# 20ES021 - Hardware Architectures for Deep Learning

**UNIT I**

**Deep learning and approximate data representation**

Introduction – Background - Stochastic computing - Deterministic low-discrepancy bit-streams - Convolutional neural networks - Convolutional neural networks - Related work - Proposed hybrid binary-bit-stream design - Multiplications and accumulation - Handling negative weights - Experimental results - Performance comparison - Cost comparison - Binary neural networks - Binary and ternary weights for neural networks - Binarized and ternarized neural networks - NN optimization techniques - Hardware implementation of BNNs

**UNIT II**

**Deep learning and model sparsity**

Different types of sparsity methods - Software approach for pruning - Hard pruning - Soft pruning, structural sparsity, and hardware concern - Questioning pruning - Hardware support for sparsity - Advantages of sparsity for dense accelerator - Supporting activation sparsity - Supporting weight sparsity - Cambricon-X - Bit-Tactical - Neural processing unit - Supporting both weight and activation sparsity - Efficient inference engine – ZeNA - Sparse convolutional neural network - Supporting output sparsity – SnaPEA - Uniform Serial Processing Element
- SparseNN – ComPEND - Supporting value sparsity - Bit-pragmatic - Laconic architectur.

**UNIT III**

Computation reuse-aware accelerator for neural networks

Baseline architecture - Computation reuse support for weight redundancy - Computation reuse support for input redundancy - Multicore neural network implementation – More than K weights per neuron - More than N neurons per layer

**UNIT IV**

**Convolutional neural networks for embedded systems**

Brief review of efforts on FPGA-based acceleration of CNNs - Network structures and operations – Convolution - Inner product – Pooling - Other operations - Optimizing parallelism sources - Identifying independent computations - Acceleration strategies - Computation optimization and reuse - Design control variables - Partial sums and data reuse - IFMs first strategy - OFMs first strategy - Proposed loop coalescing for flexibility with high efficiency - Bandwidth matching and compute model - Resource utilization - Unifying off-chip and on-chip memory - Impact of unmatched system - Effective bandwidth latency - Analyzing runtime - Estimating required off-chip bandwidth - Library design and architecture implementation - Concurrent architecture - Convolution engine - Optimal DRAM access - Restructuring fully connected layers - Zero overhead pooling - Other layers - Caffe integration - Performance evaluation - Optimizer results - Latency estimation model - Exploration strategy - Design variable optimization - Onboard runs - Network-specific runs - Cross-network run - Architecture comparison - Raw performance improvement.

**UNIT V**

**Iterative convolutional neural network (ICNN): an iterative CNN solution for low power and real-time systems**

Optimization of CNN - Iterative learning - Case study: iterative AlexNet - ICNN training schemes

- Sequential training - Parallel training - Complexity analysis – Visualization - Background on CNN visualization - Visualizing features learned by ICNN - Contextual awareness in ICNN - Prediction rank - Pruning neurons in FC layers - Pruning filters in CONV layers - Policies for exploiting energy-accuracy trade-off in ICNN - Dynamic deadline (DD) policy for real-time applications - Thresholding policy (TP) for dynamic complexity reduction - Fixed thresholding policy – Context-aware pruning policy - Context-aware pruning policy for FC layer - Context-aware pruning policy for CONV layer

Pruning and thresholding hybrid policy - Fixed percentage PTHP - Confidence-tracking PTHP - Variable and dynamic bit-length selection - ICNN implementation results - Implementation framework - Dynamic deadline policy for real-time applications - Thresholding policy for dynamic complexity reduction - Fixed thresholding policy - Variable thresholding policy - Context-aware pruning policy for parameter reduction - Context-aware pruning policy for FC layer - Context-aware pruning policy for CONV layer - Pruning and thresholding hybrid policy - Fixed percentage PTHP - Confidence-tracking PTHP - Run-time and overall accuracy - Pruning and/or thresholding - Deadline-driven

**Text books**

1. Hardware architecture for deep learning Masoud Daneshtalab [1]; Mehdi Modarressi [2] IET Publications Year : 2020.