High Speed Training Using Binary Neural Networks

Daniel Chen, Thomas Forzani, Daniel Maevsky, Sachin Mathew, Serena Zhang
Advisor: Dr. Richard Martin
2021 WINLAB Summer Internship, Rutgers University

RUTGERS
WINLAB | Wireless Information Network Laboratory

INTRODUCTION

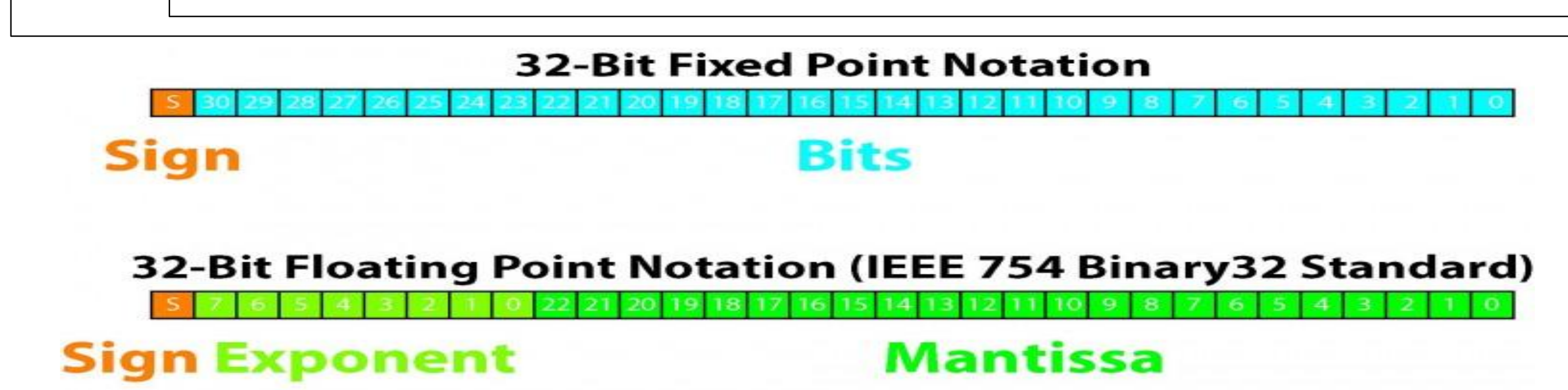
- Traditionally, power consumption has been an oft overlooked metric in the training and execution of neural networks, but the paradigm is beginning to shift as large computing systems used in deep learning continue to increase in scale.
- Integer units take up far less physical chip space than floating point units, and their power consumption is far less as a result.
- We investigated the use of integers and binary fixed-point number implementations in two neural networks trained on the MNIST-digits and MNIST-fashion datasets to see what effect their use might have on accuracy and training time.

OBJECTIVES

- **Energy:** Integer processing units use up to 60% less physical chip space on a CPU than float processing units. Since fixed-point numbers can be created using integers and bit-wise operations, fixed-point unlocks this potential of greater energy efficiency.
- **Accuracy:** Establishing and maintaining comparable accuracy while transitioning to using fixed-point numbers is vital before this method can be implemented at scale.
- **Speed:** Our fixed-point implementations of neural networks should require the same training time, allowing for power-saving benefits with next to no drawbacks.

