Quantization of Neural Network

Aidan Yap (AidanYap@gmail.com)
Analog Mixed Signal IC Lab
Harvard Westlake School, Class of 2022
USC Viterbi | Department of Electrical and Computer Engineering, SHINE 2021

Introduction

In recent years, neural networks take on increased significance because they can be applied to current and emerging technologies. For example, Artificial Intelligence (AI) systems rely on neural networks to achieve high accuracy in applications ranging from robotics, speech recognition, image recognition, semantic parsing. etc.

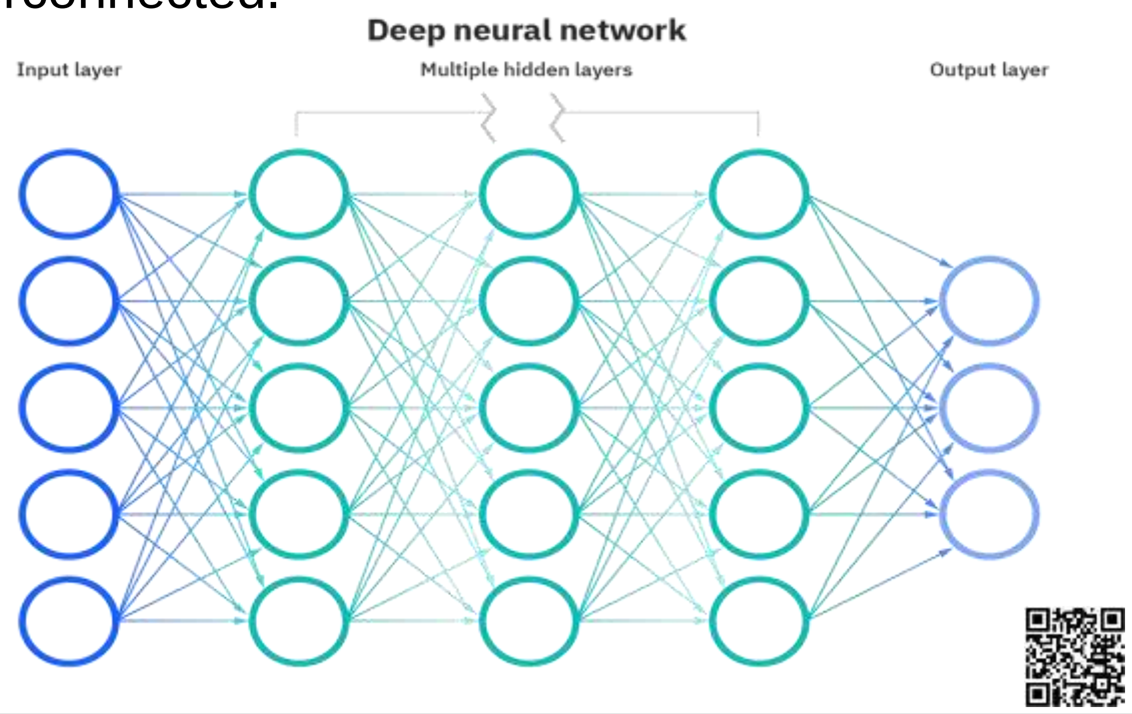
Dr. Chen's Analog Mixed Signal IC Group conducts research to create the most energy efficient neural network systems by finding improvements in both hardware and software, that would balance the accuracy of the network against complexity of the system, which impacts energy usage.

This research is important because energy is a finite and costly resource. Energy efficient circuits and system architectures are good for our planet. At the same time, the more accurate the neural network, the better the network.

For my SHINE project, I worked with my mentor to train a neural network model using Python and run simulations on the model to look for an optimal balance between accuracy of the network and energy consumption.

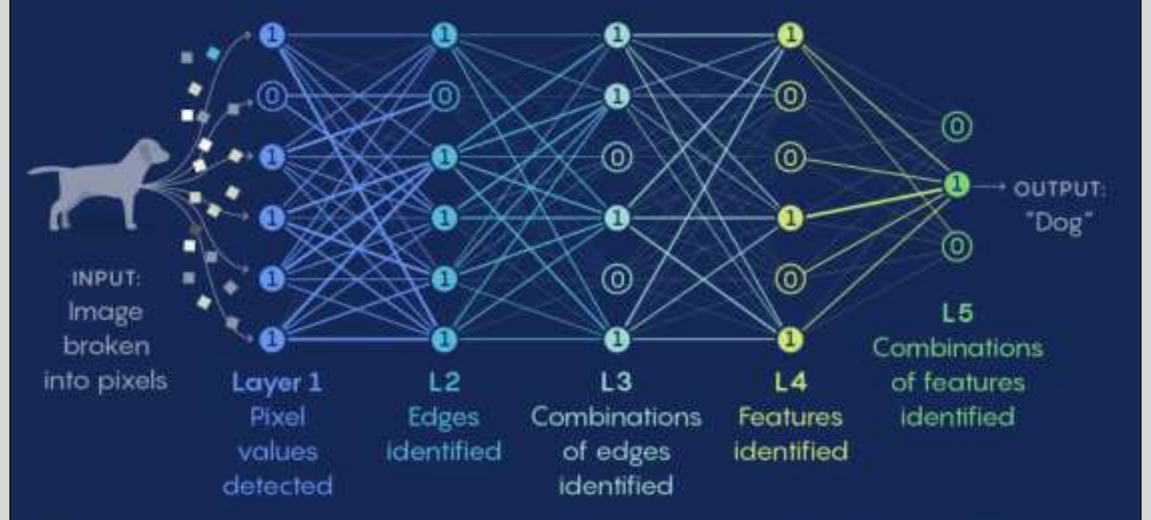
What is a Neural Network

A neural net is modeled loosely on a human brain and consists of thousands or even millions of simple processing nodes that are densely interconnected.



