Xuankang Lin

(765)398-2290
lin420@purdue.edu
http://xuankanglin.com/About-Me/

Research Interests

My current research interest is about combining advanced Machine Learning with Programming Languages techniques. I believe the PL approach can play an important role in enhancing ML, making it safer and more interpretable.

Education and Work Experience

- **Purdue University** West Lafayette, IN
  Ph.D. in Computer Science *Expected: 05/2020*
  - Under the supervision of Prof. Suresh Jagannathan and Dr. Roopsha Samanta.
  - Passed Qualifying Examination in 11/2015.

- **Intel** Shanghai, China
  Intern at SSG group *12/2012 – 06/2013*
  - Optimized Cocos2d-html5's display on devices with Retina display.
  - Improved Cocos2d-html5's audio engine using Web Audio API (code merged into official repository).

- **Tongji University** Shanghai, China
  B.E. in Software Engineering *06/2013*
  - Outstanding Graduate of Shanghai in 2013.
  - Exchange student at Rose-Hulman Institute of Technology (Terre Haute, IN) in fall 2012.

Skills

Proficient: Python, C/C++, Scala, Java, Git

Familiar: PyTorch, Coq, OCaml, LaTeX

Selected Projects **Github @AndriyLin**

- **Repairing Neural Networks towards Safety Properties (Ongoing)** *10/2017 – Now*
  - Starting from mid 2017, there are works verifying neural networks against certain safety properties. Now that it can be verified, it should not be intractable to repair the network regarding the same property.
  - We encode the Repairing-NN problem as a Synthesis/Repair problem, using the roughly trained network as the sketch, and call SAT/SMT solver to find a satisfying weight update that is also minimal in differences.

- **Neural Embedding based Anomaly Detection** *08/2017 – 10/2017*
  - Experimented on anomaly detection using neural embeddings of each item.
  - The embeddings are trained via a Siamese model from randomly sampled item pairs.

- **Learning Latent Memory Models from Litmus Tests** *10/2016 – 04/2017*
  - We present a new approach to learn probabilistic memory models from litmus tests.
  - The herd7 tool (in OCaml) is adapted to generate executions from litmus tests on the weakest memory model where most relaxations are allowed. The assertions attached with each litmus test are used as labels for these concrete executions.
  - Conditional Random Field and Decision Tree techniques are implemented (in Scala) to learn a probabilistic memory model specification from these labeled executions.
  - More details can be found in the technical report at https://goo.gl/prmE1v.
  - Code is available at https://goo.gl/1LFvXG.