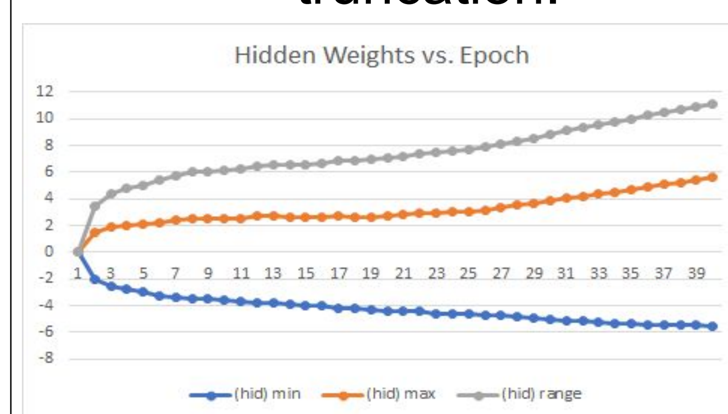
FIXED VS FLOATING POINT

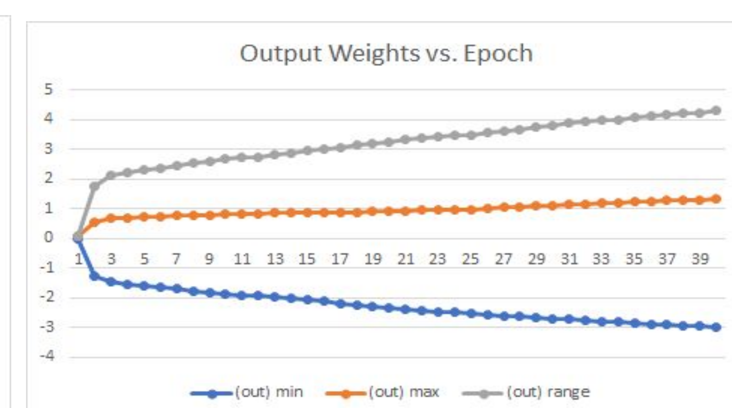
- **Floating Point:** Consists of sign, exponent, and mantissa making it similar to scientific notation
  - Current standard for ML because of large range and variable precision
  - ALU intensive for most arithmetic operations
- **Fixed Point:** Consists of sign and bits with implied point fixed between some two predifined bits.
  - Limited range, constant precision
  - ALU nonintensive, uses modified integer math
- Generated Fixed Point matrix mathematics library of helpers for training,/prediction

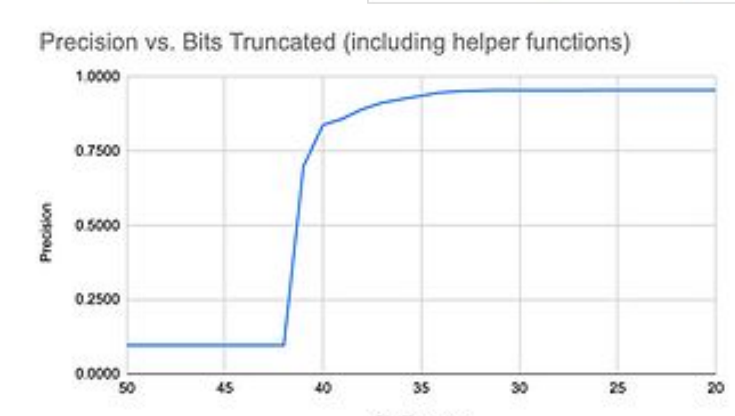


REPRESENTATION BOUNDS

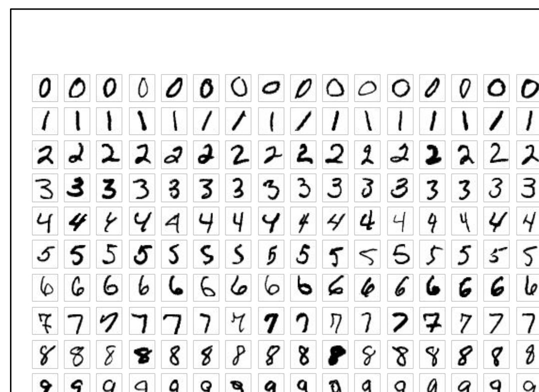
- **Fixed Point Parameters:** Range, Precision
  - **Range:** Determined by magnitude of representation (number of bits before point)
    - Find dynamic range by periodically finding maxima and minima of weight matrices in floating point model during training
  - **Precision:** Determined by minimum difference in representation (number of bit following point)
    - Find precision by finding accuracy drop of fixed point model at varying numbers of bit truncation.