How Neural Network Works

Neural networks pass an input, like an image, through multiple layers of digital neurons. Each layer reveals additional features of the input. Mathematicians are revealing how a network's architecture — how many neurons and layers it has and how they're connected — determines the kinds of tasks that the neural network will be good at.



When data is fed into a network, each artificial neuron that fires (labeled "1") transmits signals to certain neurons in the next layer, which are likely to fire if multiple signals are received. This process reveals abstract information about the input.

A weight is a learnable parameter inside the network. It represents the strength of the connection between two nodes. Therefore, weights collectively represent the relative strength of the different connections between nodes after model training. This can be likened to a human brain that has learned, for example, how to speak Spanish or do carry addition. For more information, please check out my video presentation here:

How to Reduce Cost

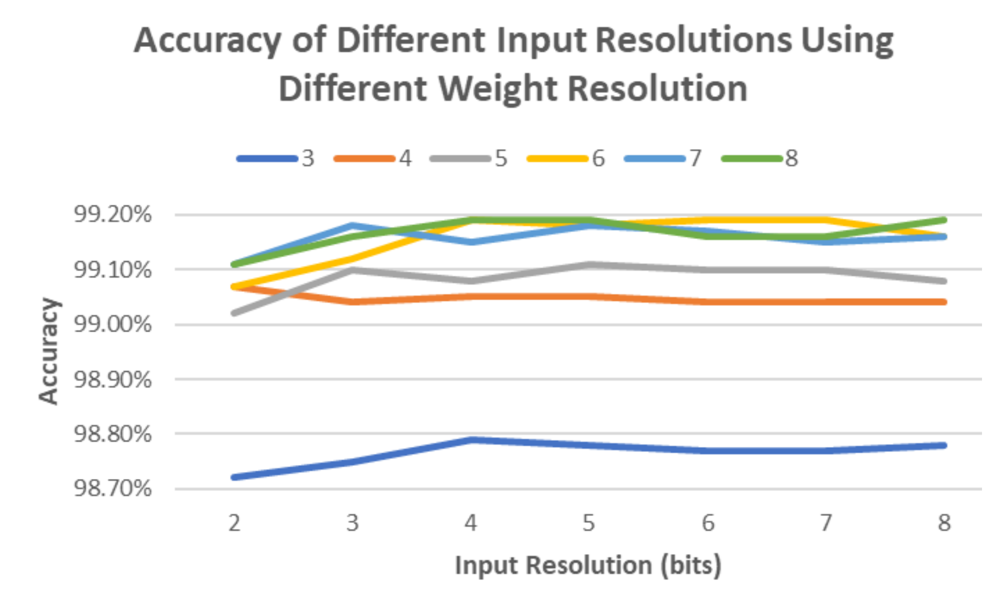
Neural networks incur significant computational costs because they are resource intensive algorithms. Reduced-precision computation can be used to reduce memory bandwidth demand and increase power-efficiency for neural networks. For example, instead of using 32-bit floating point numbers, researchers can use 16, 8, or fewer bits of precision. This is possible because neural networks are tolerant of reduced precision. This results in power saving because computation using lower precision number generally consumes lesser energy.

Results

		Weight Resolution						
		2	3	4	5	6	7	8
Input Resolution (bits)	2	74.50%	98.72%	99.07%	99.02%	99.07%	99.11%	99.11%
	3	75.80%	98.75%	99.04%	99.10%	99.12%	99.18%	99.16%
	4	75.70%	98.79%	99.05%	99.08%	99.19%	99.15%	99.19%
	5	75.80%	98.78%	99.05%	99.11%	99.18%	99.18%	99.19%
	6	76.00%	98.77%	99.04%	99.10%	99.19%	99.17%	99.16%
	7	76.10%	98.77%	99.04%	99.10%	99.19%	99.15%	99.16%
	8	76.10%	98.78%	99.04%	99.08%	99.16%	99.16%	99.19%

A summary of our simulation results is listed in the table (right). It is clear that weight = 2 should not be used because regardless of the input resolution, the accuracy of the network hovers between 74.5% to 76.10% well below the 98%+ for weights > 3.The line graph of the weights 3 to 8.

From the line graph, is it obvious that weights > 4 should be used and input resolutions between 2 through 8 produce very small difference in accuracy (≈ 0.5%). Therefore, in terms of efficiency, an input resolution of 2 is a good compromise between accuracy and cost.



Simulation Setup

We used Python to set up a model of a quantized neural network. Then, we performed iterative simulations of the model using different parameters that affect accuracy of the network (such as number of bits) and simulated for the accuracy of MNIST test dataset.

Our Python code can be found here:

Skills Learned

I learned to code in Python. I had limited coding experience before this program and am glad that SHINE gave me the opportunity to learn how to code. Coding is an essential skill set, especially in STEM, and Python is a good language to learn because it is widely used in the technology industry.

I also learned about dealing with trade-offs in engineering designs. Trade-off strategies are an important part of engineering designs, but this decision-making skill is better learned in research opportunities provided in SHINE than in a classroom.

My STEM Coursework

I learned about neural networks and its applications to many fields in STEM, including robotics. My robotics team 62B (https://hw-robotics.web.app/#team_info) will benefit from what I had learned about neural networks. It will help us to navigate our robot better and build AI functionalities into our robot as we look to defend our VEX Robotics World Championship title next year.
I will also benefit from learning how to code as I further my education in STEM.

Acknowledgements

I would like to thank Dr. Chen, my mentor, Rezwan A. Rasul, for all his help and patience and the SHINE team for putting together an amazing program and being supportive every step along the way.