PoET-BiN: Power Efficient Tiny Binary Neurons

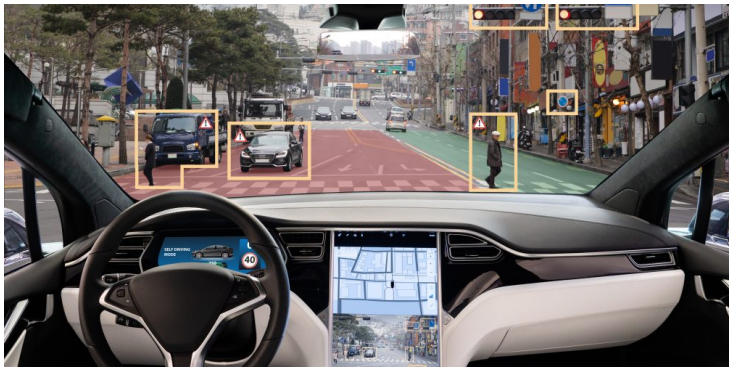
Sivakumar Chidambaram[1], J.M. Pierre Langlois[2], Jean Pierre David[1]

Department of Electrical Engineering [1]
Department of Computer and Software Engineering [2]
Polytechnique Montréal
Montréal, Canada

03 March 2020

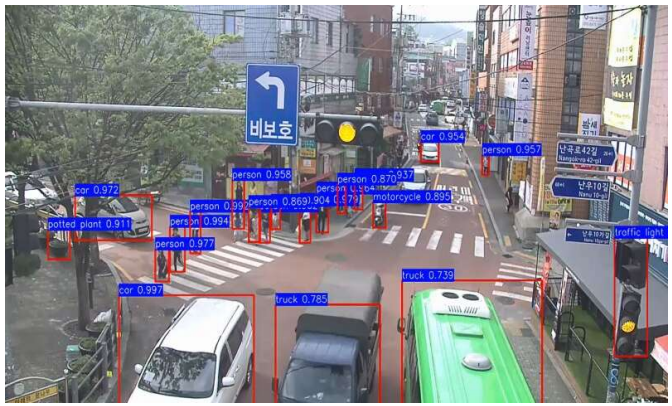
Contents

Real-time deep learning use cases



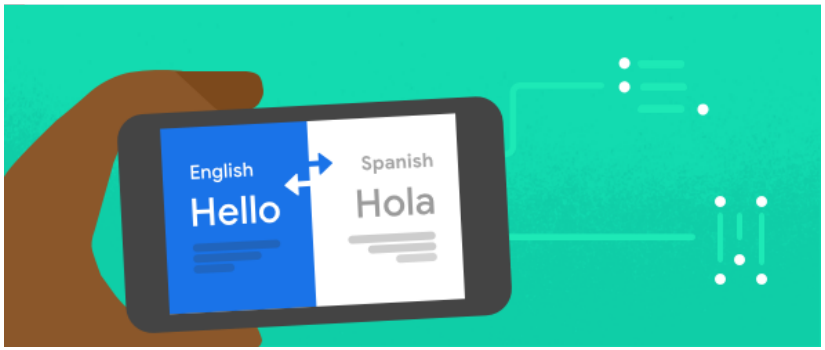
Autonomous Driving

www.alten.com/sector/automotive/next-generation-camera-based-adas-development/



CCTV Monitoring

www.munhwa.com/news/view.html?no=2019100101



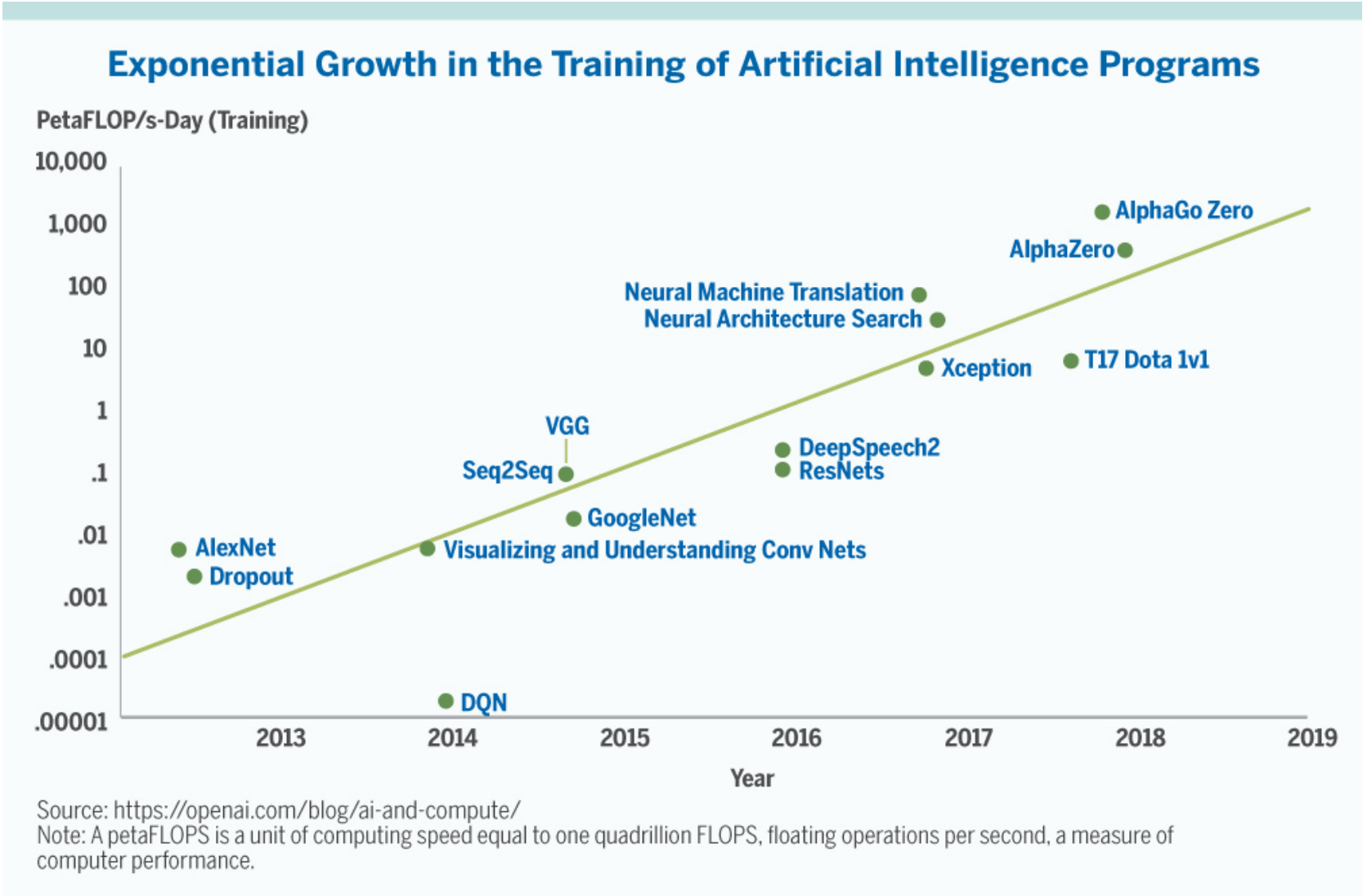
Translation

Source : www.firebase.google.com/docs/ml-kit/translation

