ECE 498NSU/598NSG - Deep Learning in Hardware

Instructor: Naresh Shanbhag **TAs:** Rex Geng and Howard Li

Prerequisites: ECE 313 and ECE 385 (498NSU);
ECE 313 and ECE 482 (598NSG);

Text: List of papers and instructor notes; **Lecture**: Tu and Th 11:00-12:20, ONL

Course Description: This course will present challenges in implementing deep learning algorithms on resource-constrained hardware platforms at the Edge such as wearables, IoTs, autonomous vehicles, and biomedical devices. Fixed-point requirements of deep for deep neural networks and convolutional neural networks including the back-prop based training will be studied. Algorithm-to-architecture mapping techniques will be explored to trade-off energy-latency-accuracy in deep learning digital accelerators and analog in-memory architectures. Fundamentals of learning behavior, fixed-point analysis, architectural energy and delay models will be introduced in just-in-time manner throughout the course. Case studies of hardware (architecture and circuit) realizations of deep learning systems will be presented. Homeworks will include a mix of analysis and programming exercises in Python and Verilog leading up to a term project.

Syllabus

Being a first-time offering, this outline has been designed to provide for some flexibility by allowing the instructor to choose a large subset of the listed topics. The list will become more precise in the second offering.

1. **Introduction (Week 1):** modern day applications in human-centric (e.g., biomedical/wearable devices) and autonomous (unmanned vehicles) platforms. Historical overview of AI, connections to neuroscience, early single stage neural networks (ADALINE, perceptron). Introduction to DNNs, standard datasets, networks, inference tasks.
2. **Deep Neural Networks (Weeks 2-6):** algorithmic view of DNNs and CNNs, design of minimum precision fixed-point dot-products (DPs) kernels, fixed-point DNNs for inference, the LMS algorithm, the stochastic gradient descent (SGD) algorithm and its use in DNN training using backpropagation, fixed-point DNN training, low-complexity DNNs, reducing DNN complexity using training. Estimating computational and representational (storage) costs.
3. **DNN Accelerators (Weeks 7-9):** algorithm transforms for mapping algorithms to architectures, roofline and floorline plots, data reuse, systolic architectures, case studies of digital , data-flow models of fixed-point deep learning algorithms. Efficient algorithm-to-architecture mapping techniques including data reuse, output, weight and row stationary architectures. Neuromorphic architectures. Energy and latency models to estimate and compare associated costs of various mapping techniques and explore trade-offs. Case studies of digital deep learning architectures (Eyeriss, DianNao series, TPU, Cambricon, TrueNorth), and practical IC realizations.
4. **In- and Near Memory Architectures (Weeks 11-14):** DRAM-based (e-DRAM), 3D architectures (HMC, HBM), SRAM-based deep in-memory architectures, architectures based on non-volatile resistive memories (RRAM PCM, CBM crossbars). Energy, latency and accuracy trade-offs in analog computation. Case studies of architectures (ISAAC, PRIME) and practical IC realizations.
5. **The Future (Week 15):** challenges and opportunities in deep learning hardware – designing programmable architectures, Shannon-inspired models of computation, developing CAD design methodologies, enabling emerging beyond CMOS fabrics, obtaining fundamental limits, and others.

Grading: Course grade will be based on homeworks (40%) involving Python and Verilog programming well as design and analysis problems, mini-project (NSU)/paper review (NSG) (20%), and a term design project (NSU)/research project (NSG) (40%). Since this course will be taken by ECE 598NSG students as well, each homework will have an extra problem or two specifically for the graduate students. These problems will be optional for ECE 498NSU students.

Lecture Schedule				
Wk #	**Lec. #**	**Date**	**Topic**	**Remarks**
1	1	25-Aug	Introduction - Applications, History, Elements, Course	Paper Sign-up
	2	27-Aug	Deep Networks - inference tasks, standard datasets, network structure, standard networks	
2	3	1-Sep	Deep Networks - function, linear predictor, design of networks	
	4	3-Sep	Finite-precision Dot-Product - quantization, number representations	HWK 1 assigned
3	5	8-Sep	Finite-precision DNNs (Inference): precision analysis framework, accuracy vs. precision	
	6	10-Sep	DNN Training - Overview and the LMS Algorithm	
4	7	15-Sep	DNN Training - LMS Example (continued)	
	8	17-Sep	DNN Training - SGD and its variants, Backprop	HWK 1 due/HWK 2 assigned
5	9	22-Sep	Training DNNs in Fixed-point with LMS Example	
	10	24-Sep	Low-complexity DNNs - SqueezeNet, MobileNet, ShuffleNet, ThunderNet	
6	11	29-Sep	Reduced complexity DNN via Training - learned quantization & model compression	
	12	1-Oct	DA - Mapping Algorithms to Architectures Systematically - Algorithm Transforms	HWK 2 due/HWK 3 assigned
7	13	6-Oct	DA - Mapping Algorithms to Architectures Systematically - Algorithm Transforms	
	14	8-Oct	DNN Accelerators - Overview, Roofline & Floorline plots, Reuse opportunities	
8	15	13-Oct	DNN Accelerators - Systolic Architectures, TPU and Eyeriss Case Studies	
	16	15-Oct	DNN Accelerators - Case studies (DianNao, IBM chip, Minerva, GraphCore)	
9	17	20-Oct	DA - Methodology (Eyeriss-v2)	
	18	22-Oct	DA - Fundamental Energy-Delay Trade-offs	
10	19	27-Oct	DA - Statistical Error Compensation - I	HWK 3 due/HWK 4 assigned
	20	29-Oct	DA - Statistical Error Compensation - II	
11	21	3-Nov	Deep In-memory Architectures (DIMA) - Introduction	598 Project Proposals due
	22	5-Nov	DIMA - UIUC Case Studies	
12	23	10-Nov	Project Overview	HWK 4 due, NSU Project Assigned; NSG Project Finalized
	24	12-Nov	DIMA - UIUC Case Studies	
13	25	17-Nov	DIMA - Others	
	26	19-Nov	DIMA - A Compositional Framework	
11/21/2020-11/29/2020 **Thanksgiving Break**				
14	27	1-Dec	DIMA - A Compositional Framework	
	28	3-Dec	Advanced Topics - Non von Neumann Computing	
15	29	8-Dec	Advanced Topics - Non von Neumann Computing	
		11-Dec	Project Report Due	