Deep Learning Inference on Embedded Devices: Fixed-Point vs Posit

Seyed Hamed Fatemi Langroudi, Tej Pandit, Dhireesha Kudithipudi

Neuromorphic AI Lab
Department of Computer Engineering
Rochester Institute of Technology

EMC^2 Workshop, March 25, 2018

Introduction

Why Deep Learning Inference on Embedded Devices?

Most digital neuromorphic chips are specialized for data centers

Drawbacks of performing Deep Learning inference in data centers:

- ❖ **Latency**
- ❖ **Accessibility**
- ❖ **Security**

[3]

[2]

[1]

Introduction

What are the challenges in designing/performing Deep Learning architectures for embedded devices ?	❖ **real time performance** ❖ **Energy consumption**
Solutions for energy reduction ?	❖ **Low precision Arithmetic** ➢ **Fixed-point number system**
π **3.14159 26535** [4]	π **3.14** [4]

Introduction

Current Approaches	❖ **Most previous work requires** ➢ **Quantization techniques** ➢ **Retraining**
What is missing ?	❖ **Fixed-point number system represents numbers uniformly** ❖ **The parameters are distributed non-uniformly**
What is the solution ?	❖ **Using Posit Number System [5,6]** ➢ **Tapered number system [7]** ➢ **(non-uniform distribution)** ➢ **More accurate than floating point**

Posit Number System

- Proposed by John L. Gustafson, 2017 [5,6]

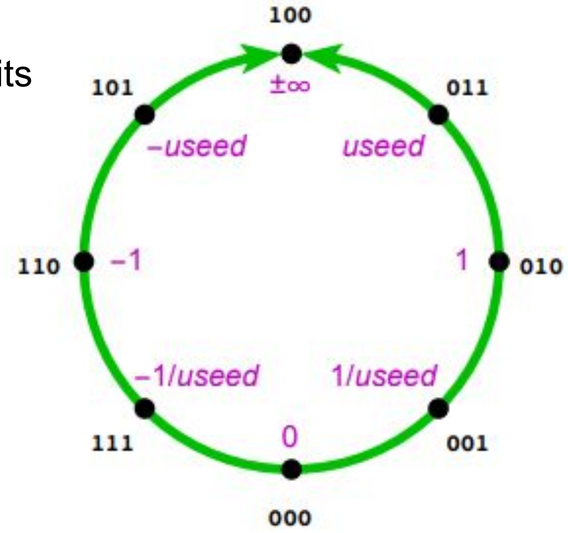
- Define : $p_{(n,es)}$

n= number of bits
es= number of exponent bits

$$useed = 2^{2^{es}}$$

$$min = useed^{-n+2}$$

$$max = useed^{n-2}$$



[6]

$$X = (-1)^{sign} \times (useed)^{r_{value}} \times 2^{exponent} \times (1 + fraction)$$