Required Attributes :

- Accuracy
- Latency and Throughput
- Power and Energy constraints
- Memory and Hardware costs

Computation needs



Exponential rise in computations

Current Deep Learning Software Acceleration Techniques

Quantized Neural Networks

- Quantization of weights and activations
- Binarizing, Ternarization, Multi-bit quantization
- Helps in generalization on the unseen data

Pruning - Remove certain neurons from the vanilla neural network

- A bagging technique that averages various randomly pruned networks
- Introduces noise in the system that helps perform better on unseen data

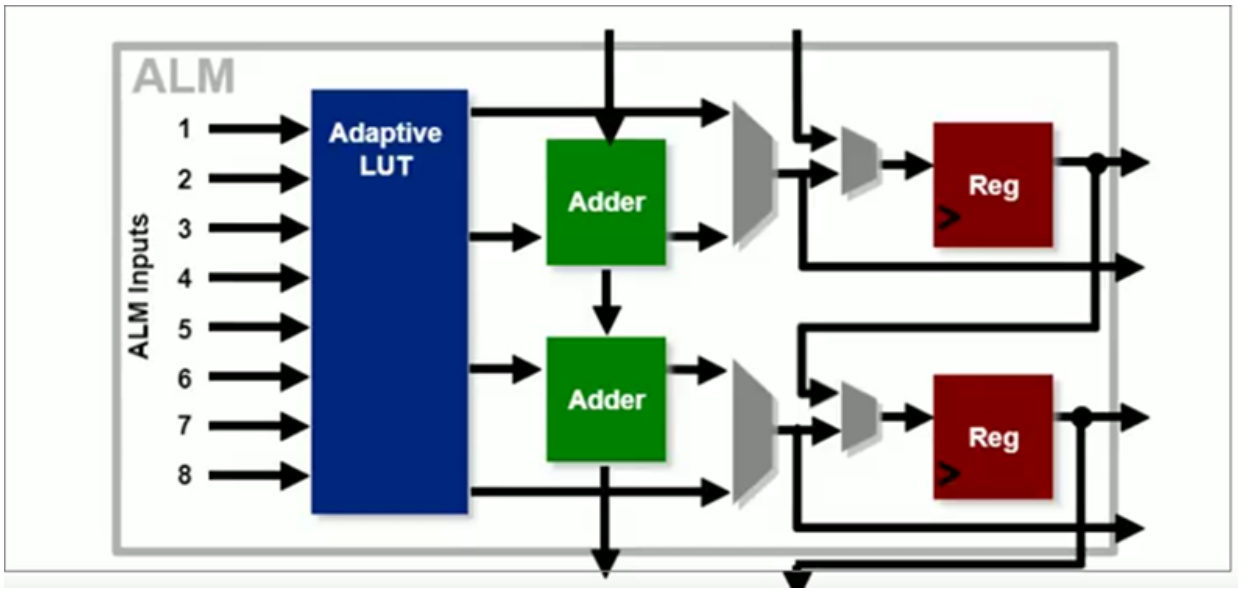
Sparsification - Sparse matrix multiplication

