Neural Networks as Function Primitives
Software/Hardware Support with X-FILES/DANA

Schuyler Eldridge\textsuperscript{1}  Tommy Unger\textsuperscript{2}  Marcia Sahaya Louis\textsuperscript{1}
Amos Waterland\textsuperscript{3}  Margo Seltzer\textsuperscript{3}
Jonathan Appavoo\textsuperscript{2}  Ajay Joshi\textsuperscript{1}

\textsuperscript{1}Boston University Department of Electrical and Computer Engineering

\textsuperscript{2}Boston University Department of Computer Science

\textsuperscript{3}Harvard University School of Engineering and Applied Sciences

Boston Area Architecture Workshop ’16
Neural Networks as Function Primitives

Motivation

- Neural networks and machine learning are everywhere (again)
  - Broad use in high tech and big data, e.g., Google’s Tensorflow [1]
  - Enable automatic parallelization, e.g., ASC [3]
  - Provide a means for approximate computing, e.g., NPU [2]

- Our vision
  - Neural networks are a new functional primitive useful at various scales of computation [4]

- With that in mind, we’ve developed software and hardware for the use of accelerator-backed neural network computation

Our Contributions Towards this Vision

**X-FILES: Software/Hardware Extensions**

**Extensions for the Integration of Machine Learning in Everyday Systems**
- A defined user and supervisor interface for neural networks
- This includes supervisor architectural state (hardware)

**DANA: An Example Multi-Transaction Accelerator**

**Dynamically Allocated Neural Network Accelerator**
- An accelerator aligning with our multi transaction vision

**Neural Network Transactions**

*A transaction encapsulates a request by a process to compute the output of a specific neural network for a provided input*
What does that mean?

1. Grab a Rocket Chip RISC-V Microprocessor [1]
2. Build a RISC-V toolchain
3. Grab a copy of our X-FILES/DANA accelerator [2]
   - Implemented in Chisel [3]
4. Build an FPGA configuration for Rocket + X-FILES/DANA
5. User processes can safely throw transactions at X-FILES hardware
   - With support for feedforward and learning computation

[1] Rocket Chip git repository, UC Berkeley, Online: github.com/ucb-bar/rocket-chip
[2] X-FILES/DANA git repository, Boston University, Online (soon!): github.com/bu-icsg/xfiles-dana
**X-FILES Software Components**

**Supervisor API**
- Establishes sets of processes that can access neural network hardware
- Defines the neural networks that processes are allowed to access
- *More details on the poster!*

**User API**
- Works at the level of *transactions*
  - A complete request for access to neural network resources, communication of inputs, processing, and communication of outputs
- Initiating a new transaction
- Writing data
- Reading data
X-FILES/DANA Hardware

Components

- X-FILES Hardware Arbiter maintaining transaction state
- DANA to move transactions towards completion
  - With support for feedforward or learning computation

Figure: X-FILES/DANA hardware architecture
Open Source Plans

Remaining Items
- Linux kernel integration
- Support for asynchronous data transfer

Open Source Availability
- Should be ready by the end of February
- On GitHub: github.com/bu-icsg/xfiles-dana
Acknowledgments

This work was supported by the following:

- A NASA Space Technology Research Fellowship
- An NSF Graduate Research Fellowship
- NSF CAREER awards
- A Google Faculty Research Award