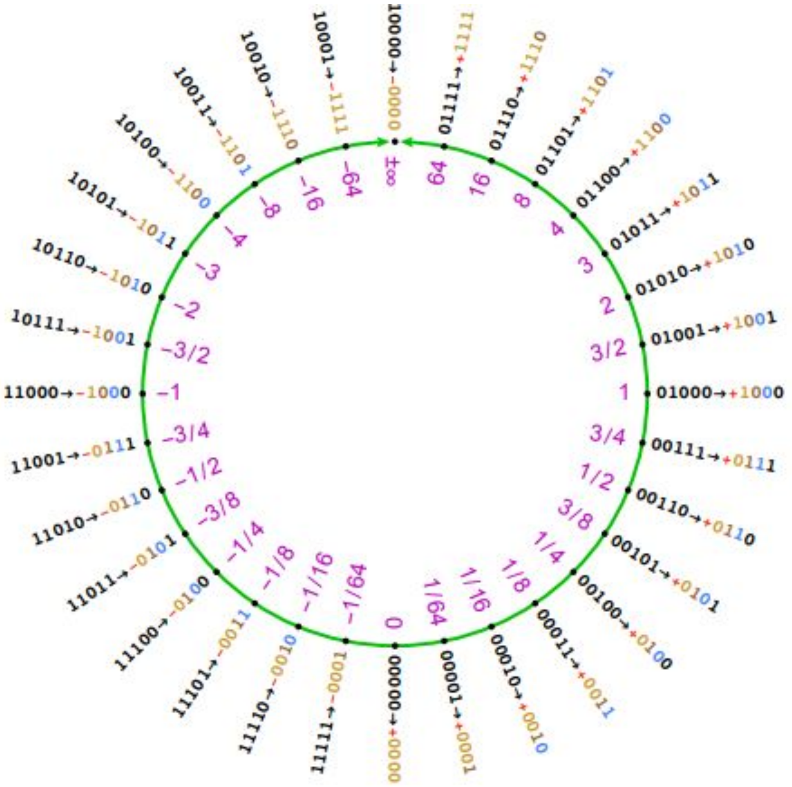
Posit Number System

Example: $P_{(5,1)}$

$$useed = 4$$

$$max = 64$$

$$min = 1/64$$



[6]

Posit Number System

Conversion from Posit to real number

- ❖ Sign
- ❖ r_{value} → Leading zero detection, Leading one detection
- ❖ Exponent value
- ❖ Fraction value