- Removing connections between neurons
- Reduces the number of multiplication and additions
- Reduces number of memory reads

- Implemented on hardware devices such as FPGA, microprocessors, microcontrollers etc.

Hardware : FPGAs

- Goto device for rapid prototyping of accelerators
- FPGAs consist of Arithmetic Logical Modules (ALMs), programmable interconnects, IOs and BRAMs
- ALMs are the main computational unit
- 100,000s of ALMs in a typical FPGA
- Each ALM has a LookUp Table (LUT) with up to 8 inputs and up to 2 outputs
- Programmed using Hardware Description Languages



Source : https ://hackaday.com/2018/03/01/another-introduction-to-fpgas/

Problem Definition and Objectives

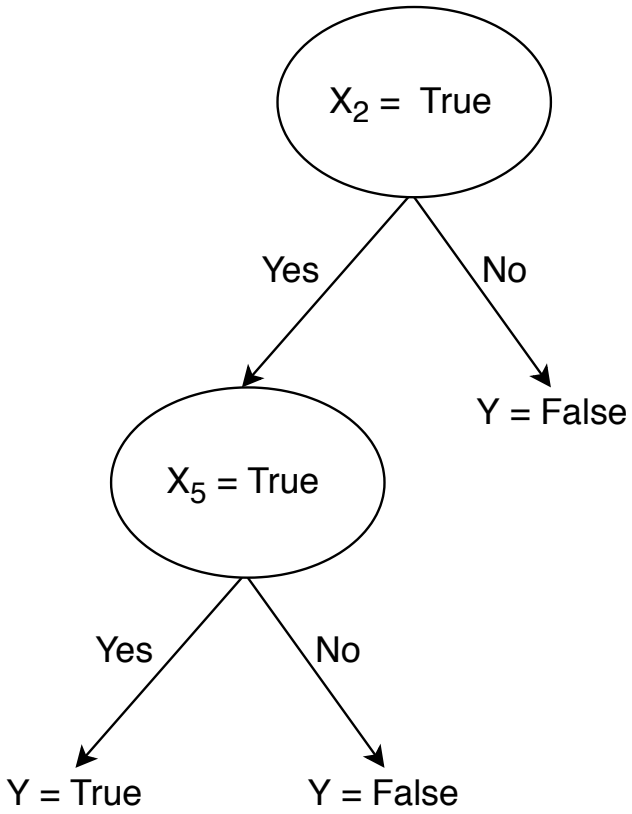
Problem Definition :

- Vanilla neural networks are computation, power and area intensive
- Current acceleration approaches are still computationally intensive
- Quantized neural networks and pruning are not optimized for FPGAs

Objectives/Contributions :

- A modified Decision Tree training algorithm to better match LUTs with a fixed number of inputs.
- The Reduced Input Neural Circuit (RINC) : A LUT-based architecture founded on modified Decision Trees and the hierarchical version of the well known Adaboost algorithm to efficiently implement a network of binary neurons.
- A sparsely connected output layer for multiclass classification.
- The PoET-BiN architecture consisting of multiple RINC modules and a sparsely connected output layer.
- Automatic VHDL code generation of the PoET-BiN architecture for FPGA implementation.