TRAINING ON DATASETS



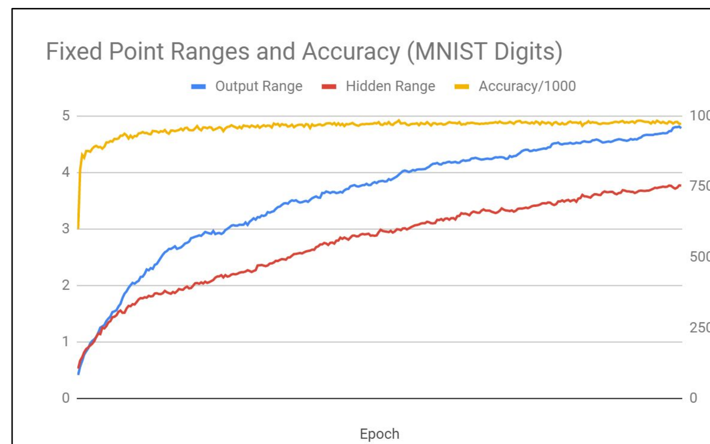
MNIST Digits

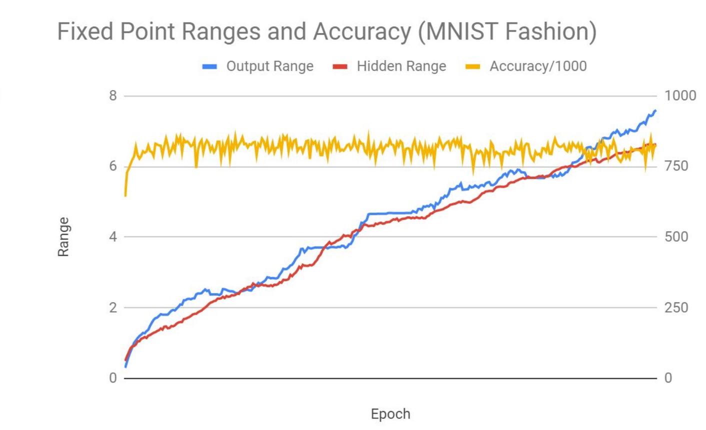
- Handwritten digits 0-9
- 28x28 grayscale image
- Easy to incorporate & train
- Highly Implemented with near perfect accuracy

MNIST Fashion

- Articles of clothing Ex. Sneakers, shirts, dresses, etc.
- 28x28 grayscale image
- Easy to incorporate & difficult to train
- More applicable for CV tasks

RESULTS





- **Final Fixed Point Representation:** Sign bit, 15 bits preceding point, 48 following
  - **Range:** About -2^{16} to $+2^{16}$
  - **Precision:** About 2^{-48}
- **Model:** Same model used for both MNIST-digits and MNIST-fashion
  - 3 Layers: Input - 784 nodes (1 per pixel), Hidden - 100 nodes, Output - 10 nodes
- **Accuracy in Models:** Accuracy was collected after 6 epochs of training over the the training set
  - **Final Accuracy for MNIST-digits in fixed point model** is 97.72%, a 0% reduction in accuracy compared to the floating point model
  - **Final Accuracy fo MNIST-fashion in fixed point model** is 80.00 %, a 3% reduction in accuracy compared to the floating point model

ACKNOWLEDGEMENTS

We'd like to thank Professor Richard Martin for being our mentor throughout this project. Likewise we'd like to thank the entire faculty at WINLAB for their help and know how in bringing this project to fruition.

REFERENCES

[1] E. Garcia-Martin, C. F. Rodrigues, G. Riley, and H. Grahn, "Estimation of energy consumption in machine learning," *Journal of Parallel and Distributed Computing*, vol.134, pp. 75-88, 2019, doi:10.1016/j.jpdc.2019.07.007.
[2] *Intel and Floating-Point*, Intel, n.a.
[3] W. Ho, K. Chong, B. Gwee and J. S. Chang, "Low power sub-threshold asynchronous QDI Static Logic Transistor-level Implementation (SLTI) 32-bit ALU," 2013 IEEE International Symposium on Circuits and Systems (ISCAS), 2013, pp. 353-356, doi: 10.1109/ISCAS.2013.6571853.

WINLAB