ECE284: Low-power VLSI Implementation for Machine Learning (Instructor: Mingu Kang)

This course provides "hands-on" VLSI design guideline of the machine learning (ML) accelerator architectures across top-to-down vertical layers including algorithm, architecture, and circuit. The overview/theory of training and inference of deep neural network and other ML algorithms are provided. Students are supposed to train and validate their own network models for computer vision and natural language processing (NLP) applications via python (pytorch) programming. Then, the network model is mapped on the hardware by applying multiple low-power techniques including quantization, pruning, compression, and sparsity-aware circuit techniques. Students design their own architecture with verilog programming and verify the functionality with their test benches from python. Finally, the design is synthesized and evaluated with the Quartus Prime for FPGA emulations.

- Recommended preparation: ECE111 or equivalent course (which covers verilog & digital logic design).
- Prior knowledge of machine learning is not required to take this course.

Tentative schedule

WK #	Session #	Topics	**Hand-on examples** (ex: example, HW: homework, v: verilog, p: pytorch)
1	1 (Sep 27)	Course introduction / overview	ex1. Multiply-and-accumulator (MAC) design (v) HW. Special function unit (ReLU / accumulator) (v)
	2 (Sep 29)	ML overview, loss function, momentum, Gradient descent algorithm	ex1. Regression with gradient descent (p) ex2. Perceptron training with gradient descent (p)
2	3 (Oct 4)	Deep neural network, Back propagation	ex1. Two-layer perceptron back propagation (p) ex2. Multi-layer perceptron training for MNIST (p) HW. Manual calculation of back propagation (p)
	4 (Oct 6)	Convolutional neural network, pooling, drop out, batch normalization	ex1. Batch-normalization demo (p) ex2. CNN training for MNIST dataset (p) HW. CNN raining for CIFAR10 dataset (p)
3	5 (Oct 11)	VGGNet, ResNet, GoogleNet, DenseNet, Quantization, Number representation	HW. VGGNet & ResNet training (p)
	6 (Oct 13)	Post-traing quantization, Quantization-aware training Local vs. global, uniform vs. unequal quantization	ex1. Weight and activation quantization (p) HW. VGG16 post-training quantization (p) HW. MAC design for 2D systoic array architecture (v)
4	7 (Oct 18)	Customized loss function and gradient, 2-D systolic array architecture, Weight-stationary & output-stationary data map	HW. VGG16 quantization-aware training (p)
	8 (Oct 20)	Data and instruction flow in 2D systolic array, Workload tiling	ex1. VGGNet weight stationary hardware mapping (p) HW. VGGNet output stationary hardware mapping (p)
5	9 (Oct 25)	Processing optimization in 2D systoic array, Pruning and compression	HW. Processing element (PE) tile design (v) HW. PE row & PE array design (v)
	10 (Oct 27)	CSC & CSR & Huffman encoding, Structured vs. unstructured pruning	ex1. Unstructured pruning for MNIST (p) ex2. Structured pruning for MNIST (p) HW. Pruning for quantized VGGNet (p) HW. Input and output FIFO design (v)
6	11 (Nov 1)	Guest lecture (Sambanova)	
	12 (Nov 3)	Natural language processing with Transformer, Attention mechanism	ex1. Embedding for token (p)
7	13 (Nov 8)	Natural language processing with BERT, NLP on hardware	ex1. MeMN2N model for facebook bAbi dataset (p) ex2. Memory write and read (v) HW. MeMN2N online pruning (p) HW. Memory write and read with VGG model (v)
	14 (Nov 10)	Project consultation with instructor (Review of previous materials)	
8	15 (Nov 15)	Midterm	
	16 (Nov 17)	Project consultation with instructor	
9	17 (Nov 22)	Advanced ML architectures: In-memory computing, Mixed-signal accelerator architectures	HW. CNN for CIFAR10 in noisy Hardware (v)
	18 (Nov 24)	(optional if time permits) Other ML algorithms: Xgboost, K-NN, HD neuromorphic computing	ex1. Xgboost algorithm training (p) HW. Xgboost algorithm inference with noise (p)
10	19 (Nov 29)	Project presentation	2D systolic array based instruction set architecture (v), VGGNet mapping and functional verification (vp), Measure power/freq./area in FPGA emulation in Quartus Prime (v)
	20 (Dec 1)	Project presentation	