Binary Decision Trees



Binary Decision Trees

What are Binary DTs :

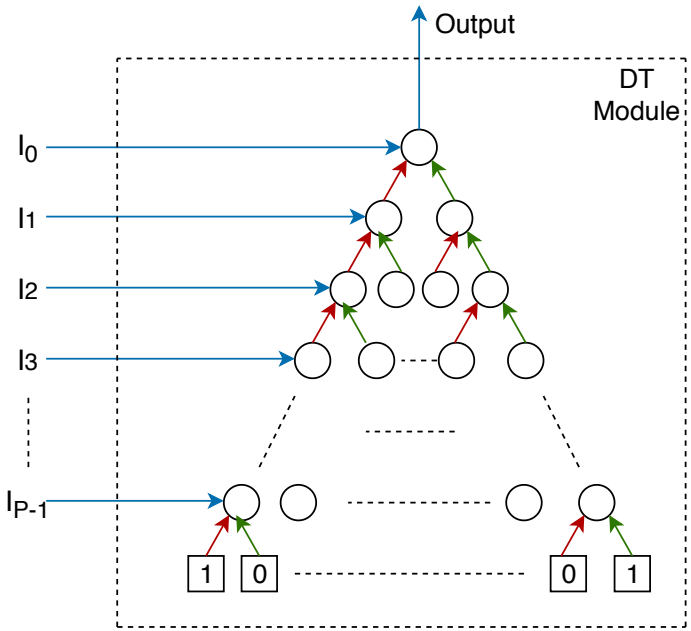
- Inputs(X) and Outputs(Y) are binary
- Node wise creation of Decision Tree from root to leaves
- The feature that most reduces the entropy is chosen
- Divides the representation space to classify data

Challenges :

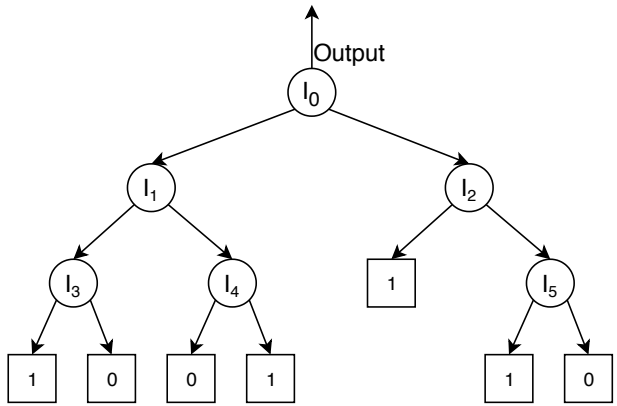
- To classify large datasets - need larger Decision Trees
- Results in large implementations on the hardware- complex and high power consumption
- To effectively implement on FPGAs we need small Decision Trees of ≤ 6 inputs to fit in one LUT

RINC-0 : Modified Decision Tree Algorithm

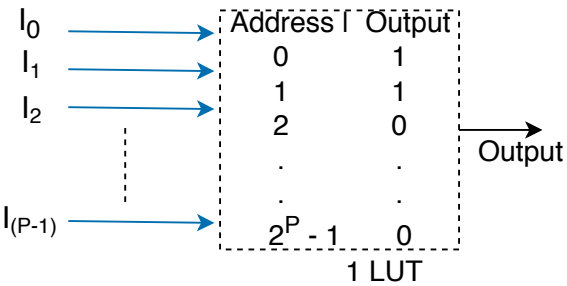
- Modified DT algorithm - level based entropy reduction rather than node based
- Decision Trees are restricted by the number of inputs(I)
- A node-wise off-the-shelf 6-input Decision Tree would have only 7 leaf nodes
- Level-wise Decision Tree will have $2^6 = 64$ leaf nodes
- More granularity



