ACACES'22 – 18th International Summer School on Advanced Computer Architecture and Compilation for High-Performance Embedded Systems

Characterizing DNN Workloads on General Purpose CPUs

Phd Student: Nitesh Narayana Gondlyala Sathya, **Advisors:** Franyell Silfa, Antonio González

Department of Computer Architecture, Universitat Politècnica de Catalunya

Email: nitesh.narayana.gondlyala@upc.edu

Motivation

DNN applications are now being deployed on mobile and embedded platforms

CPUs

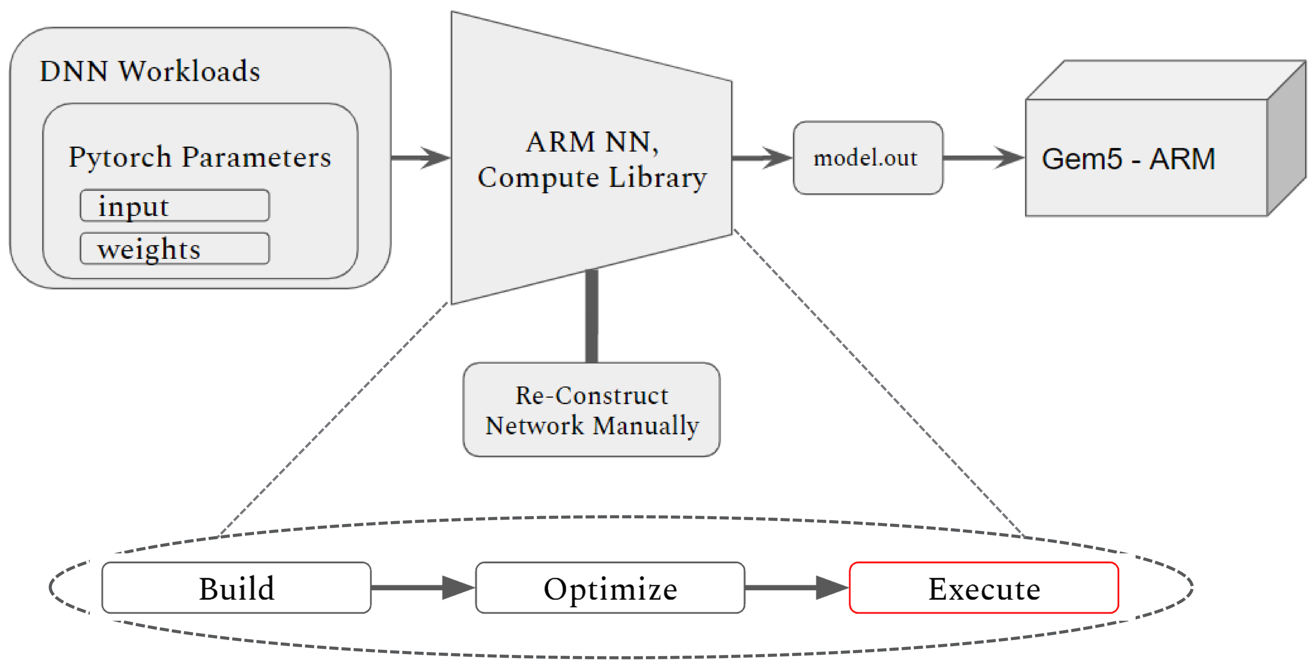
- Have high flexibility and availability
- Provide easy integrations between DNN workloads and business services
- Have low cost of design and deployment compared to accelerators
- Readily available mature and optimized software stacks
- Widely deployed in data centers, client and edge devices

Recent SIMD ISA extensions have enhanced CPU performance in data parallel computing

Important to understand the behavior of DNNs on CPUs to develop CPU tailored optimizations for DNN Inference.

Methodology

- Extract each networks architecture, weights from PyTorch
- Port the networks to ARM-NN, compile using ARM-GCC
- Execute the networks on gem5 Simulator to generate stats



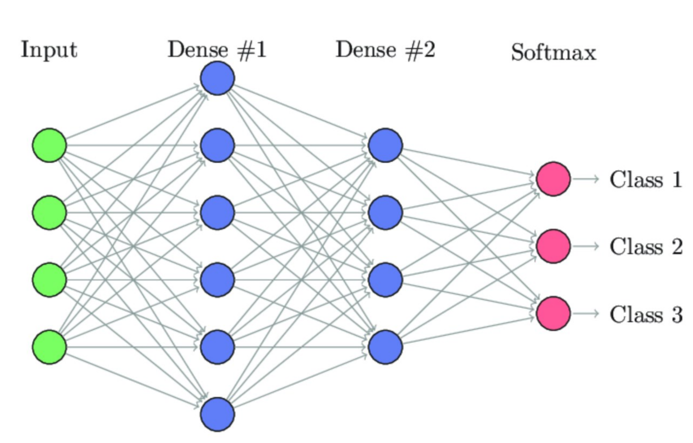
Simulation work flow

ARM-NN work flow

Background

ARM Cortex-A76 like CPU Configuration

CPU (@1.5GHz)	Functional Units	Cache	DRAM
128 Int RF, 192 FP RF 48x128-bit Vector RF	2xInt ALUs, 2xInt Vector/FP FUs 2xLoad + 1xStore	64KB 4-Way LRU L1-I/L1-D 256KB 8-Way LRU L2	8GB Dual-Channel DDR3-1600



