Deep neural network with limited precision

Chetan Gupta
A20378854

Index

Problem Statement

- Training of large-scale deep neural networks is often constrained by the available computational resources. We are going study the effect of limited precision data representation and computation on neural network training and testing over GPU. Within the context of low precision floating-point computations, we will going to observe the rounding scheme to play a crucial role in determining the network's behavior during training and its testing.

Introduction of Deep Learning

- Deep learning (also known as deep structured learning or hierarchical learning) is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms. Learning can be supervised, partially supervised or unsupervised. The training of deep neural networks is very often limited by hardware. A neural network is a very powerful machine learning mechanism which basically mimics how a human brain learns.
- In Deep learning we create multiple layer of nodes,

  (1) take some data,

  (2) train a model on that data, and

  (3) use the trained model to make predictions on new data

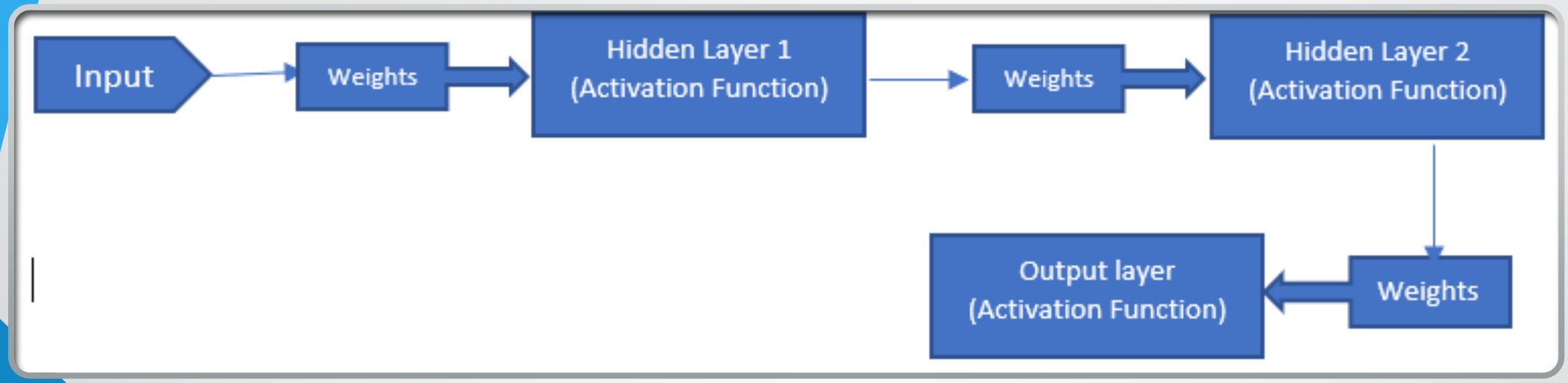
Introduction about GPUs

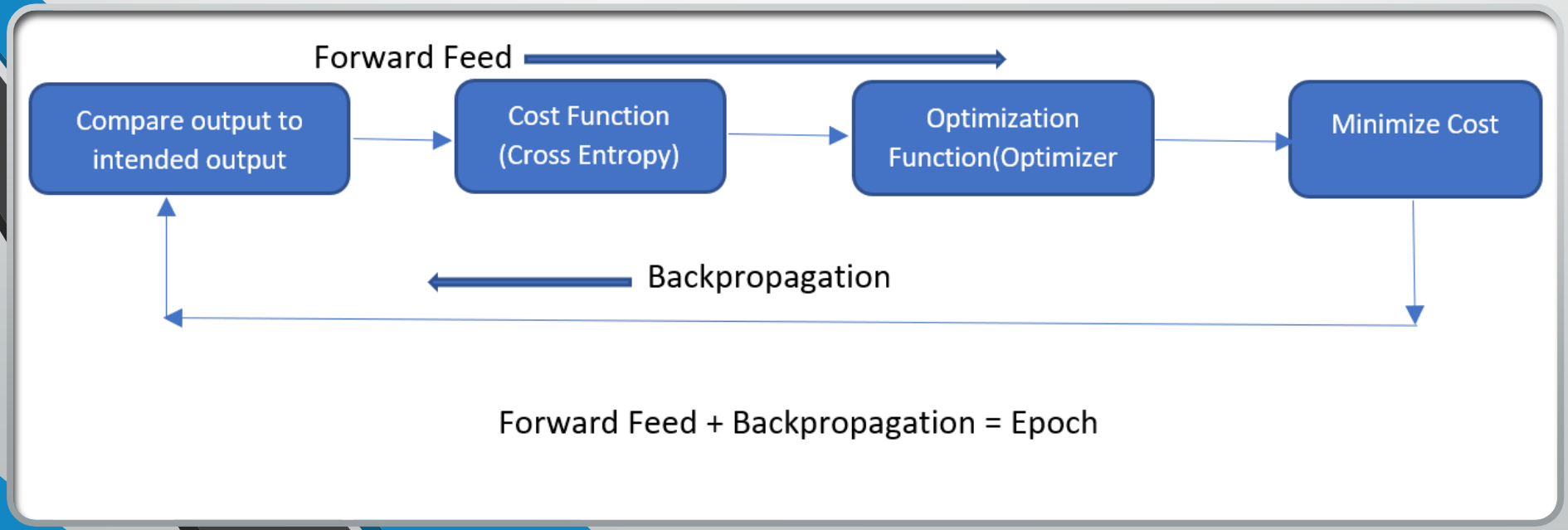
- GPUs is very efficient for processing neural network as compare to CPUs because they have hundreds of simpler cores, thousands of hardware concurrent threads and have maximize floating point throughput. GPUs was proven better option for Deep learning neural networks
- GPUs was proven better option for Deep learning neural networks.