Level based Decision Tree



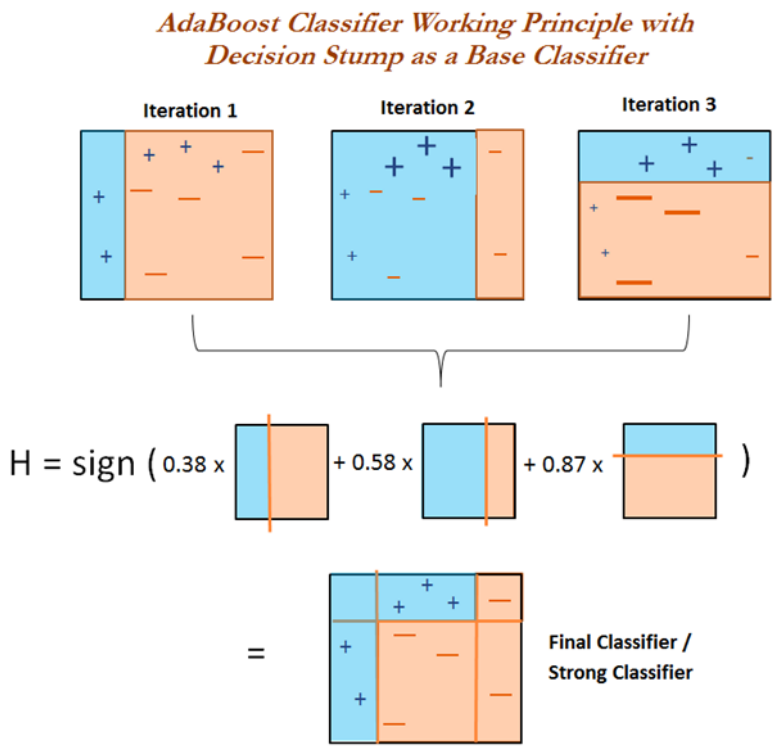
Node based Decision Tree



Decision Tree as LUT

RINC-1 : Incorporating Adaboost

- A single Decision Tree is a weak classifier
- Ensemble methods such as Boosting and Bagging are used to create strong classifiers from weak classifiers
- We use the well-know Adaboost algorithm

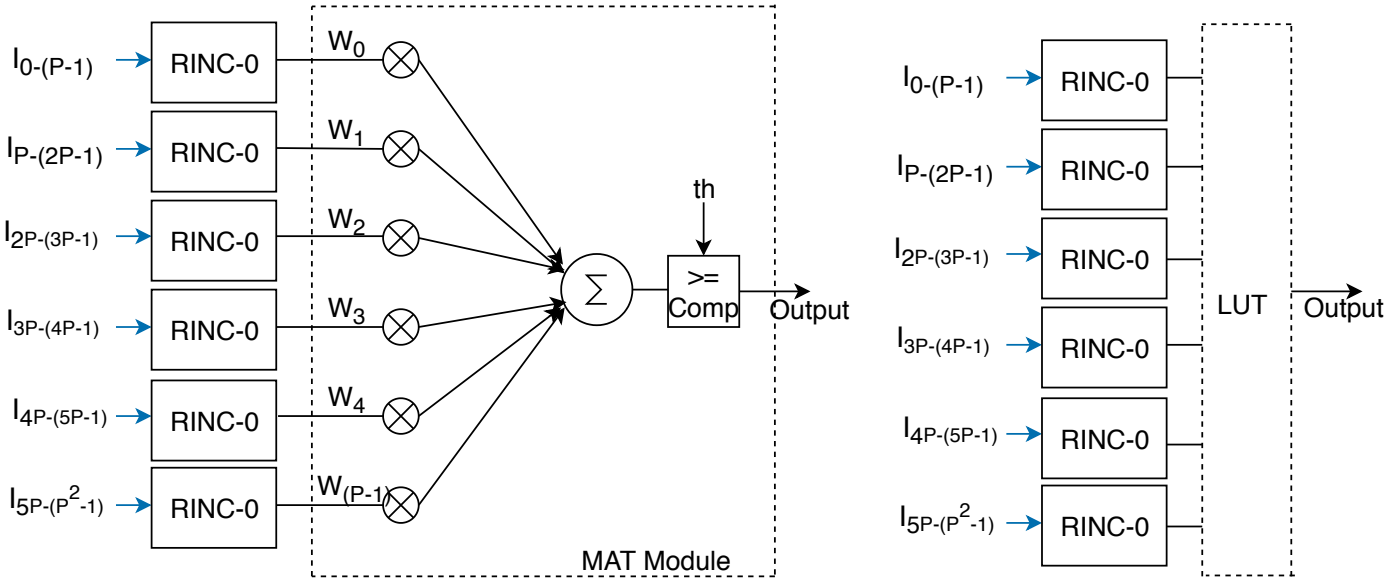


Adaboost Algorithm

Source :https ://packtpub.com/book/bigdataandbusinessintelligence/adaboost-classifier

- The weak classifiers are created serially
- The samples are given equal weight initially
- The first weak classifier is trained on the data
- The mis-classified sample's weights are increased
- Subsequent classifier focuses on the incorrect samples
- Each classifier is assigned a weight based on the number of correctly classified samples
- A weighted sum of all the weak classifier outputs forms the strong classifier

