Summary Discussion

*Friday AM*
Need for more computing capability

Need for more computing is driven by data and models:

- Growth in rates: Larger / denser / faster data collection devices
- Growth in number: Ubiquity of sensors (cameras, thumb-sequencers,..)
- Better models: ML training with more compute on more data
- More complex models: simulations with more detail, e.g., traffic
Categories of computing

- Processing data (often at the edge)
  - Error correction, filtering, feature detection, compression, encryption
  - Pattern: Stream through data and do fairly localized computation

- Understanding data: building models
  - Solving “inverse” problems broadly: What model explains data?
  - Solving inverse problems: Iterate over possible models to find the best
  - Pattern: Iterative algorithm using all (or selected subsets) of data
  - NP-hard problems in understanding, e.g., Bayesian models (non-DL)

- Prediction: evaluating models
  - ML inference
  - Scientific simulation
  - Pattern: Depends?

Are we in the Linpack days of Machine Learning?
What research is needed?

• Many good ideas to synthesize from yesterday
What are the crosscutting ideas?

• Generalized specialized? (Adrian, Sarita)
  • Vs. 10x10 (many fixed function accelerators)
  • What is next after GPUs?
  • Programmable vs.? Reconfigurable (not necessarily FPGAs)
• Algorithm-driven architecture (Josep, Mattan)
  • Algorithms (and variations) not being studied (and their architectures)
  • Extreme sparsity and graph algorithms (range of sparsity / structure)
  • Memory-intensive specialized?
  • How to communicate between different computations?
  • Sparsity
• Whole workflow constraints (Vivek, Sasa)
  • Different specialization for power / energy / size on edge vs data center
  • Moving between different models of learning (GPUs -> NN)
  • Productivity stuff
What research is needed?

- What architectures?
- What programming systems? Power issues?
- What should academia do?
- Understanding precision
- What infrastructure would researchers need to do this?
- What is the right funding/organizational model?
- Very high level programming: what’s missing for experts
  - Getting from demo to “actual” implementation
  - End-to-end productive
- Very low power machine learning
  - Only need 1 bit for inference (?)
- How to get to chip building?
- Layers of abstraction
- Are there better ways of piecing things together