Training Convolutional Neural Networks on FPGAs

Kyle Daruwalla (daruwalla@wisc.edu) and Akhil Sundararajan (asundararaja@wisc.edu)

Introduction and Motivation

- CNNs are popular for image classification, speech recognition, and object recognition
- CNN training phase is limited by general purpose hardware
  - Memory access latency
  - Complex arithmetic units
- Field-programmable gate arrays (FPGAs) are a platform for designing specialized hardware
- Apply traditional hardware design techniques to the CNN training algorithm

Goals and Objectives

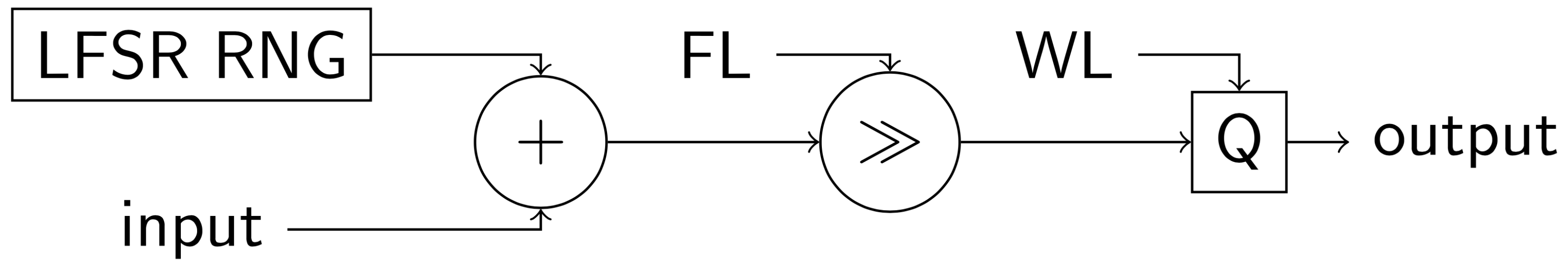
- Train a CNN using TensorFlow on various Amazon EC2 Spot Instances
- Design and validate limited precision CNN training hardware for FPGAs
- Simulate limited precision CNN algorithm in MATLAB
- Measure and compare accuracy and training time for equivalent number of gradient computations

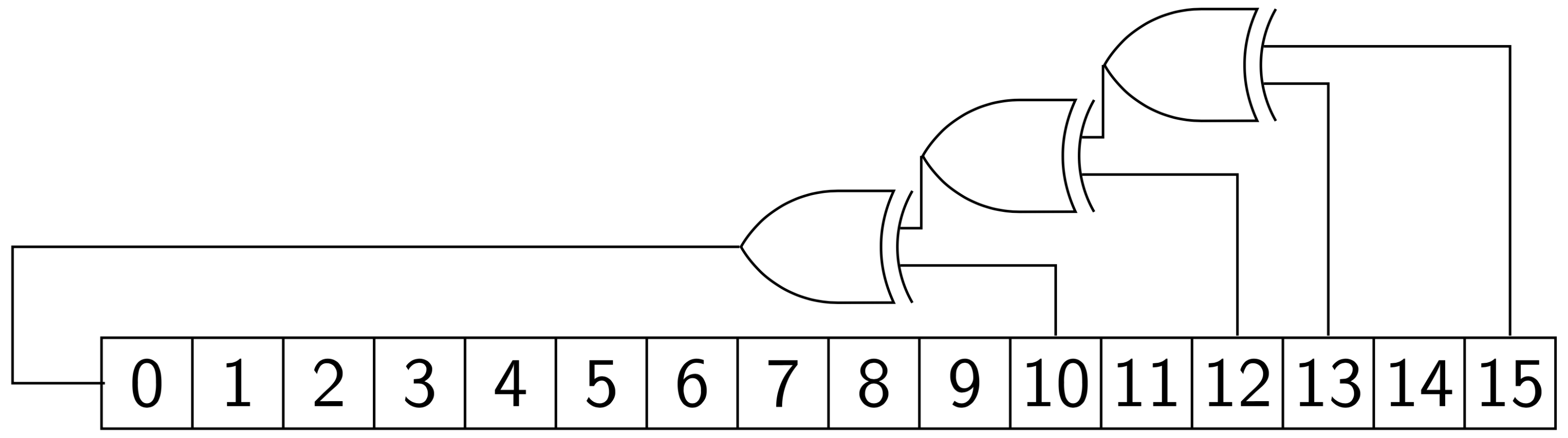
Design Specifications

- CNN Structure: CONV -> POOL -> FC
- Software baseline implemented in TensorFlow
  - Run on Amazon EC2 c2x.large
  - Run on Amazon EC2 c8x.large
- MATLAB simulation custom built for limited precision
- FPGA design targeting Xilinx Artix-7 Series

