks failing to navigate smaller spaces, such as a failure to squeeze between two blocks when this is required.

2.1 Style

3 Experiments/Approach

Bi-LSTM modification

Bi-LSTMs were first introduced by Huang et al. (2015) for Bi-LSTMs. The general principle of Bi-LSTMs is to run an LSTM in both directions on a given input sentence - with an individual hidden state and LSTM unit for each direction. The result is that at every time-step in the LSTM the output in a concatenation of the output from both the forward pass and the backward pass, resulting in information propagation from the entire sequence as opposed to only the previous time-steps.

In this specific architecture, the advantages of a Bi-LSTM are somewhat mitigated by the averaging of the LSTM outputs over time-steps, but nonetheless may provide an alternative way to have information propagate from the whole time sequence.

Convolutional Architecture Modifications

Convolutional Neural Networks (CNNs) have been used in computer vision since they were popularized by Lecunn et. al LeCun and Bengio (1995), and have proven wildly successful at such tasks. However, they are plagued by problems of their own (such as a limited window of attention depending on the size of a convolution, investigated by Luo et al. (2017)). This is often solved by having many layers of convolutions in a deep architecture - which is why the relatively shallow nature of the image-encoder in this architecture was interesting to experiment with.

Given that the decision unit in the architecture has to make a decision based on the exact state of the board (i.e. - where each piece on the board is in order to estimate a path around it), intuitively the CNN "encoder" must learn to express the position of all pieces (sometimes 17+ blocks) within the embedding of the board input, which is somewhat of a complex task - especially when it is to be learned in a reinforcement learning context.

The hypothesis we bring forth is that there is a bottleneck within the image processing pipeline which leads to uncertainty for the decision unit when placing a block. Such uncertainty would ex-

plain both the inability to properly "STOP" a block on the correct position, as well as a difficulty in movements and actions with small margins - such as squeezing between two blocks.

Thus, some of the areas of augmentation and experimentation attempted were:

- Adding more layers to the CNN
- Modifying CNN filters to be larger or smaller at the onset vs. the output, in order to encourage a wider or narrows "field of view".
- Increasing size of image embedding to allow more information encoded

Attention Based Approach

"Attention" has recently gained popularity as a method to improve performance of different architectures. The key idea behind attention is for a network to compute some form of vector which can decide which components of an input are "interesting" to pass on to the rest of the network. The principle was generalized in Figure 3 by Vaswani et al. (2017) with the concept of "keys", "queries" and "values".

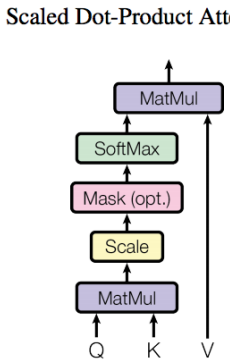

Figure 3: Visualization of keys, queries and values from Attention is All you Need.

Inspired by the work of Yang et al. (2016), attention may prove a good fit for mapping instructions and visual observations to actions, given the nature of the natural language instruction as a "query" and the visual image a natural candidate to be segmented into "key/value" pairs (for instance - one hypothetical outcome would be individual block types acting as keys, and their corresponding locations on the visual observation of the board as values). The work by Yang et al. (2016) showed promising results applying this natural-language based image-attention and it may be a promising avenue for this project as well.

Multi-Headed Attention

Multi-headed attention, first introduced by Vaswani et al. (2017), expands on the generalization of attention by also introducing multiple parallel executions of attention on the same data ("channels"). This approach has also shown promising empirical results.

4 Results & Discussion of Results

Bi-LSTM modification

The Bi-LSTM modification did not improve the test-time accuracy of the system, but did increase the train-time accuracy slightly - i.e. overfitting. As can be seen in Table 1, there was a slightly faster convergence during training, but no meaningful difference in test-results.

Experiment	Results		
	Train@2-epoch	Train@5-epoch	Test
BiLSTM	3.75	3.72	3.89
BiLSTM without averaging output	7.5	5.82	6.02
BiLSTM with 500-wide hidden state	3.72	3.69	3.79

Table 1: Results from Bi-LSTM experiments.

This is not entirely unexpected, as several metrics suggested the key information from the sentences was often already being extracted correctly. For instance, the model selected the correct block in $+95\%$ of the actions it output, and the "minimum distance" metric suggested the destination was often correctly identified by the network.

Convolutional Architecture Modifications

Experimental modifications and associated results to the CNN can be seen in figure Table 2.