- Just for the example, Google has manufactured extremely powerful system to do their processing, which they had specially built for training huge nets. This system was monstrous and was of $5 billion total cost, with multiple clusters of CPUs. After that researchers at Stanford built the same system in terms of computation to train their deep nets using GPU. And guess what; they reduced the costs to just $33K ! This system was built using GPUs, and it gave the same processing power as Google's system

	Google	Stanford
Number of cores	1K CPUs = 16K cores	3GPUs = 18K cores
Cost	$5B	$33K
Training time	week	week

We can see that GPUs rule. But what exactly is the difference between a CPU and a GPU?

Working of NN Model





Functionality used in neural networks

- Rectified Linear Unit (ReLU)
  - An activation function with the following rules:
    - If input is negative or zero, output is 0.
    - If input is positive, output is equal to input.
- Softmax function

A function that provides probabilities for each possible class in a multi-class classification model. The probabilities add up to exactly 1.0. For example, softmax might determine that the probability of a particular image being a 0 at 0.9, a 1 at 0.08, and a 2 at 0.02.

- Adam optimizing function

Adam is an optimization algorithm that can used instead of the classical stochastic gradient descent procedure to update network weights iterative based in training data.

System Specs

Intel i7 7500u which is having 2 cores at 2.70 GHz

Nvidia GPU Tesla P100 chipset is GP100 having 3584 cores per GPU with RAM of 16 GiB GDDR5

