EE P 566: Deep Learning for Embedded Real Time Intelligence

Overview
The course focuses on deploying deep learning in performance-critical applications across industries like Google, Amazon, Microsoft, and Boeing. The course aims to teach effective integration strategies for real-time intelligent applications with limited compute resources. We will combine theoretical and programming components, including DNN efficiency and performance, parallelism and data reuse, optimize latency and memory usage, leverage open-source resources, etc., for applications across vision, NLP, and medical systems.

Prerequisites:

- EE 235/341 (Linear and Discrete-Time Signal Processing)
- Programming in C/C++ (CSE 374 or Equivalent)
- Familiarity with basics of machine learning

Goals:

- Understand the fundamentals of modern deep neural networks
- Learn algorithmic techniques to improve efficiency and real-time performance of DNNs, including rank approximations, spiking, depth separation, bit-width quantization, sparsification, and more
- Analyze parallelism and data reuse in DNNs
- Optimize latency and memory usage (RAM) for practical deployment in performance-critical applications with limited compute resources
- Utilize open-source resources like HALDL (developed by MIT CSL) for deep learning hardware
- Implement real-time software programming paradigms

Grading:

- Attendance/Participation: 10%;
- Four Assignments; Each 10%;
- Final Project: 50% (project presentation+demo: 20%; final project report: 30%)

Late Policy:

- Extensions: A maximum of one day may be granted.

- **Requests must be made at least 3 days before the deadline with a valid reason.**
- Exceptions: For emergencies (e.g., sickness), the 3-day notice is waived.
- Penalties: Otherwise, a 20% penalty is applied per late day.
- **Please send your request to kw99@uw.edu or message Andy K Wu on Slack.**

AI Usage Policy:

- **Permitted Use:** You may use AI tools to help understand concepts or address specific technical questions.
- **Restrictions:** Do not use AI to generate your entire answer or solution.
- **Originality:** All discussions and analytical content must reflect your own original thoughts and understanding.

Tentative Organization:

Module 1 Introduction to Artificial Neural Networks and Deep Learning

- History of Neural Networks; Hopfield models
- Recent explosion in data-driven deep learning
- Introduction to MCU/CPU, GPU, DSP, FPGA and domain-specific DL Accelerator
- Introduction to Software stacks for Deep Learning
- Brief introduction of three applications: object detection, voice recognition and medical data analysis
- .Embedded computing: Azure RTOS C programming on MCUs

Module 2 Deep Learning Concepts and Constructs

- Convolution: depth-wise, point-wise
- Transformer, encoder, decoder
- Attention
- Dilation and reception fields
- Skip, residual connection and 1x1 coolution
- Gating
- Relu, Pooling, Activation

- KV Cache
- Mixture of Experts

Module 3 Techniques for Implementing Deep Learning on Embedded Devices

- Quantization
- Sparsification
- Channel separation
- Network disfill: teacher-student-model
- Techniques for Parallelism and Data Use in Deep learning

Module 4 Halide language

- Tensor variables and data objects expressions
- Vectorize, parallelize, unroll, and tile your code
- Realize functions over arbitrary domains
- Multiple-state pipelining and scheduling
- Multiple passes, data updates and reduction
- Cross compilation and compiler optimization

Module 5 Convolution Neural Networks

- AlexNet
- ResNet
- Yolo
- MobileNet
- SqueezeNet

Module 6 Recurrent Neural Networks

- RNN
- LSTM
- GRU

Module 7 Graph Neural Networks

- Basics of Graph
- Convolutional GNN
- LSTM/GRU GNN

Module 8 Spiking Neural Networks

- Leaky-Integrate-Fire (LIF) model
- An R-C-Rectifier Implementation (Nature 2019 paper)
- The Legendre memory units (LMUs)
- Hybrid Spiking Neural Networks