Experiment	Results		
	Train@2-epoch	Train@5-epoch	Test
+1 Conv layer	3.99	3.77	3.82
+1 Conv layer + wider layers	3.56	3.42	4.10
+2 Conv layer	6.85	4.72	4.91
larger receptive field (larger convolutions)	3.85	3.74	3.85
larger image embed size	3.82	3.75	3.81

Table 2: Results from CNN experiments.

Increasing the width and depth of the convolutional network only slightly increased the real-world speed of convergence of the network, but introduced a very significant propensity to overfit. This suggests that the convolutional network potentially may have been a bottleneck in the original architecture but that a finer balance needs to be

found between increasing complexity and avoiding overfitting.

Attention Based Approach

Attention was implemented both as additive attention Vaswani et al. (2017) and multiplicative attention. Early results on training data were promising, as can be seen in Table 3, but overall there was little-to-no improvement on the test results. However, it was possible to qualitatively verify that the attention mechanism was performing as expected, at least to some degree, by investigating the "areas of interest" in the image. Please see Figure 4 for an example of the areas chosen by attention when placed towards the end of the second convolutional layer. The improvement in train accuracy suggests the addition of attention did increase the model complexity to some extent but that this complexity did not improve the overall generalizability of the model.

Experiment	Results			
	Train@2-epoch	Train@5-epoch	Test	Avg. Steps Test
Vanilla	3.82	3.72	3.92	37.7
Single-Head image attention 2nd conv layer	3.02	2.72	3.88	31.6
Single-Head image attention 3rd conv layer	3.11	2.97	3.93	33.5
Multi-Head image attention	3.14	2.89	4.01	32.3

Table 3: Results from Attention experiments.

Figure 4: Visualization of rough attention focus on relevant pieces about to moved. Boundaries should be treated as rough approximations due to convolutions occurring before this layer.

Another interesting metric as can be observed in Table 3 is that, just as in the improved CNN, the model exhibits more frequent use of the "STOP" instruction than in the vanilla case (a shorter average length of instruction set generally indicates that "STOP" was used in a greater number of test-cases).

Multi-Headed Attention

Multi-headed attention was attempted as per the implementation tested in Vaswani et al. (2017) and the work did not produce significantly different results from "single-head" attention. Results in Table 3.

5 Conclusion and Further Work

The overfitting behaviour exhibited by modifications to the image pipeline of the model both by the increased complexity convolutions and the attention mechanism is discouraging. However, the metric of increased "STOP" behaviour is promising as it hints that this may be linked to some ambiguity in the model regarding the visual observation processing part of the system. There may be a way to increase the complexity of this part of the model and implement some form of regularization to prevent the overfitting but capture the improvements. It may be worth trying a combination of the deeper convolutional layers with the attention mechanism, while implementing drop-out in the convolutional layers or other widely-used techniques.

There being relatively no difference in results with the modified text-pipeline suggests that indeed any bottleneck on accuracy and performance lies elsewhere (i.e. possibly the image pipeline as suggested above).

The experiments performed give a limited insight into which parts of the model are constrained and may help guide any future attempts at improving the state-of-the-art on this model.

6 Acknowledgements

I would like to greatly thank Dipendra Misra for bringing me up to speed on the project in time for me to both understand the existing model/work and complete the above experimentation in the few weeks that I had available. I would also like to fully extend my gratitude to Assistant Professor Yoav Artzi for teaching a very informative and useful class and for putting up with my frequently late submissions - while the project did not extend as far as I had hoped I strongly believe I learned much more getting my hands dirty as opposed simply reading and reviewing the same literature by and of itself. While many dozens of experiments were run in addition to the listed results, the biggest limiting factor was that of weak performance on the cloud based GPU available, resulting in a 8-10 hour turnaround times per experiment.

References

Yonatan Bisk, Deniz Yuret, and Daniel Marcu. 2016. Natural language communication with robots. *Proceedings of the 2016 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies https://doi.org/10.18653/v1/n16-1089.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR* abs/1508.01991. http://arxiv.org/abs/1508.01991.

Y. LeCun and Y. Bengio. 1995. Convolutional networks for images, speech, and time-series. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*. MIT Press.

Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard S. Zemel. 2017. Understanding the effective receptive field in deep convolutional neural networks. *CoRR* abs/1701.04128. http://arxiv.org/abs/1701.04128.

Dipendra Kumar Misra, John Langford, and Yoav Artzi. 2017. Mapping instructions and visual observations to actions with reinforcement learning. *CoRR* abs/1704.08795. http://arxiv.org/abs/1704.08795.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR* abs/1706.03762. http://arxiv.org/abs/1706.03762.

Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. 2016. Stacked attention networks for image question answering. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* https://doi.org/10.1109/cvpr.2016.10.