Prakhar Sharma

prakharsharm@umass.edu · +1-413-362-4721 · github.com/sharmaprakhar

Education

University of Massachusetts — Amherst, MA
Masters in Computer Science — Expected Graduation: May 2019

Birla Institute of Technology and Science — Pilani, India
B.E. Electronics and Instrumentation — Graduation: May 2013
MSc. Economics

Experience

SRI International: *Research Intern* — Menlo Park
Advisor: Melinda Gervasio (Associate Technical Director) — Sept 2018 — Dec 2018

- Trained a custom python based domain using RL algorithms written in object oriented python
- Worked on OpenAI baselines (object oriented python and Tensorflow)
- Trained various advanced reinforcement learning algorithms using pytorch and object oriented python

Robert Bosch : *ML Software Intern* — Pittsburgh
Advisor: Joseph Szurley (Research Scientist) — May 2018 — August 2018

- Developed a python library for Keras model manipulation
- Worked on cosine similarity regularization for CNNs
- Deployed ARM-CMSIS NN library on Cortex M4 using Mbed
- Designed pruning methods for CNNs for audio classification

National University of Singapore : *Research Associate* — Singapore
Advisor: Prof. Massimo Alioto (IEEE Fellow) — Jan 2016 — August 2017

- Designed resource efficient accelerators for deep learning
- Designed a plug and play DRAM framework on Xilinx Zynq-7000 FPGAs

AMD : *Design Engineer* — Bangalore, India
Mentor: Prashanth Vallur (Senior Manager) — August 2013 — December 2015

- CPU-GPU clock design; CAD tool scripting; Unix

Projects and Publications

University Of Massachusetts - Amherst

Deep Learning
Advisor: Prof. Erik Learned-Miller

- Pruning CNNs based on cosine similarity **presented at NIPS workshop 2017** *(pytorch)*
- Using images to generate music using google-magenta RNN *(jupyter, python)*

AI, Robotics and RL
Advisor: Philip Thomas, Shlomo Zilberstein

- Inplemented reinforcement learning (SARSA, Q) for meta level control
- Implemented NAC-LSTD (Peters and Schaal)

ML and Systems

- Implemented ridge, lasso, KNN, SVM and bayesian frameworks using scikit learn *(python)*
- Memory allocator in RUST and distributed chat client in Golang *(RUST, Golang)*

Courses

Semester 1: Deep Learning, Reinforcement Learning, Machine Learning
Semester 2: Robotics, Artificial Intelligence, Systems

Computer Proficiency

Language:	Python, Go, C++, MATLAB, Latex
Scientific computing:	PyTorch, TF, Keras, Numpy, joblib
Hardware:	Vivado, Verilog, ARM-Mbed

Project Descriptions

SRI International: Explainanble Autonomy *(Python, Tensorflow, Reinforcement learning, OOP)*
As an intern with Dr. Melinda Gervasio, I applied RL algorithms like options critic (OC) and proximal policy optimization (PPO) to custom environments and reason about the policy being learnt. We are working to build an environment that has two intermediate choices (subgoals) and a stochastic goal state. Given such a domain, I investigated whether HRL has any clear benefits such as convergence speed, robustness to hyperparameter perturbation etc., as compared to a flatter approach. Empirically, I found correlation between subgoals and hierarchical options, and hence a rudimentary explanation for the agents actions.

Bosch Research: Classification of raw sounds *(Keras, Pytorch, scikit-learn, C)*
As a software intern at Bosch, Pittsburgh in summer 2018, my goals were twofold: to train a CNN to classify environmental sounds (ESC-50) using raw sound waves, as opposed to using log-Mel features; then reduce the trained network to fit on an embedded device like ARM Cortex-M4. Our trained network achieved mean precision 0.66 and F1 score 0.53 over 50 classes. While this performance wasnt better than the state of the art, it is (to the best of our knowledge) one of the best models trained using raw waveforms. I created an open source network surgery library in python for the keras-tensorflow framework that enables manipulation of kernels in a CNN trained with Keras. This enabled easy pruning and quantization of trained network parameters for transfer learning

UMass: Metareasoning using Reinfrocement learning *(Python, OOP, RL)*
I am working on a project jointly advised by Prof. Thomas and Prof. Zilberstein at UMass. Anytime algorithms can generate increasingly better solutions to NP hard problems given enough time; but the cost of time added to the utility of the solution gives rise to a convex solution profile. The project aims to apply RL algorithms to choose an optimal stopping point for anytime algorithms. I implemented Q-learning and SARSA to test the hypothesis and preliminary results are encouraging. This technique has the potential to reduce offline computation in mobile robot maneuvering