Maximilian C. Lam

agnusmaximus@gmail.com

Education

Stanford University *2017 - Current*
M.S. in Computer Science - Artificial Intelligence

University of California, Berkeley *2013 - 2017*
B.A. in Computer Science
Graduated with High Distinction (*magna cum laude*)
GPA: 3.93 / 4.0

Publications

Jeffrey Regier, Kiran Pamnany, Keno Fischer, Andreas Noack, **Maximilian Lam**, Jarrett Revels, Steve Howard, Ryan Giordano, David Schlegel, Jon McAuliffe, Rollin Thomas and Prabhat. *Cataloging the Visible Universe through Bayesian Inference at Petascale*. International Parallel and Distributed Processing Symposium (IPDPS), 2018.

Dong Yin, Ashwin Pananjady, **Max Lam**, Dimitris Papailiopoulos, Kannan Ramchandran, Peter Bartlett. *Gradient Diversity Empowers Distributed Learning: Convergence and Stability of Mini-batch SGD*. Artificial Intelligence and Statistics (AISTATS), 2018.

Kangwook Lee, **Maximilian Lam**, Ramtin Pedarsani, Dimitris Papailiopoulos, and Kannan Ramchandran. *Speeding up Distributed Machine Learning Using Codes*. IEEE Transactions on Information Theory (IEEE Trans. Inf. Theory), 2017.

Xinghao Pan, **Maximilian Lam**, Stephen Tu, Dimitris Papailiopoulos, Ce Zhang, Michael I. Jordan, Kannan Ramchandran, Chris Re, and Benjamin Recht. *Cyclades: Conflict-free Asynchronous Machine Learning*. Advances in Neural Information Processing Systems (NIPS), 2016.

Work Experience

Google – Platforms Team, Software Engineering Intern *May - Aug 2017*
Project: TensorFlow Virtual Performance Estimator
- Modeled TensorFlow performance of neural machine translation training on: proprietary (Google) topology, HPC multi-node cluster of *Intel-Xeon NVIDIA NVLINK'ed Tesla-class Volta-100/Pascal-100/Kepler-80/Kepler-40 GPU* Nodes
- Enhanced early stage implementation of TensorFlow's performance estimator into a working application
  - Enabled performance modeling of running TensorFlow graphs on generalized HPC hardware models
- Developed new queuing theory algorithm to estimate performance of distributed machine learning
- Successfully predicted performance of neural machine translation and convolutional neural networks (98% correlation between estimated times and actual times)
- Presented TensorFlow virtual performance estimator in Google tech talk to over 100 people

Google – Search Team, Software Engineering Intern *May - Aug 2016*
Project: Music Knowledge Graph / Knowledge Panels
- Enhanced music knowledge panels with music genres from the Google Metajam database
- Calculated music genre coverage statistics displayed by the Google search engine
- Tuned knowledge graph pipeline load balancer and improved performance of knowledge graph triples generation by 100%
- Launched entire Google search stack on private cluster to evaluate quality of new music genres query results

Linkedin – Tools Team, Software Engineering Intern *May - Aug 2015*
Project: Codesearch
- Designed and integrated new regular expression search feature to company-wide codesearch tool
- Improved performance of regex searches by 2x-5x by using trigram index algorithms
- Deployed internal codesearch engine and regex feature company wide

Evernote – Software Engineering Intern *May - Aug 2014*
Project: Penultimate 6 iPad Writing App
- Researched and implemented new architectural patterns for iPad handwriting

- Implemented core test-cases of Penultimate 6 into Xcode and Jenkins integration server
- Built major portions of Penultimate 6 including: login, settings, search

Teaching

Teaching Assistant for Stanford CS107 (Computer Organization & Systems) *Autumn 2017*
With Professors Julie Zelenski and Chris Gregg
The third course in computer science curriculum for undergraduates. Explores introductory topics in computer systems and low level programming with C programming, x86 assembly and gdb debugging.

Projects

Cataloging the Visible Universe through Bayesian Inference at Petascale
Worked with founders of the Julia programming language. Scaled Julia code to do machine learning on terabytes of astronomical data on NERSC supercomputers, reaching petaflops of compute. Enhanced parallelism by developing the algorithm for conflict free multi-core parallel optimization used in paper.

Gradient Diversity
Researched the relationship between batch size, compute performance and theoretical convergence properties in distributed machine learning. Performed experiments showing that dropout reduces the similarity of aggregrated gradients, allowing increased batch size and more efficient distributed machine learning training. Analyzed and aggregated results for presentation in paper.

Cyclades
Developed multi-core conflict free parallel algorithms for machine learning optimization. Investigated the benefits of NUMA aware caching in speeding up machine learning performance. Analyzed and aggregated results for presentation in paper.

Matrix Multiply with Codes
Investigated the use of information theory and correction codes to reduce the impact of stragglers in distributed machine learning. Applied these concepts to speed up distributed machine learning training for paper.

Cache Efficient SGD
Investigated the impact of cache friendly ordering of data to speed up machine learning optimization.

Skills

Programming Languages
C, C++, Objective-C, Java, Python

Other
TensorFlow, PyTorch, OpenMp, MPI, AWS, Deep Learning, Big Data, Git