Sample DNN

- DNN with four layers
- Input and Softmax are input and output layers respectively
- Dense#1 and Dense#2 are hidden layers
- All layers are fully connected

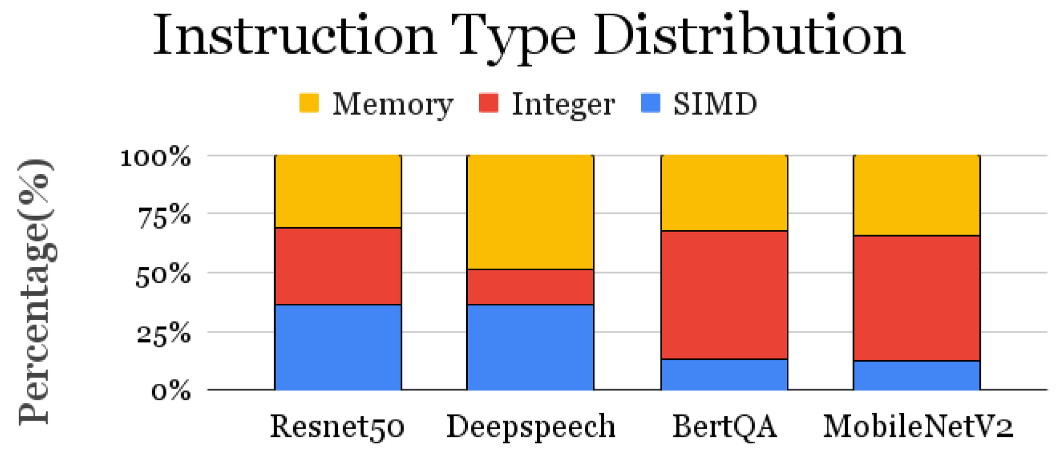
Summary of DNN workloads

Network	App Domain	Network Type	# Layers	# Params	Memory Footprint(MB)	Base Accuracy	Dataset
BERT-QA	Question Answering	Transformer	24	340M	1208	exact_match: 84.1%	SQUAD
MobileNet-V2	Image Classification	Convolution	53	3.4M	14	Acc@1 71.8 %	ImageNet
Resnet50	Image Classification	Covolution	50	25.6M	204	Acc@1 77.5 %	ImageNet
DeepSpeech2	Speech Recognition	LSTM	5	52M	120	WER 10.24	LibriSpeech

- Subset from MLPerf Inference Benchmark Suite
- MobileNetV2 and Resnet50 take images as input and predict their classes
- BertQA takes contexts(paragraphs) and questions to give start and end index of the answer from the paragraph
- DeepSpeech takes and audio file as input and gives transcription of the audio