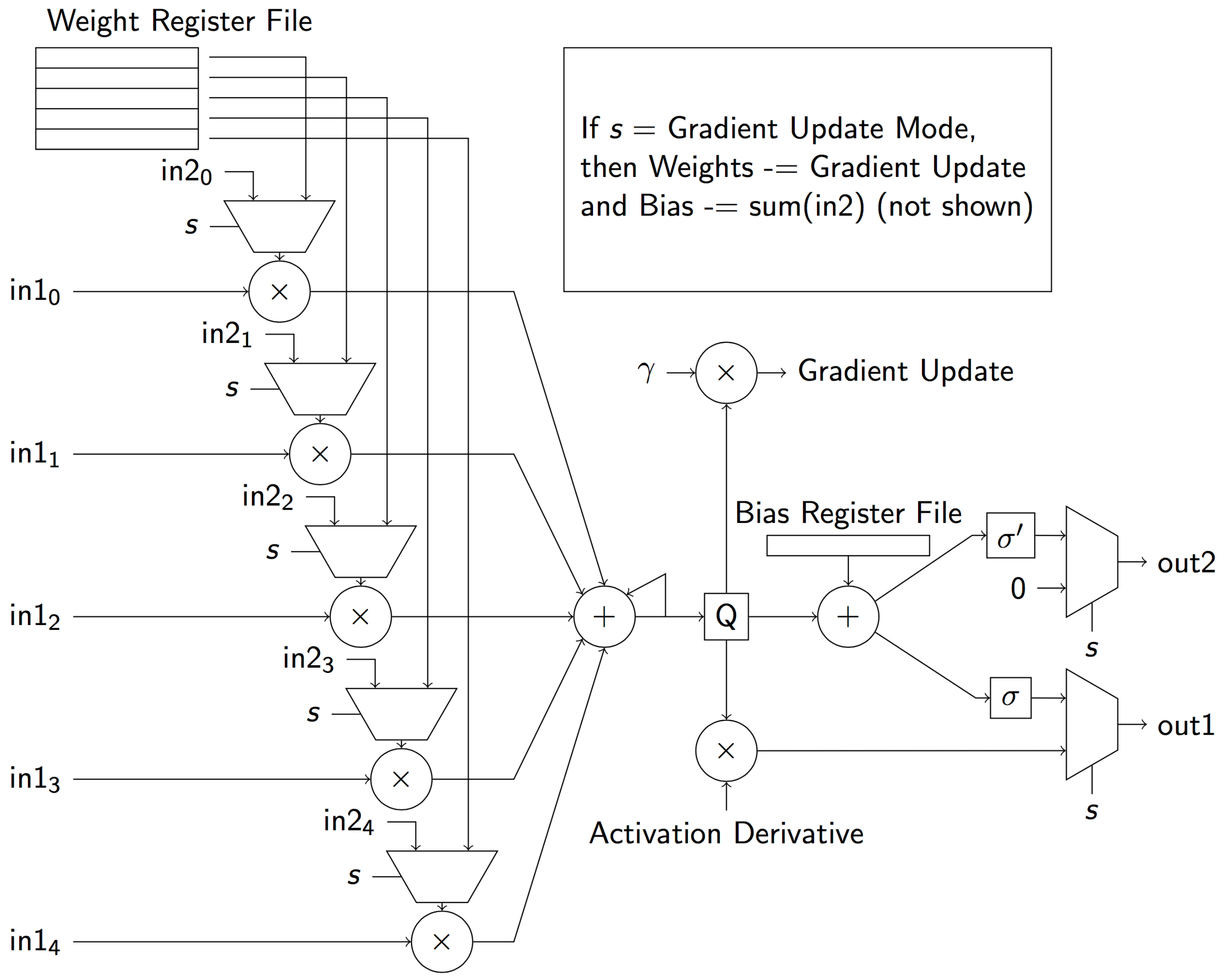
Hardware Design

- All numbers are represented in 16-bit fixed point
  - 1 sign bit, 1 integer bit, 14 fractional bits
- Stochastic rounding is used for quantization