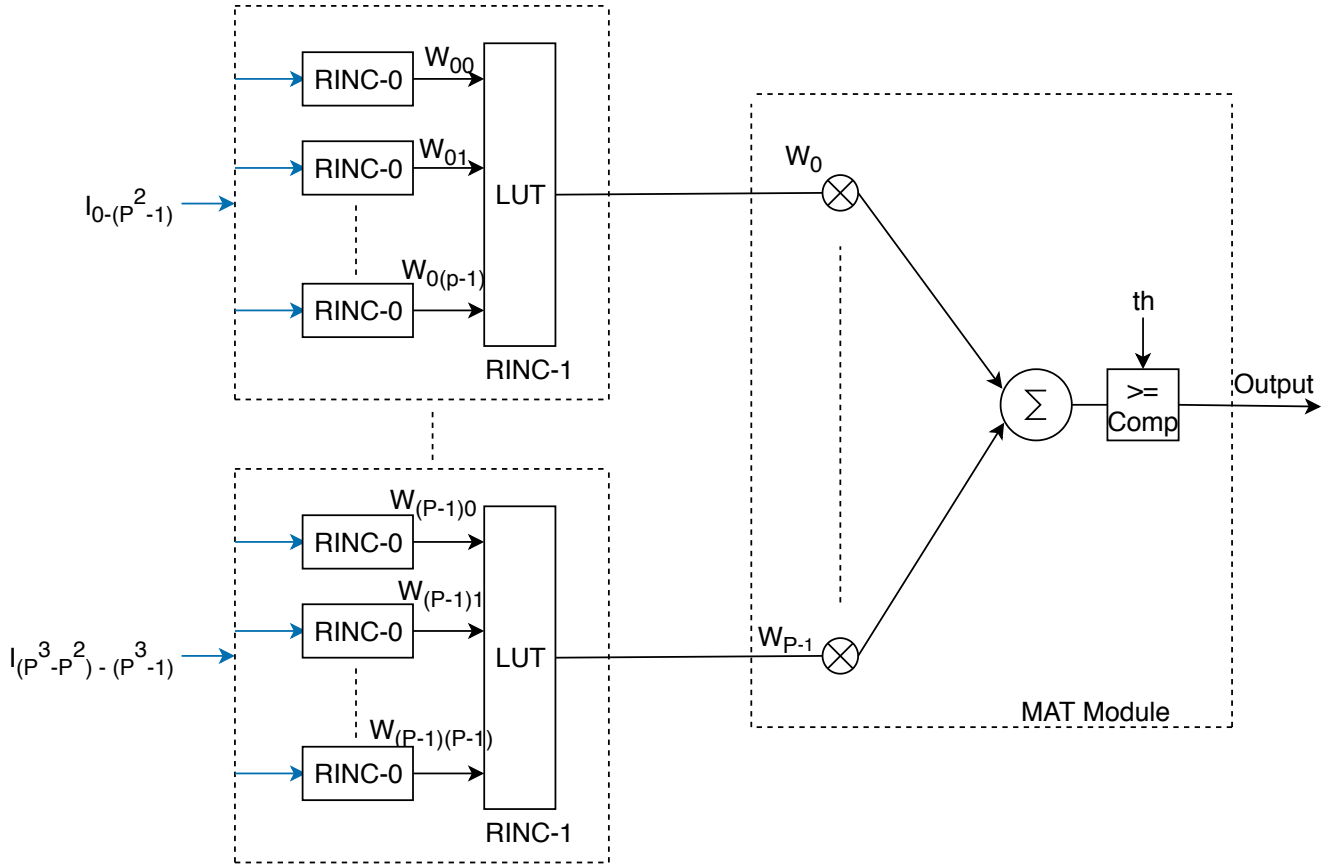
RINC-1 Module



RINC 1 module

- The MAC and threshold operations can be implemented in a LUT
- Can group up to a maximum of P Decision Trees
- However, P Decision Trees with P^2 inputs are not enough compared to a MAC operation in a neural network
- Neuron in a neural network can have up to 4096 inputs as compared 36 (when P=6) in RINC-1 modules
- Hence, we introduce the hierarchical Adaboost algorithm