MNIST Dataset

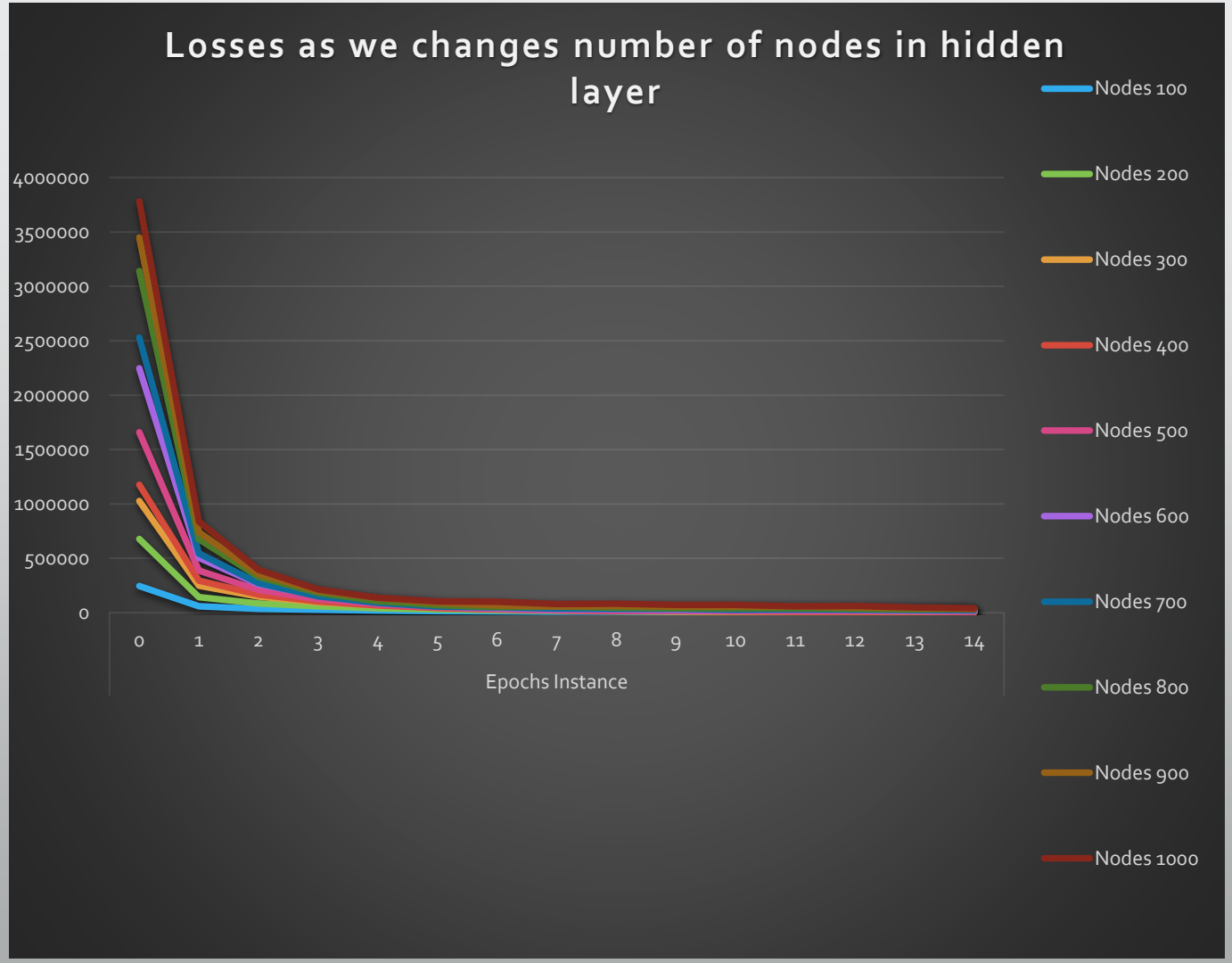
Python

- TensorFlow
- CUDA
- CUDNN

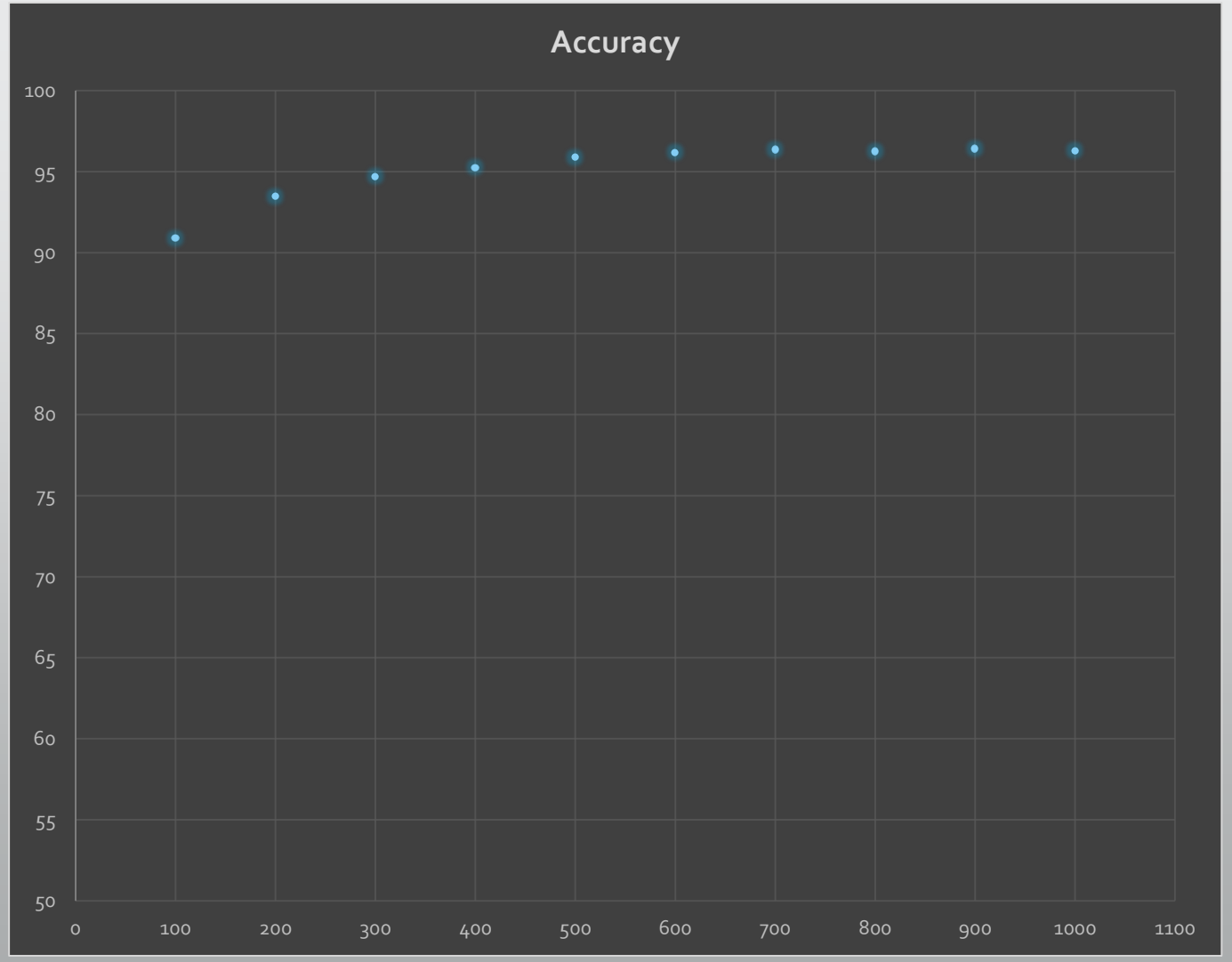
MNIST

- MNIST database of handwritten digits which has a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image. The most commonly used sanity check. Dataset of 28x28, centered, B&W handwritten digits.