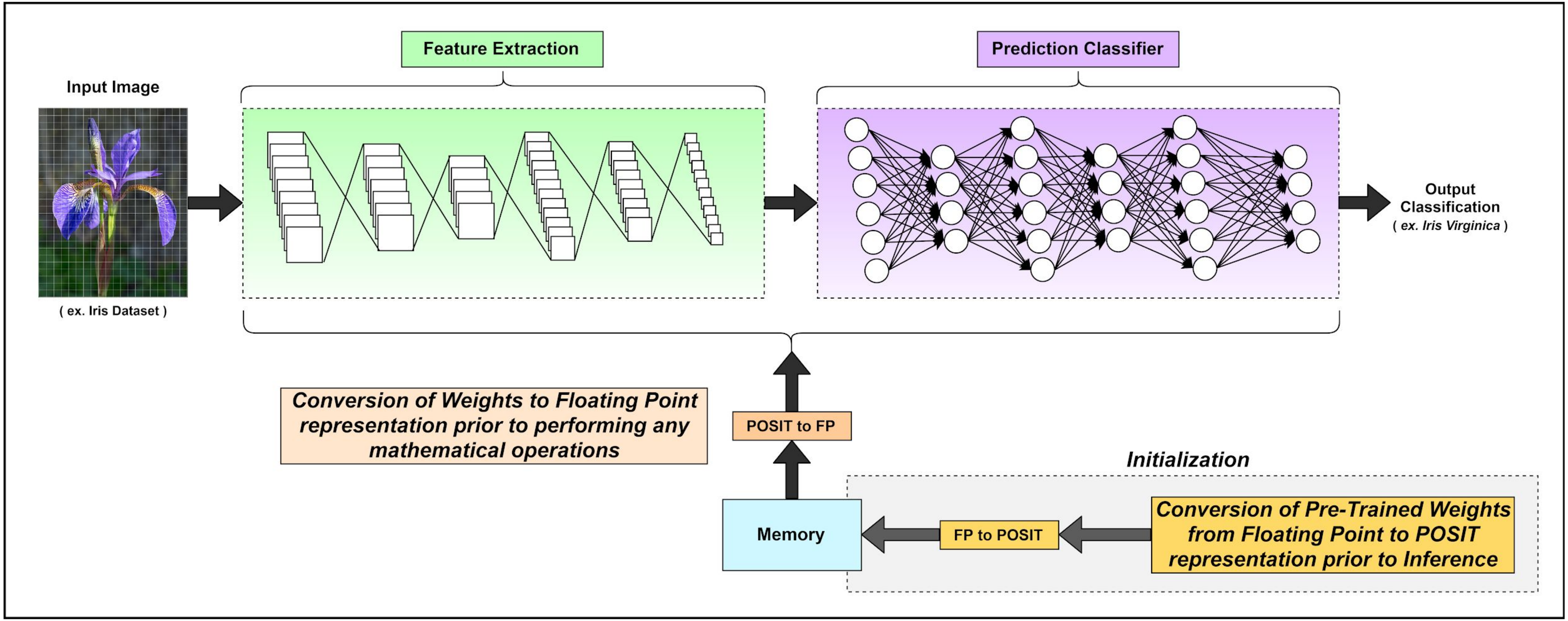
Conversion from real number to Posit

- ❖ r_{value}= Divide or multiply by 2 → [1,useed)
- ❖ exponent=Divide or multiply by 2 → [1,2)
- ❖ Fraction = rest of bits

$X_p =$	S	r	r'	e	f_{11}	f_{10}	f_9	f_8	f_7	f_6	f_5	f_4	f_3	f_2	f_1	f_0
$X_b =$	0	1	0	1	0	1	0	0	0	1	1	1	1	0	1	1

$$X_d = 4^0 \times 2^1 \times (1 + 0.280) = 2.56$$

Proposed Architecture



Experiments

Task:

- Handwritten numeral classification, Image classification

Parameters:

- Weights

Number Systems :

- Single Precision Floating Point Number System
- Variable Length Fixed-point Number System
  - Integer part = 1 bit
  - Fraction part = [0,15]
- Normalized Posit Number System , $p_{(i,0)}$ where i = [2,8]

Deep Neural Networks:

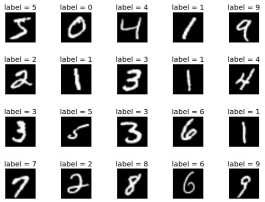
- LeNet-4 (2 Conv, 2 FC), ConvNet (3 Conv, 2 FC), AlexNet (5 Conv, 3 FC)

Metric:

- Top-1 accuracy, memory utilization, memory access

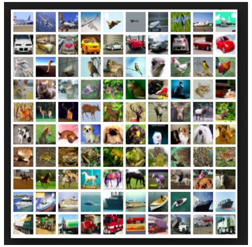
Datasets

- MNIST Dataset [9]
  - Categories = 10
  - Inference = 10000



[8]

- Cifar-10 Dataset [11]
  - Categories = 10
  - Inference = 10000



[10]

- Subset of ImageNet [13]
  - Categories =10
  - Inference = 10000

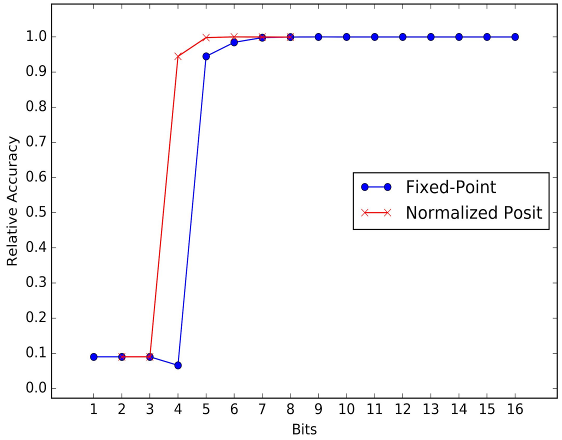
[12]

Baseline Results

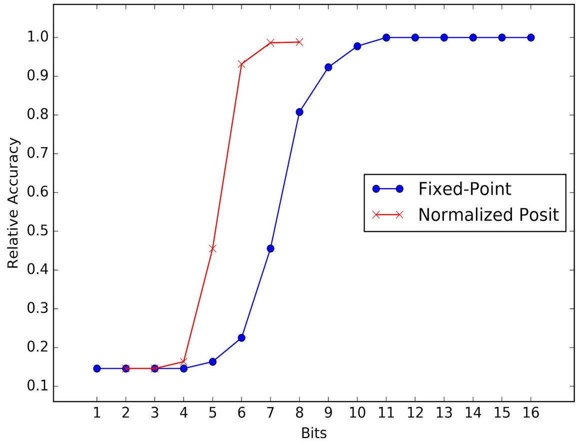
Single Precision Floating Point Number System:

