RUTGERS

Field Programmable Gate Arrays for Machine Learning Acceleration

WINLAB

Group Members

Michael Yakubov
Rutgers ECE 2022

Milos Seskar
Rutgers ECE 2022

Background information:

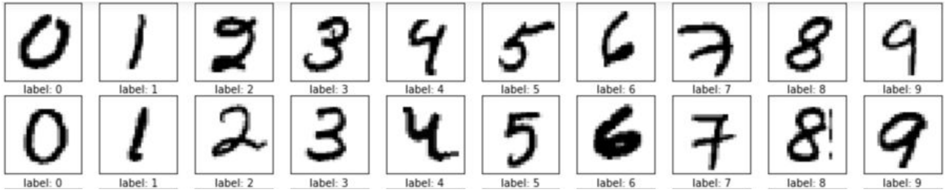
- **Field Programmable Gate Array (FPGA)** : hardware component that allows for a customisable circuit implementation (re-programmable)
- **Machine Learning (ML):** The study of computer algorithms that automatically improve performance through “training” and experience.
- **Neural Network (NN):** “web of nodes” structure used for ML, designed to improve classification accuracy by training with datasets → similar to human nervous system
- **Central Processing Unit (CPU):** hardware component responsible for processing and executing instructions (the brains of a computer)
- **Graphics Processing Unit (GPU):** hardware component designed to rapidly process instructions in parallel

Motivations

- Typically machine learning is performed using central processing unit (CPU) or graphics processing unit (GPU)
- Evaluate performance of FPGA-based Machine Learning (ML) accelerators when used for real-time inference and/or signal processing.
  - Application Specific Integrated Circuits (ASICS) are the best hardware choice, but costly and difficult to make
  - FPGAs are reprogrammable, therefore they are multipurpose!

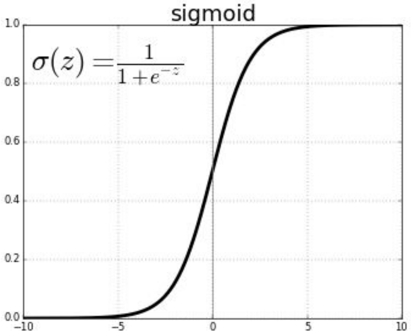
Methodology

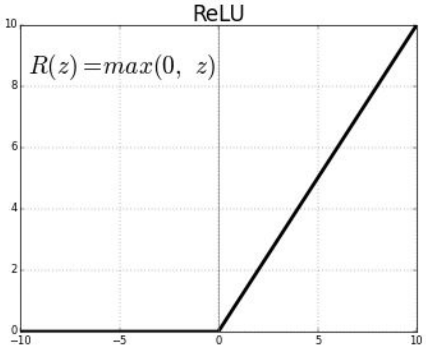
- Began with a perceptron model to classify sonar data.
  - Written in Python and deployed on local laptop CPU
- A simple 3-layer neural network (NN) written in Go
  - Moved to handwritten digit classifier (MNIST dataset)



Challenges faced

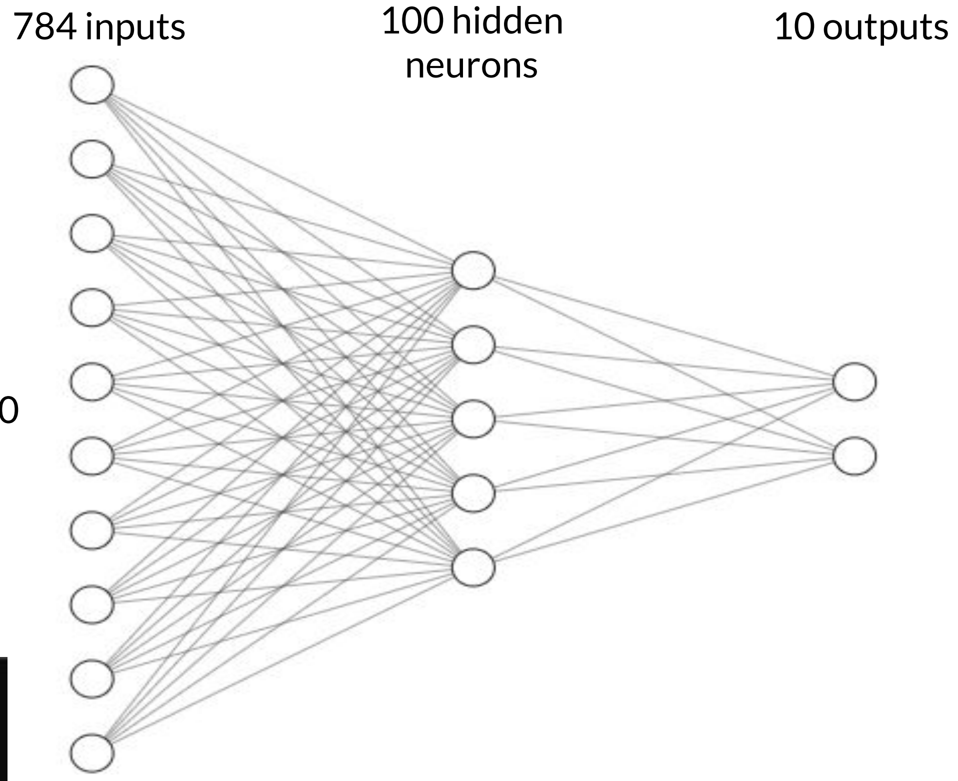
- Integer arithmetic is preferable on an FPGA
- Switched our NN to use Rectified Linear Unit (ReLu) instead of Sigmoid activation functions
    - Biggest problems: “Dying ReLUs” and “holes” for weights when using gradient descent → getting lost in the woods so to speak

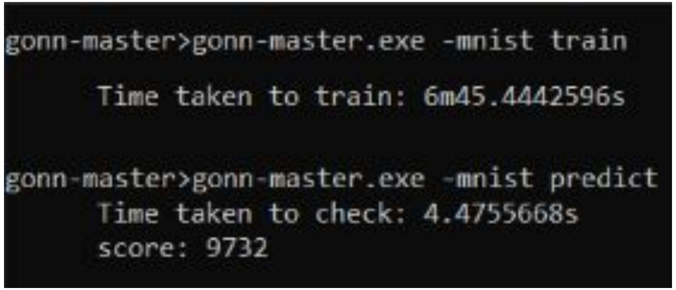




Results

- Sweetspot:
  - Learning rate: 0.0005
  - Scaling inputs to be from 0 to 10 (grayscale pixel values)





```
gonn-master>gonn-master.exe -mnist train

      Time taken to train: 6m45.4442596s

gonn-master>gonn-master.exe -mnist predict
      Time taken to check: 4.4755668s
      score: 9732
```

The Future?

- Implement Fixed Point Arithmetic (FiPA) or scale values to achieve independence from Floating Point Arithmetic (FPA)
- Preprocess our input data (pixel data from images) on the CPU, deploy the NN on FPGA, compare time
  - Test on Rutgers "iLab" machines to see how GPU performs for comparisons to CPU and FPGA
- Move to more complicated classifiers (colored images)

A special thank you to our mentors: Richard Martin, Jennifer Shane, and Prasanthi Maddala!

Questions?