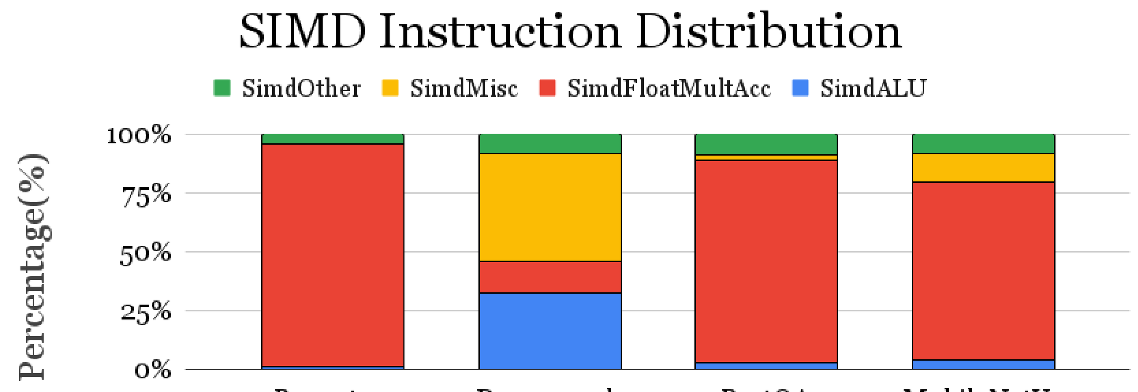
Results

Instruction Type Distribution



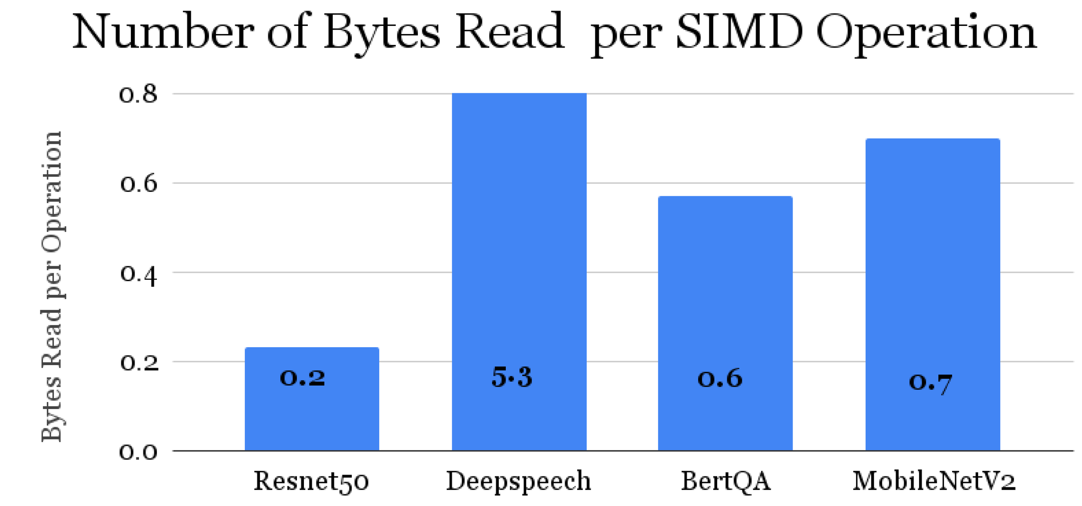
- Memory -> loads or stores (scalar or vector), mainly performed to fetch weights/inputs and store each layers' output
- SIMD -> instructions executed in SIMD unit, used to evaluate each layer, i.e., perform matrix-vector multiplications
- Integer -> instructions executed in Integer units, mainly added by ARMNN framework to manage and orchestrate the execution of the model

SIMD Instruction Distribution



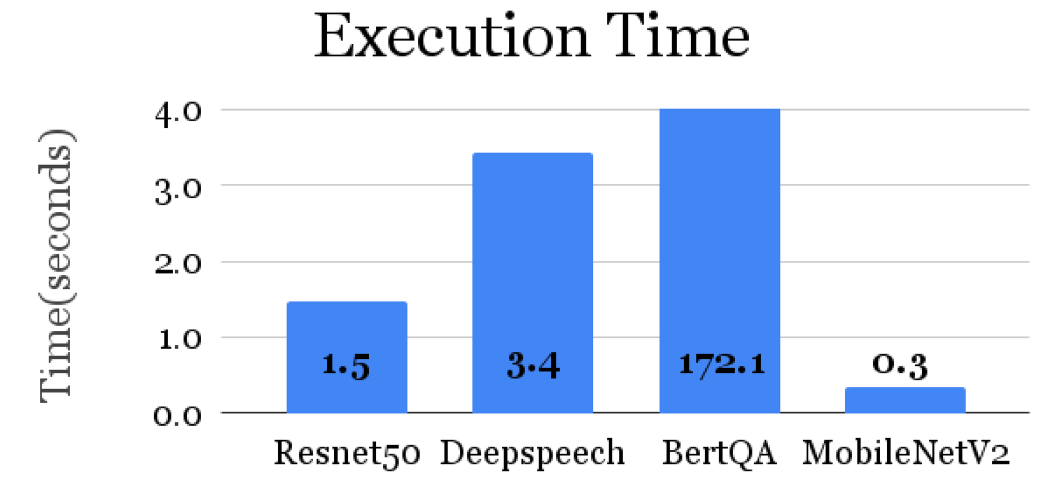
- SimdFloatMultAcc -> includes FMLAs and its variants
- SimdAlu -> includes `vabs, fmul, fexp`; used to compute activation functions
- SimdMisc -> includes `vmov, tlb, sqxt`
- SimdOther -> includes `simdCmp, simdShift`
- ML Models predominantly use vector-matrix multiplications -> FMLAs
- Deepspeech needs many computations for gating and activation related operations at each LSTM gate

Number of Bytes Read per SIMD Operation



- SIMD Operation -> an operation executed in any vector lane
- Several operations are performed for once loaded value for Resnet50, BertQA and MobileNet -> Bytes per operations < 1
- Recurrent nature of layers in LSTM makes it difficult for weight reuse

Execution Time



- Large models require many computations and data transfers
- BertQA and Deespeech have large execution time
- MobileNetV2 is designed for low-power devices, incurs lowest latency

Conclusions

Optimizing SIMD unit in-terms of operations and reducing memory activity can be a good starting point to improve DNN performance and energy efficiency on general purpose CPUs