Task	Dataset	# inference set	Network	Layer	Top-1 Accuracy
Digit Classification	MNIST	10000	Lenet	2 Conv and 2 FC	**99.03%**
Image Classification	CIFAR10	10000	Convnet	3 Conv and 2 FC	**68.45%**
Image Classification	ImageNet	10000	AlexNet	5 Conv and 3 FC	**55.45%**

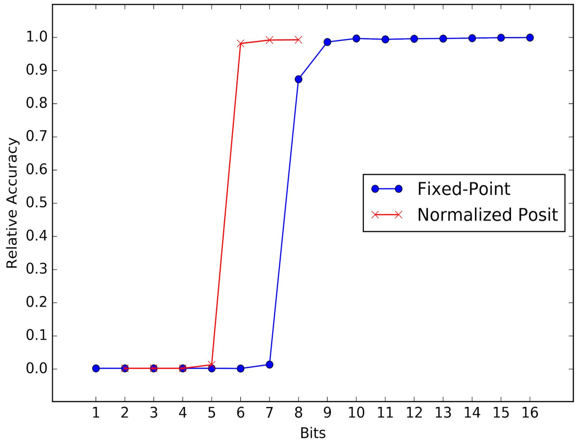
Results for proposed architecture on different datasets



MNIST



CIFAR-10



ImageNet

Summary of Results

Dataset	Network	# bits (FP)	# bits (FIX) 1% accuracy degradation	# bits (NP) 1% accuracy degradation	Memory utilization
MNIST	Lenet	32	7	5	**28.6%**
CIFAR10	Convnet	32	11	7	**36.4%**
ImageNet	AlexNet	32	9	7	**23%**

FP = Floating Point   FIX = Fixed-point   NP = Normalized posit

- It can also reduce the number of memory accesses through memory concatenation schemes.

Conclusions

- ❖ Exploring the use of Posit Number System in DNNs
  - ➢ Weights
  - ➢ 3 DCNNs and 3 Datasets
  - ➢ Posit outperformed the fixed-point implementations in terms of accuracy and memory utilization
  - ➢ We estimate that the use of Posit can help reduce the number of memory accesses

- ❖ Future work
  - ➢ Hardware implementation
  - ➢ Consideration of conversion overheads
  - ➢ Using the Posit number system for activation
  - ➢ Posit number system for other deep neural networks and Training Deep Learning Networks

Questions

References

1. Danny Shapiro, “Where Cars Are the Stars: NVIDIA AI-Powered Vehicles Dazzle at GTC Europe” https://blogs.nvidia.com/blog/2017/10/12/electric-autonomous-vehicles-gtc-europe/, 2017
2. https://fedotov.co/watch-amazon-first-package-delivered-drone/
3. http://www.aberdeenessentials.com/techpro-essentials/an-iot-deviceto-protect-your-iot-devices/
4. https://phys.org/news/2010-08-japanese-whizzes-news-pi-.html
5. Gustafson, John L., and Isaac T. Yonemoto. "Beating Floating Point at its Own Game: Posit Arithmetic." *Supercomputing Frontiers and Innovations* 4.2 (2017): 71-86.
6. John L. Gustafson, “Posit Arithmetic”, Oct. 2017, https://posithub.org/docs/Posits4.pdf
7. Morris, Robert. "Tapered floating point: A new floating-point representation." *IEEE Transactions on Computers* 100.12 (1971): 1578-1579.
8. http://corochann.com/mnist-dataset-introduction-1138.html
9. LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324.
10. https://www.cs.toronto.edu/~kriz/cifar.html
11. Krizhevsky, Alex, and Geoffrey Hinton. "Learning multiple layers of features from tiny images." (2009).
12. “ImageNet Data Set” http://vision.stanford.edu/resources_links.html
13. Russakovsky, Olga, et al. "Imagenet large scale visual recognition challenge." International Journal of Computer Vision 115.3 (2015): 211-252.