ECE 498NSU/NSG – Resource-efficient Machine Learning for the Edge

Instructor: Naresh Shanbhag **TA:** Han-Mo Ou

Prerequisites: ECE 313 and ECE 385 or instructor's consent

Text: List of papers and instructor notes; **Lecture**: Tu and Th 11:00-12:20, ECEB 2013

Instructor Office Hours: Thursdays 2PM-3PM, CSL 414

TA Office Hours: Fridays 1PM-2PM, ECEB 3034

Course Description: This course will present challenges in implementing machine learning algorithms on resource-constrained hardware platforms at the Edge such as wearables, IoTs, autonomous vehicles, and biomedical devices. Simple single-stage classifiers will be discussed first followed by deep neural networks. Finite-precision analysis will be employed to design fixed-point networks to minimize energy, latency, and memory footprint. Training algorithms of both single-stage and deep nets (back-prop) will be presented followed by their fixed-point realizations. 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 just-in-time throughout the course. Case studies of hardware (architecture and circuit) realizations of deep learning systems will also be presented. Homeworks will include a mix of analysis and programming exercises in Python and Verilog leading up to a term project involving the implementation of deep nets on an embedded hardware platform such as an FPGA/MCU. Graduate students additionally will submit a term paper based on the literature review of a specific topic of their interest.

Grading: Course grade will be based on weekly homeworks (NSU: 30%, NSG: 25%) involving Python and Verilog programming well as design and analysis problems, one midterm (NSU: 30%, NSG: 25%), a term design project (NSU only: 40%), and term research project (NSG only: 30%), and a term paper (NSG only: 20%).

Syllabus

The following is a rough outline of the course. The topics have been re-sequenced from previous offerings to begin with hardware architectures before moving into algorithms.

1. **Introduction (wk 1):** requirements of human-centric (e.g., biomedical/wearable devices) and autonomous (unmanned vehicles) Edge applications; historical overview of AI and early neural networks (ADALINE, MADALINE, perceptron, Hopfield networks); hardware platforms for the Edge (GPU, CPU, FPGA, MCU), their resource constraints and energy/latency costs; roofline/floorline model of digital accelerators and the relationship between memory BW, peak and achievable performance, and energy efficiency
2. **Computing with Finite-Precision (wk 2):** signal quantization (uniform, non-uniform Lloyd-Max, OCC); fixed-point (FX) dot-products.
3. **Learning in Finite-Precision (wk 3):** obtaining optimum weight parameters of a linear predictor; learning using the least mean-squared (LMS) algorithm (FL and FX); interplay between learning behavior, precision of computation, energy, and latency of LMS on Edge platforms.
4. **Shallow Networks (wk 4):** the support vector machine (SVM) and single-layer perceptron; ensemble methods (random forest, AdaBoost); spiking neural networks. Interplay between learning behavior, precision of computation, energy, and latency on Edge platforms.
5. **Deep Neural Networks (wk 5):** DNN structure, overview of popular deep nets and datasets; minimum precision DNN inference; lightweight DNNs (MobileNet, SqueezeNet, ShuffleNet).
6. **Training DNNs (wk 6):** training DNNs (SGD, backprop, variants); training in finite-precision, estimating computational and storage costs; quantization-aware training (QAT) methods.

7. **Compressing DNNs (wk 7):** Reducing network complexity via model compression and network architecture search (NAS).
8. **Pushing the Limits (wk 8):** Algorithm-to-architecture mapping techniques; Statistical error compensation to push the limits of energy-latency trade-offs;
9. **Case studies of DNN accelerators (wk 9):** TPU, Eyeriss, diannao family. Case studies presentations by graduate students (UNPU).
10. **In-Memory Computing (wk 10):** In-memory architectures for SRAM and embedded non-volatile resistive memories. Energy, latency and accuracy trade-offs and design methods for in-memory computing. Case studies of in-memory architectures.
11. **The Future (wk 11):** challenges and opportunities in resource-constrained machine learning; adversarial robustness issue;