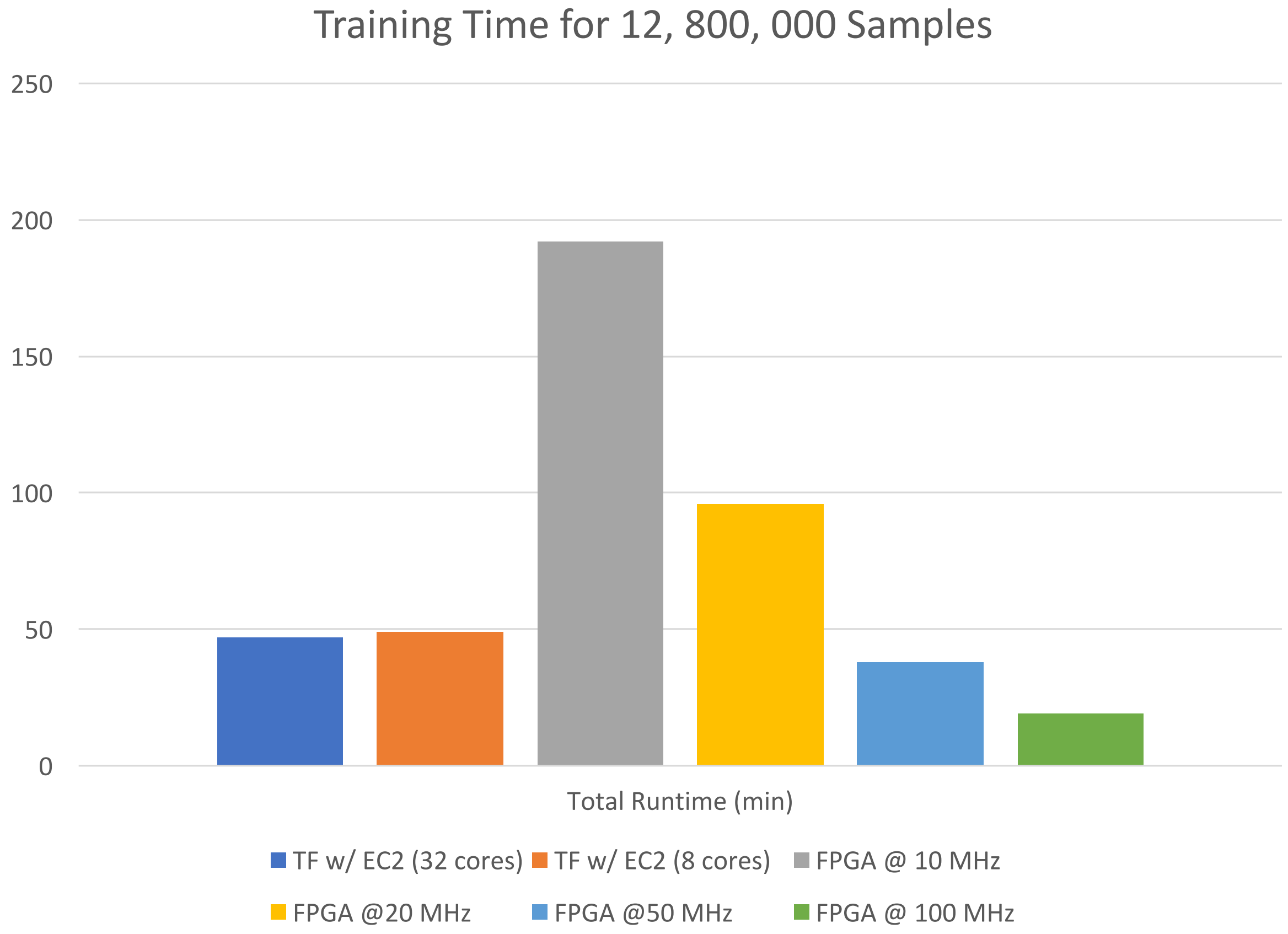




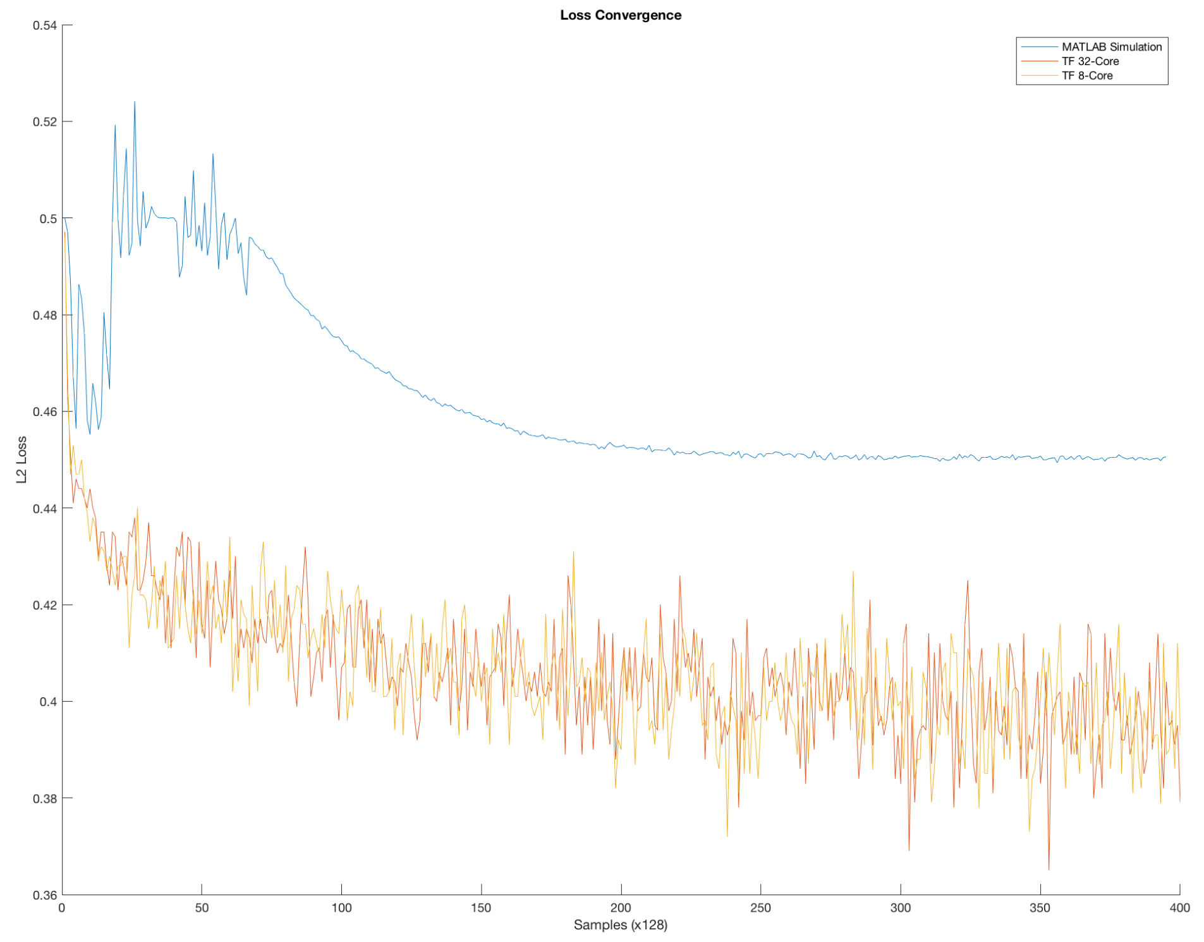
- Convolution layers are condensed into a series of parallel "filter units"
- Each unit is capable of performing feed-forward pass and backpropagation
- Units keep local register files with weights and biases


If s = Gradient Update Mode, then Weights -= Gradient Update and Bias -= sum(in2) (not shown)

Results



Limited Precision Convergence



Conclusions and Future Work

- CNNs do not fit on current FPGAs (memory constrained)
- Hardware controller does not scale well as CNN grows
- Need CNNs with smaller weight kernels
- Bulk of future work is in partitioning and communicating between FPGAs
- Use quantization noise intelligently to make CNN robust