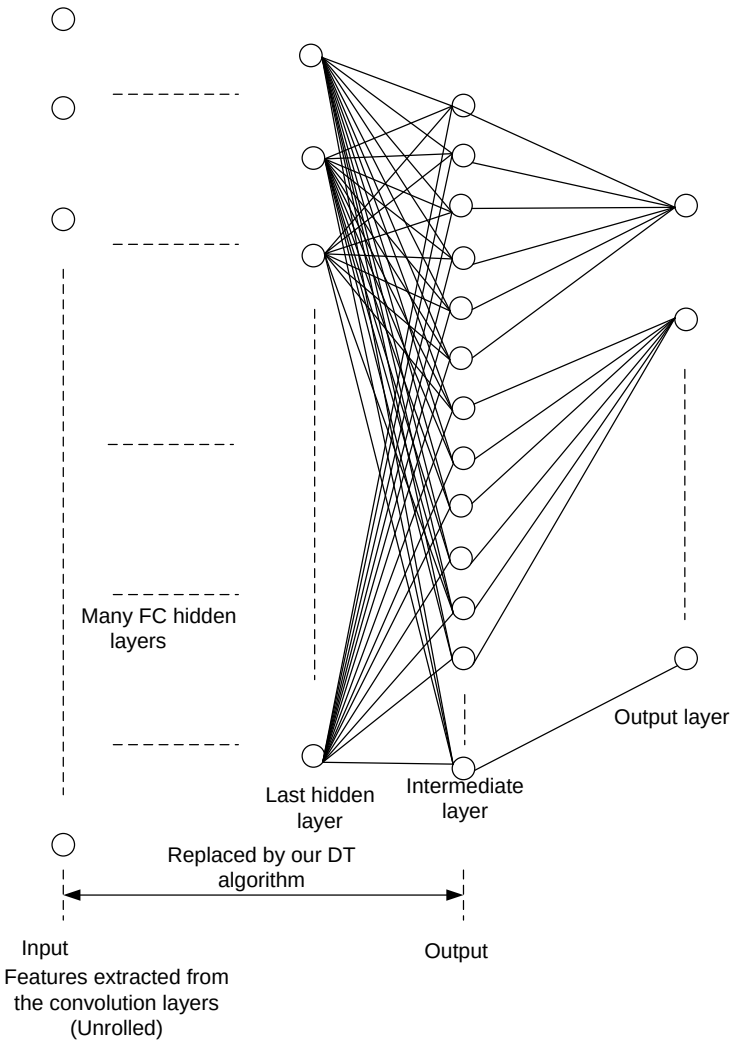
RINC-2 : Hierarchical Adaboost



RINC-2 module

- The RINC-2 modules have adequate capacity to represent MAC operations
- Can only be used for binary classifications

Binary to Multiclass Classification



Intermediate Layer

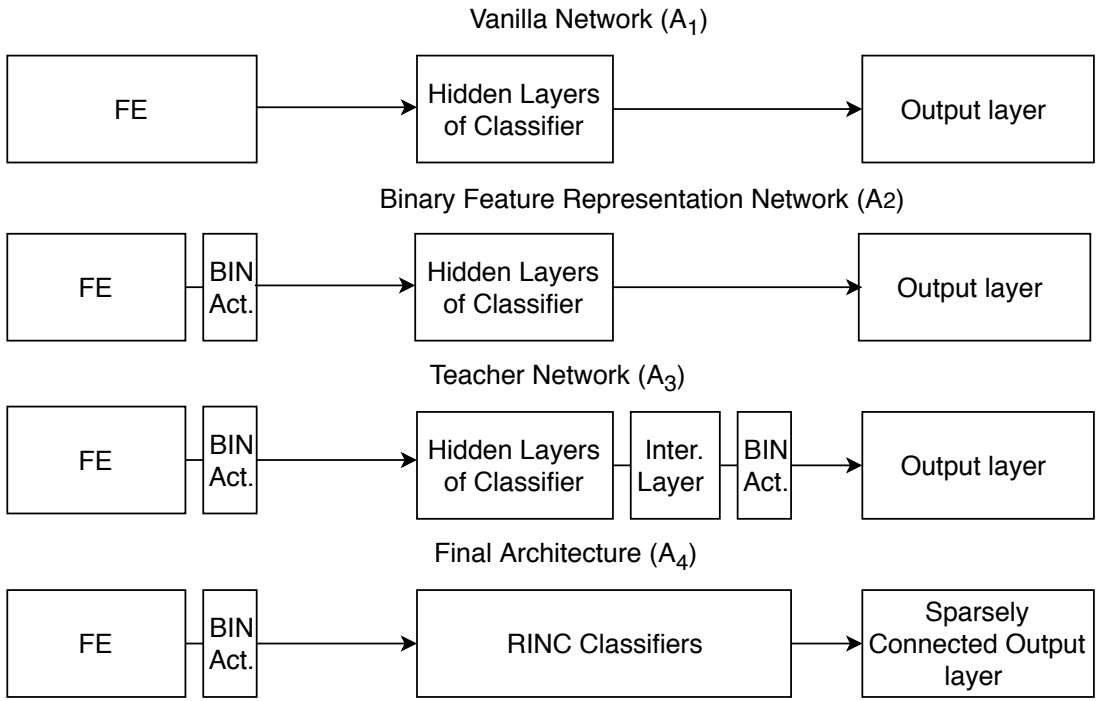
Current Methods :

- Multiclass DTs : costly to implement
- One-vs-All classification : leads to reduction in accuracy

Our Approach :

- A sparsely connected intermediate layer before the final output layer for multiclass classification
- Only P neurons of the intermediate layer connected to each neuron in the output layer
- The neurons in the output layer need to have multiple bits to represent the probabilities and cannot be binary values
- Implemented as LUTs

