CSE 571 - Robotics
Guided Project 1 - Cartpole

1 Description

This is the first **out of 2** homework-guided projects, which will span 4 weeks each. These projects are designed to be similar to the normal course projects in structure, but more restricted in scope and environment so students can explore more course topics in depth. For this project, you must select an idea that involves deep learning (in PyTorch) and the cartpole simulator given to you in Homework 1. You can borrow code from Homework 1.

The projects will be completed in teams of two to three. In addition to restricting scope and environment, teams MUST use PyTorch (as opposed to other deep learning frameworks) in order to implement their networks.

2 Timeline

- Proposal: due on Friday 4/17/20.
- Mid-Progress Report: due on Tuesday 4/28/20.
- Final Report: due on Friday 5/8/20.

3 Deliverables

The proposal and mid-progress report will be hosted on a Google Doc (You will use this document for **BOTH** this guided project and the next guided project). For the final report, you will submit a PDF so that you can use Latex.

Proposal [1-page] The proposal should include:

- The team members working on the project, and each member's experience with deep learning and/or deep learning frameworks (e.g. PyTorch/TensorFlow).

- A paragraph summarizing the proposed project. Remember, this project needs to be a reasonable scope to complete within 3 weeks. The instructor/TAs will provide feedback to help guide you.
- Milestones for mid-progress and final reports.

Mid-Progress Report [1-page] The mid-progress report should report on successes and unforeseen problems. If the timeline/project outcome needs to be updated, please make those changes in the web-based blog, and note this in the mid-progress report. We highly encourage visualizations, e.g. images and/or short videos.

Final Report [2 pages] The final report will summarize the guided project, not including references. In addition to the final report, teams must submit a short 3-minute presentation video describing the project/results. For example, you can prepare some slides and speak to them using screen recording. Finally, teams must submit executable code (with a README) so the TAs can replicate the results. More details will come later.

Example Project Ideas

- Learn the cartpole dynamics model with uncertainty. Standard neural networks are not designed to produce uncertainty estimates; however, recent research has shown how deep networks can be adapted to produce uncertainty estimates in the predictions. This includes training an ensemble of networks [1], or using dropout [2]. Perhaps both can be compared.
- Learn a CNN (or RNN) to predict the underlying cartpole state given past image(s). E.g. is 1 image enough to predict physical state and velocities? What about including multiple images in the input, or providing the velocity as input?
- Predict future images given past images with a CNN/RNN. Show the effect of a CNN architecture vs. RNN architecture. Show the effect of different parameter choices, e.g. number of past images.
- Use previous images to predict action, mimicking the preset cartpole swingup policy (given in HW1). This is essentially behavior cloning. Demonstrate the method's performance for cartpole control.
- Implement a deep particle filtering network [3] for cart pole state estimation, possibly using the uncertainty estimates generated via the first item above.

The above ideas are our suggestions and you are free to combine these and propose your own ideas. We strongly encourage to include some aspects of image data in your project since that's the key strength of deep networks.

Resources

- If you don't have access to a machine with a GPU, you can use Google Colab (access to a free GPU for up to 12 hours per session). You only need a Google account. Colab operates as notebooks, very similarly to

Jupyter Notebooks if you have used them. PyTorch/TensorFlow is pre-installed in the Colab notebooks, so no need for manual installation. You can find more information here: `https://colab.research.google.com/notebooks/intro.ipynb`

- PyTorch: A framework that allows you to do Numpy-esque computations on GPU. It also has a lot of support for autodifferentation and neural networks, making it one of the most popular deep learning frameworks. You are required to use this framework for the guided projects. See tutorials here: `https://pytorch.org/tutorials/`

References

[1] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pages 6402–6413, 2017.

[2] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

[3] Rico Jonschkowski, Divyam Rastogi, and Oliver Brock. Differentiable particle filters: End-to-end learning with algorithmic priors. In *Proceedings of Robotics: Science and Systems*, 2018.