Statement of Purpose

of Ligeng Zhu (Ph.D. applicant for Fall 2020)

My research topics mainly lie in the general areas of machine learning and scalable systems. Scalable systems enable advanced learning techniques and powerful learning methods to help design better systems. It is exciting to see what we can achieve when bringing advanced techniques and systems together. There are several areas that I am particularly interested in. The details are listed below:

Scalable Distributed Training on Decentralized Data / Federated Learning

Many privacy-sensitive data (e.g., patient's medical history, personal album, keyboard input history) cannot be centralized to a data center. Federated learning [1] provides a solution where users collaboratively train a shared prediction model while keeping all the training data on the local device. However, the network latency is high under such cases and leads to poor scalability. To solve this, we propose *delayed update* to tolerate latency by putting off synchronization and *temporally sparse update* to amortize latency by reducing communication frequency. We *theoretically* justify that our algorithms have no slower convergence than SGD and *empirically* exhibit that our algorithms can train ImageNet models across the world without loss of accuracy and scalability. This work demonstrates that **Synchronous SGD can perform no slower than Asynchronous SGD under limited networking** and is going to appear at MLSys workshop at NeurIPS in 2019 [2].

We also rethink the safety of the fundamental gradient exchange scheme in modern multi-node learning systems. For a long time, people believe the numerical gradients are safe to share and will not expose the semantic training data. However, we show that **it is possible to obtain the private training data from publicly shared gradients and the leakage is pixel-wise accurate for images and token-wise matching for texts**. We name this technique as *Deep Leakage from Gradients* and such deep leakage cannot be prevented unless noise that hurts accuracy is applied. This work is accepted by NeurIPS 2019 [3] and we hope it can raise people's awareness to rethink the gradients' safety.

Neural Architecture Search / AutoML / Design Automation

Deep neural networks have become a central component of many machine learning systems. However, it still takes a lot of engineering efforts to design good and efficient models. We want to adopt machine learning to automate the process, says use machine learning to design machine learning models. NAS [4] reveals that this is possible but the vast computational resources it required are even more expensive than engineers. To address, we studied the connection between neural architecture search and model compression and considered the problem as a path-level pruning. Such a reformulation allows us to reduce the cost to the same level as regular training. This opens future avenues for direct search on large-scale datasets (e.g., ImageNet) and target hardware (i.e., CPU/GPU/Mobile/ASIC).

We name this method as *ProxylessNAS* and demonstrate the effectiveness of the directness and hardware-specialization: (i) To achieve ResNet-50 level accuracy, our proxylessNAS (mobile) runs **1.83× as fast as Google's MobilenetV2** on mobile devices. (ii) Our proxylessNAS (GPU) **outperforms MobilenetV2 by 3.1% on Top-1 while running 20% faster**. Besides, we found that different hardware platforms have different preferences for efficient neural network architectures. This suggests the widely used paradigm "one CNN for all platforms" does not fully utilize the hardware efficiently. We should specialize a CNN for each platform instead of generalizing. This work is accepted by ICLR 2019 [5] and powers our team to win **the first place** in Visual Wake-up Word Challenge@CVPR'19 and

the third place in the classification track of in Low-Power Image Recognition Challenge@ICCV'19. Till now, ProxylessNAS has 146 citations and the journal version is going to appear at IEEE Micro [6].

Accelerate Deep Learning Computations via Co-design

ProxylessNAS is a *hardware-aware design*: Search the best network architectures for a given (fixed) framework and hardware. But the ultimate solution should be *co-design*: Search best configurations for both software and hardware. Current existing works like Halide [7], TVM [8] and TensorRT [9] focus on learning to optimize single operators without considering high level neural network architectures. But such a design pattern does not provide sufficient parallelization opportunities on modern hardware, especially when many modern deep neural networks are becoming more compact and consist of many small operators. It explains why CNN like NASNet [4] has fewer FLOPs but actually runs slower.

Besides intra-operator parallelism, it is necessary to explore inter-operator parallelism and such parallelism depends on both CNN architecture and hardware specs. Built upon recent MetaFlow [10], which fuse operators by graph transformation to improve inter parallelism, we provide a more general solution to schedule concurrent executions automatically. **This technique is a platform-agonist and can serve as a plugin for existing frameworks. Our algorithm demonstrates 1.8 to 2.9× speedup on modern CNN benchmarks** by customizing scheduling for different hardware and settings. The improvements are even better on NAS specialized architectures. This work is submitted to DAC 2020 and will be open-sourced.

MIT is an ideal place for me to continue my quest because of its influential research groups, inspiring and exciting projects, and collaborative research atmosphere. Although I am quite open to a variety of research topics, there are several professors whose projects are particularly appealing to me: Professor. **Song Han** has many pioneering works on design automation and efficient deep learning; Professor. **Tim Kraska**'s inspiring projects at the intersection of systems and machine learning fascinates me a lot; I am also interested in Professor **Michael Carbin**'s lottery hypothesis and its applications.

[1] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

[2] Ligeng Zhu, Lu Yao, Yujun Lin, and Song Han. Distributed training across the world. In *Neural Information Processing Systems, Machine Learning for System Workshop*, 2019.

[3] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *Neural Information Processing Systems*, 2019.

[4] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

[5] Han Cai, Ligeng Zhu, and Song Han. ProxylessNAS: Direct neural architecture search on target task and hardware. In *International Conference on Learning Representations*, 2019.

[6] Song Han, Han Cai, Ligeng Zhu, Ji Lin, Kuan Wang, Zhijian Liu, and Yujun Lin. Design automation for efficient deep learning computing. *IEEE International Symposium on Microarchitecture*, 2019.

[7] Jonathan Ragan-Kelley, Connelly Barnes, Andrew Adams, Sylvain Paris, Frédo Durand, and Saman Amarasinghe. Halide: a language and compiler for optimizing parallelism, locality, and recomputation in image processing pipelines. In *Acm Sigplan Notices*, volume 48, pages 519–530. ACM, 2013.

[8] Tianqi Chen, Lianmin Zheng, Eddie Yan, Ziheng Jiang, Thierry Moreau, Luis Ceze, Carlos Guestrin, and Arvind Krishnamurthy. Learning to optimize tensor programs. In *Advances in Neural Information Processing Systems*, 2018.

[9] Nvidia. Nvidia tensorrt: Programmable inference accelerator.

[10] Zhihao Jia, James Thomas, Todd Warszawski, Mingyu Gao, Matei Zaharia, and Alex Aiken. Optimizing dnn computation with relaxed graph substitutions. 2019.