Experimental Setup



Experimental Setup

TABLE – Network architecture

ARCHITECTURE (ARCH.)	SYMBOL	DATASET
$LeNET_{FE} - (512FC) - (10FC)$	M1	MNIST
$VGG11_{FE} - (4096FC) - (4096FC) - (10FC)$	C1	CIFAR-10
$VGG11_{FE} - (2048FC) - (2048FC) - (10FC)$	S1	SVHN

Results : Accuracy

- A_1 : Vanilla network, A_2 : Network with binary features, A_3 : Teacher network with intermediate layer, A_4 : PoET-BiN

TABLE – Overall classification accuracy on MNIST, CIFAR-10 and SVHN dataset

ARCH.	DATASET	A_1(%)	A_2(%)	A_3(%)	A_4(%)
M1	MNIST	99.20	99.06	98.93	98.15
C1	CIFAR-10	91.02	89.88	89.10	92.64
S1	SVHN	97.36	96.98	96.22	95.13

TABLE – Comparison with other techniques

IMPLEMENTATION	ACCURACY (%)		
	MNIST	CIFAR-10	SVHN
BINARYNET(2016)	98.97	89.76	95.06
POLYBINN(2018)	97.52	91.58	94.97
NDF(2015)	99.42	90.46	95.20
OUR WORK	98.15	92.64	95.13

- There is a reduction in accuracy for each modification introduced
- Comparable accuracy with other state-of-the-art networks
- Same feature extractor
- BinaryNet - Neural Network approach
- POLYBiNN - Decision Tree approach
- NDF - Hybrid approach
- Same feature extractor for fair comparisons

Results : Power Consumption

- Measurements from Xilinx Power Analyzer tool
- Power consumption of the classification layers only

TABLE – RINC power

DATA SET	MNIST	CIFAR-10	SVHN
DYNAMIC(W)	0.468	0.300	0.374
STATIC(W)	0.045	0.041	0.043
TOTAL(W)	0.513	0.341	0.417

TABLE – Number of arithmetic operations

OP.	MNIST	CIFAR-10	SVHN
ADD.	0.26 M	18.9 M	5.2 M
MULT.	0.26 M	18.9 M	5.2 M

TABLE – Single arithmetic operation power

OPERATION (AT 62.5 MHZ)	DYNAMIC (*mW*) CLOCK	LOGIC	SIGNAL	IO	STATIC (*mW*)	TOTAL (*mW*)
MULTIPLICATION (16 BITS)	1	1	0	20	36	58
ADDITION (16 BITS)	1	0.0	1	24	36	62
MULTIPLICATION (32 BITS)	2	1	1	35	37	76
ADDITION (32 BITS)	1	0.0	2	48	37	88
MULTIPLICATION (FP)	5	6	5	46	37	98
ADDITION (FP)	4	3	5	34	37	83

Results : Energy Consumption, Latency and Hardware Costs

- The networks were implemented on a Spartan-6 Xilinx FPGA
- Energy reduction by up to three orders of magnitude when compared to recent binary quantized neural networks

TABLE – Energy consumption

TECHNIQUE	ENERGY (J) MNIST	CIFAR-10	SVHN
VANILLA	8.0×10^{-5}	5.7×10^{-3}	1.6×10^{-3}
1-BIT QUANT	2.1×10^{-7}	3.9×10^{-5}	9.2×10^{-6}
16-BIT QUANT	8.5×10^{-6}	6.0×10^{-4}	1.0×10^{-4}
32-BIT QUANT	1.7×10^{-5}	1.2×10^{-3}	3.6×10^{-4}
POET-BIN	8.2×10^{-9}	5.4×10^{-9}	4.1×10^{-9}

TABLE – Implementation results

DATA SET	MNIST	CIFAR-10	SVHN
LATENCY(NS)	9.11	9.48	5.85
NUMBER OF LUTS	11899	9650	2660

- Tens of thousands of LUTs, cannot be handcoded in VHDL
- Python library to generate VHDL from high level network information
- 8-input LUTs for MNIST and CIFAR-10, 6-input LUTs for SVHN
- Need original implementation of the other works to estimate the resource consumption for the fully connected layers for fair comparison

Conclusion

Conclusion

- Proposed a Power-efficient Tiny Binary Neuron architecture
- Removed all MAC operations and memory access in classification layers
- Achieved comparable accuracies to other state-of-the-art works

Advantages

- Reduction in energy by up to three orders of magnitude when compared to recent binary quantized neural networks
- Can be implemented in any hardware, not just FPGAs

Further Work

- Implementation for the convolutional layers
- Results for larger datasets

Acknowledgments

We thank

- Ahmed Abdelsalam for his suggestions and comments throughout the project
- MITACS and ReSMiQ for partially sponsoring the project

Thanks !

Questions ? ? ?