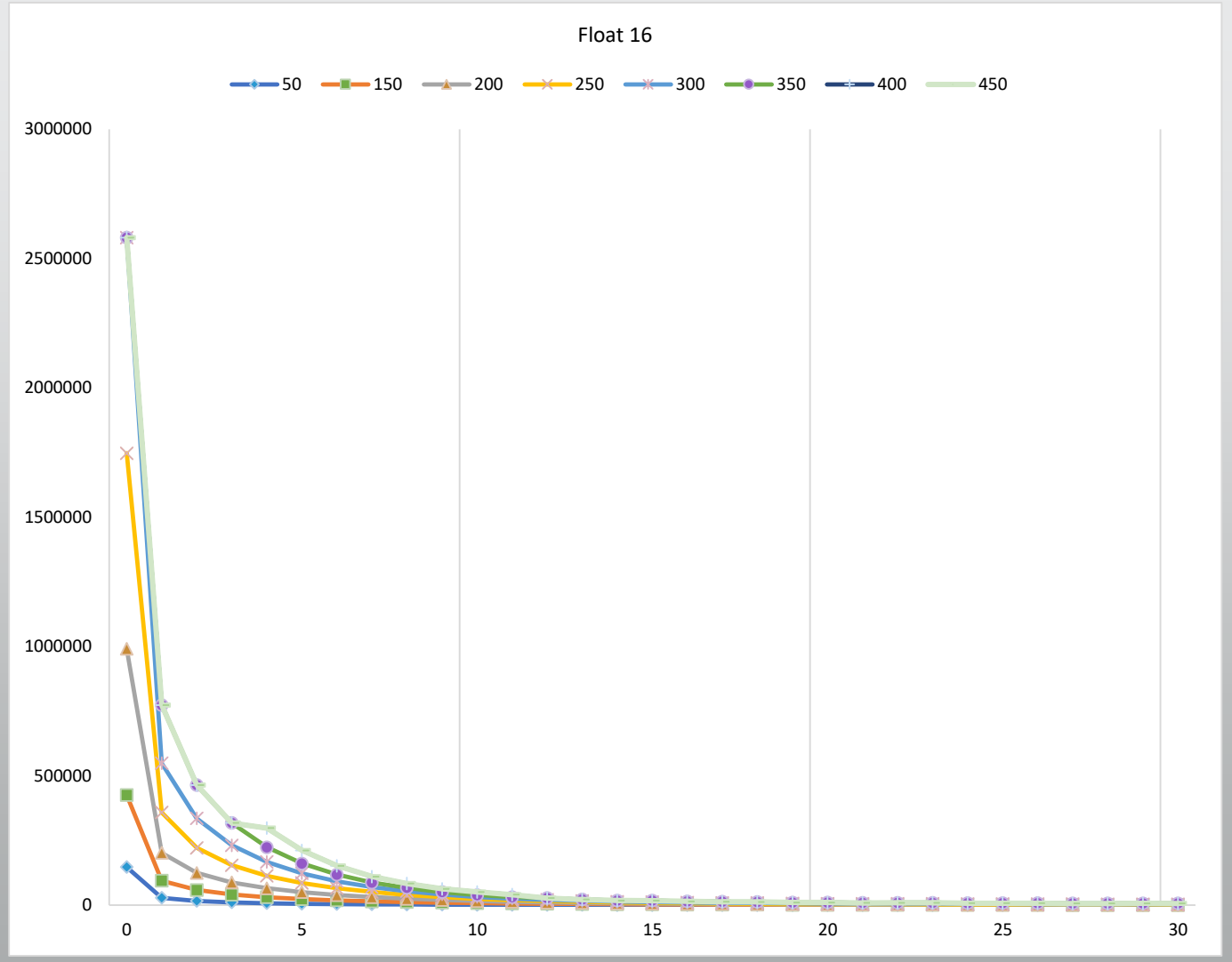
Float 32 error over CPU



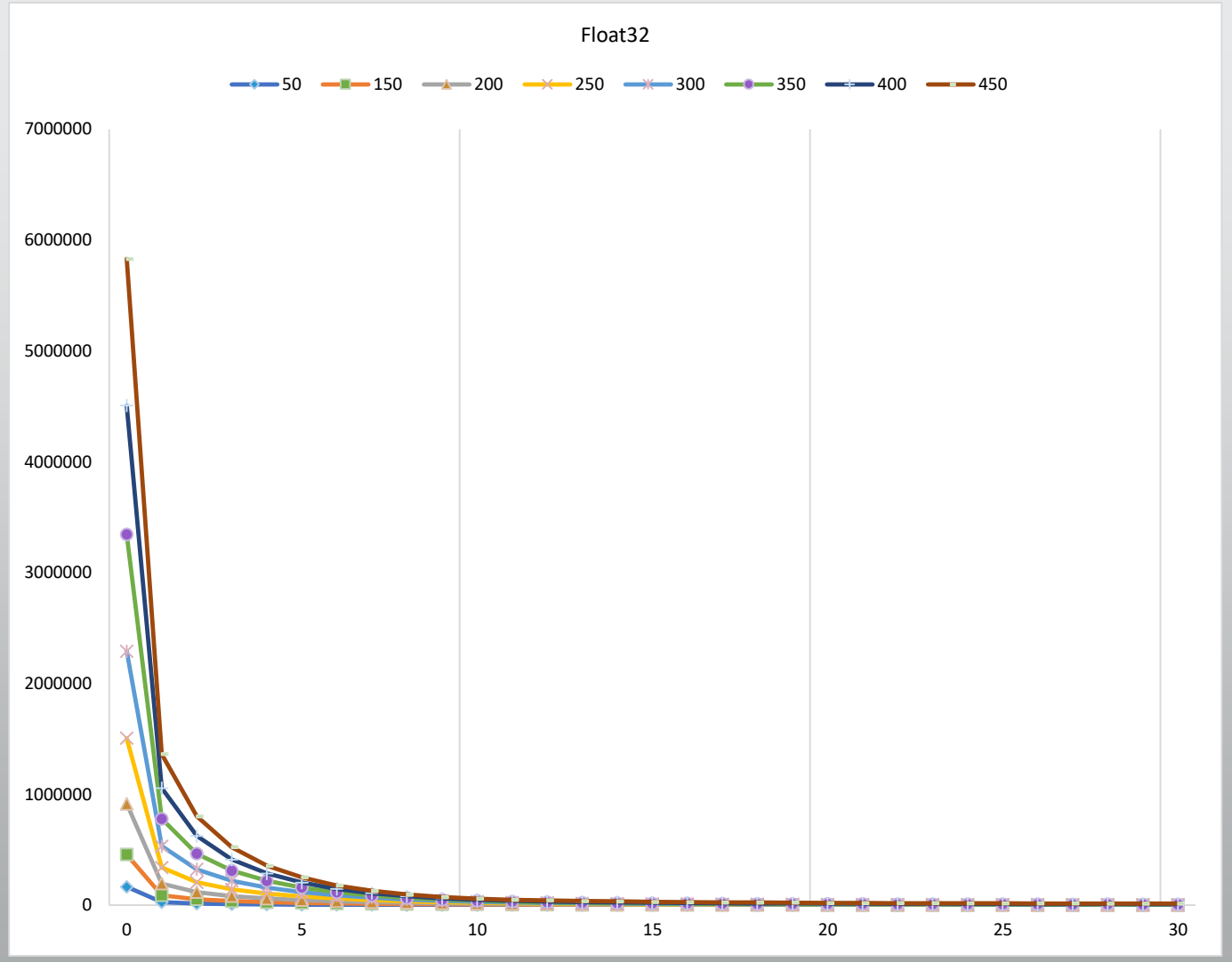
Accuracy Rate over CPU at Float32



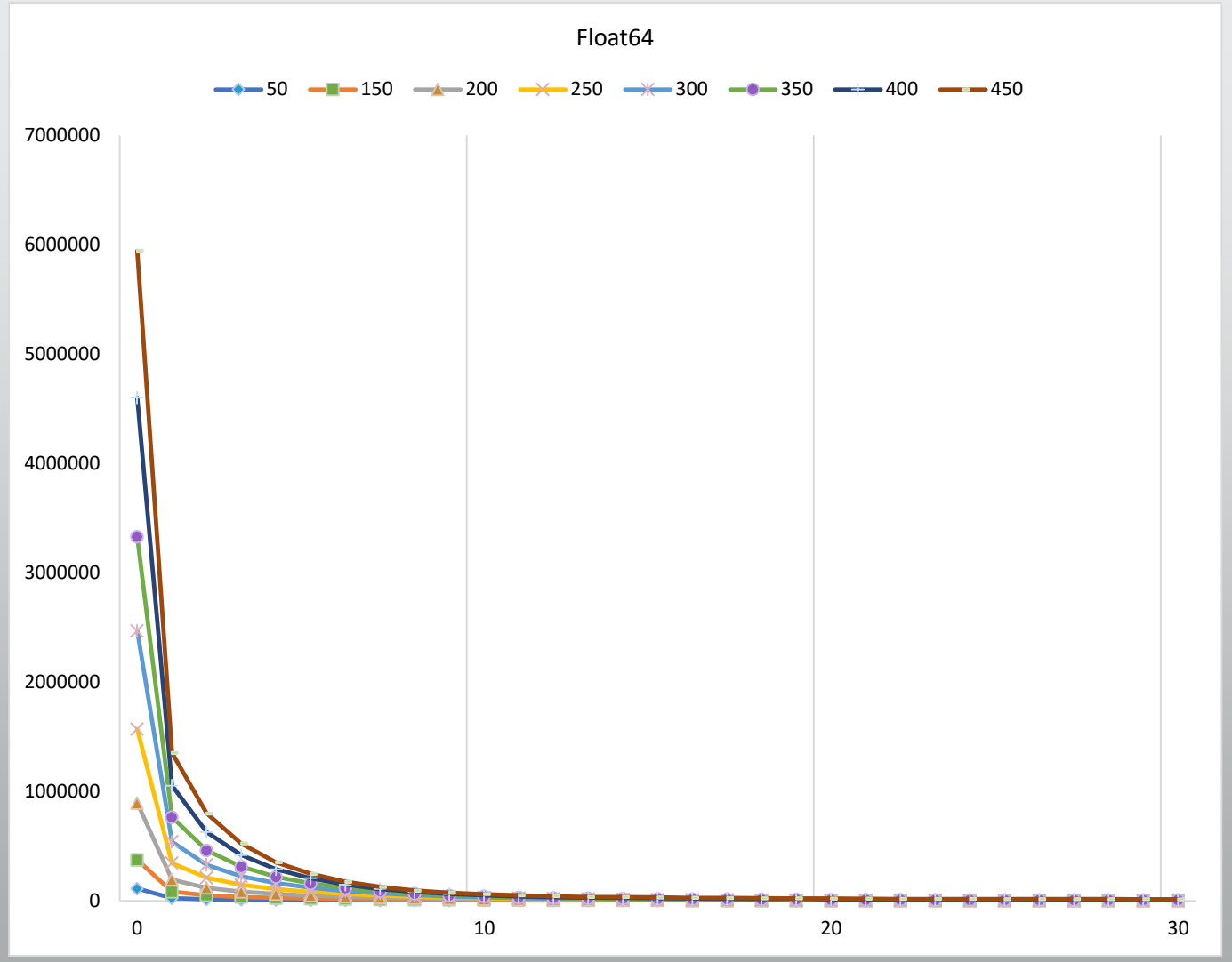
Float 16 error over GPU



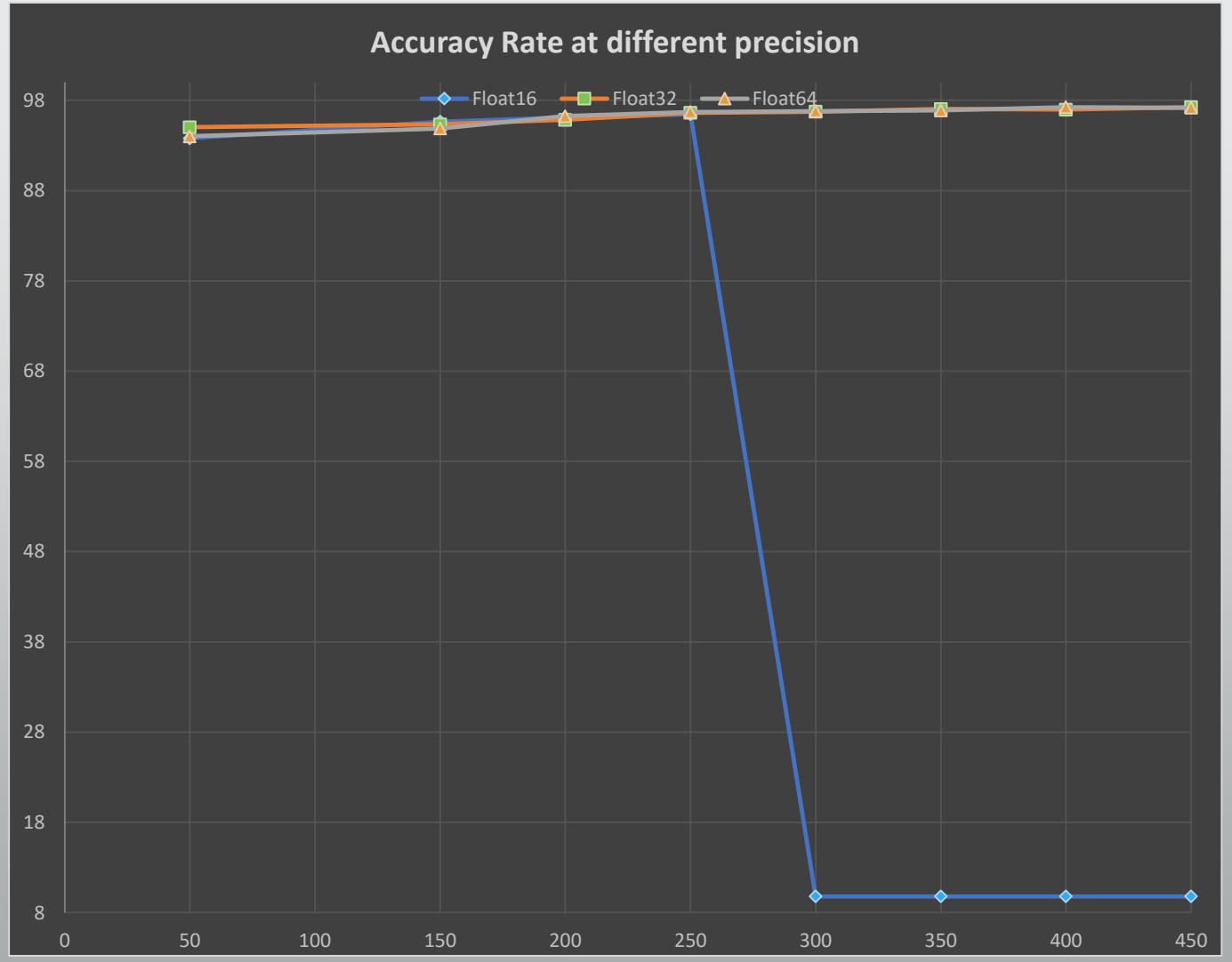
Float 32 error over GPU



Float 64 error over GPU



Accuracy rate at different precision by varying number of nodes of internal NN



GPU usage

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 375.26                 Driver Version: 375.26                    |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla P100-PCIE...  Off  | 0000:03:00.0     Off |                    0 |
| N/A   27C    P0    44W / 250W |  16126MiB / 16276MiB |     58%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla P100-PCIE...  Off  | 0000:82:00.0     Off |                    0 |
| N/A   23C    P0    30W / 250W |  16052MiB / 16276MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID  Type  Process name                               Usage      |
|=============================================================================|
|    0     92488    C   python                                      15549MiB |
|    0     93210    C   python                                        575MiB |
|    1     92488    C   python                                      15475MiB |
|    1     93210    C   python                                        575MiB |
+-----------------------------------------------------------------------------+
```

Pending work/ Conclusion

- I am planning for running this model by varying float8
- Planning to run and analyze the model for one more datasets
- Need performance matrix as per time and memory taken by GPU in processing whole request.
- Our results show that deep networks can be trained using only 16-bit wide fixed-point number representation when using typecasting the float values at 16, 32 and 64, and incur little to no degradation in the classification accuracy.

Thank you.