Communication-Efficient Learning of Deep Networks from Decentralized Data

Code Reimplementation and Extension Using Tensorflow

Georgy Marrero, Ashwinee Panda - CS294-131: Special Topics in Deep Learning

Motivation

Private data critical in deep learning

Exponentially increasing frequency of data breaches on centralized servers

Data transfer expensive (bandwidth)

Unused computing power of laptops, phones with neural chips

Problem Statement

Data distributed non-iid across devices

Low device liveness/data availability

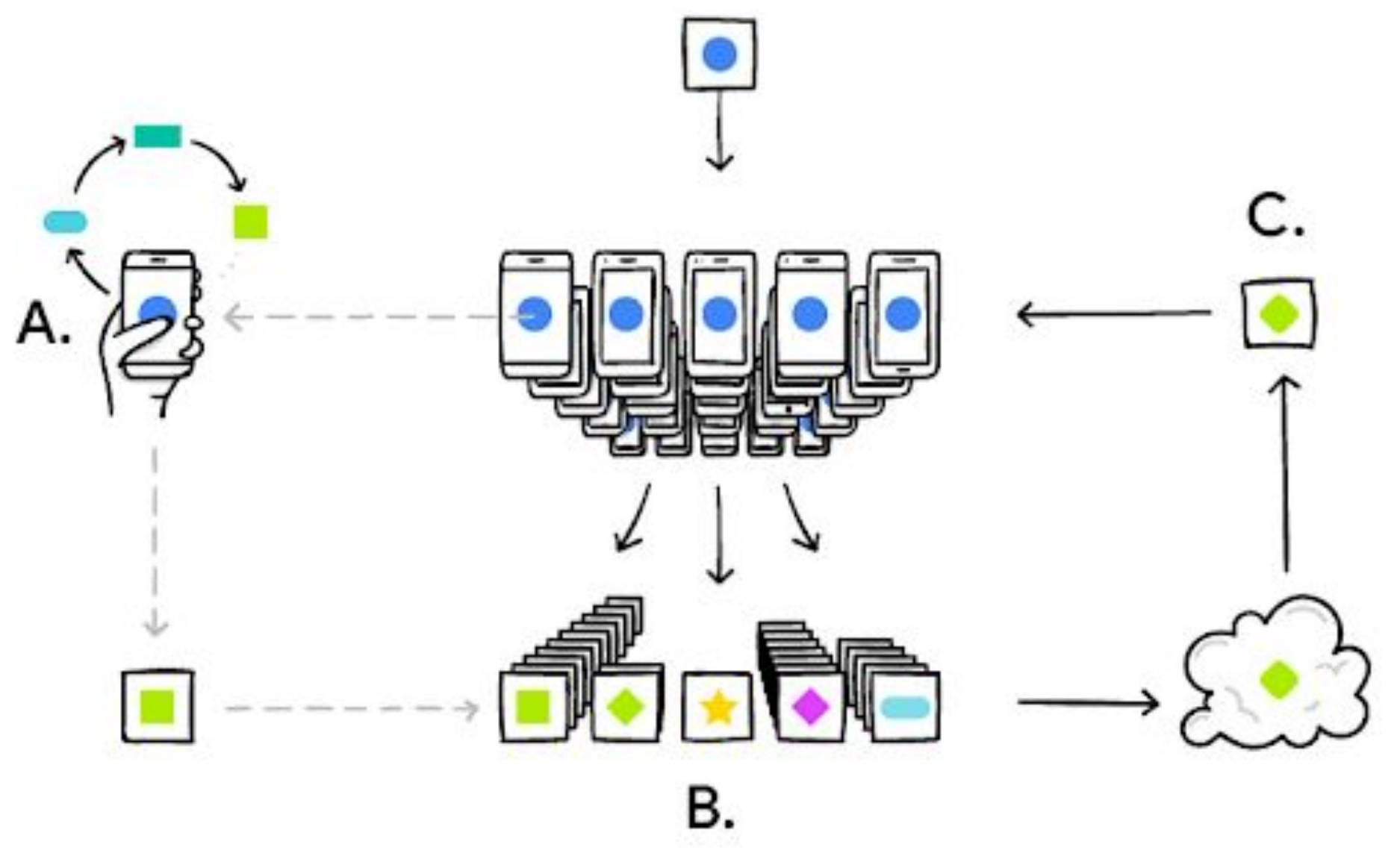
Huge latency issues in naive approach

Need interoperable solution that works across most deep neural networks

Approach: Federated Learning

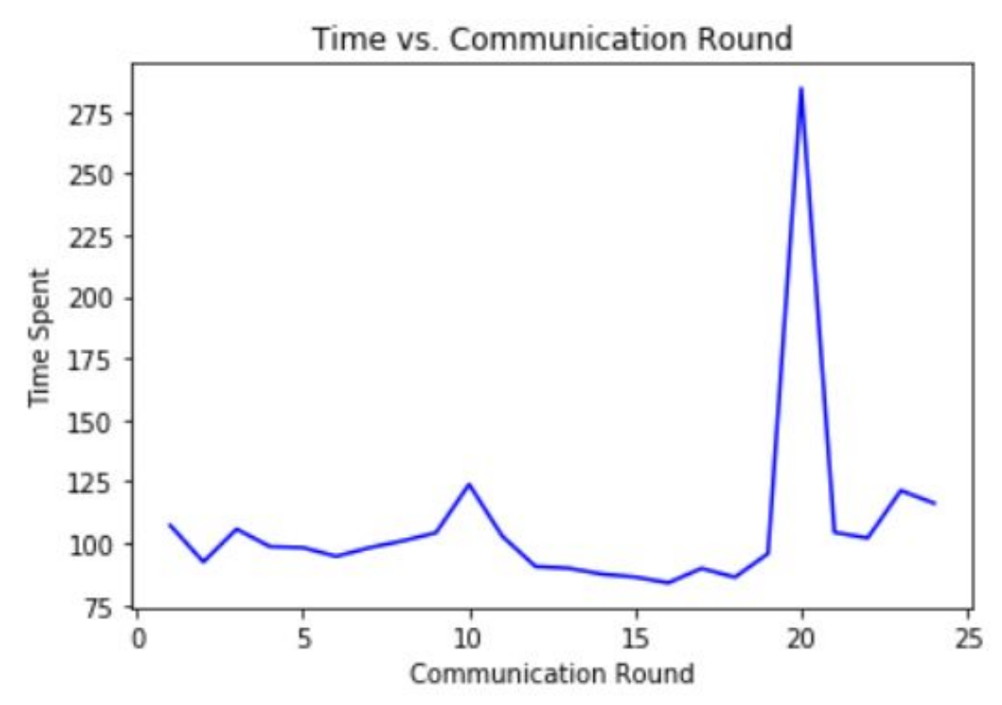
Each round: device pulls latest version of model, trains model on-premises using local data, pushes new weights

Reach consensus on new update by weighting updates by amount of data

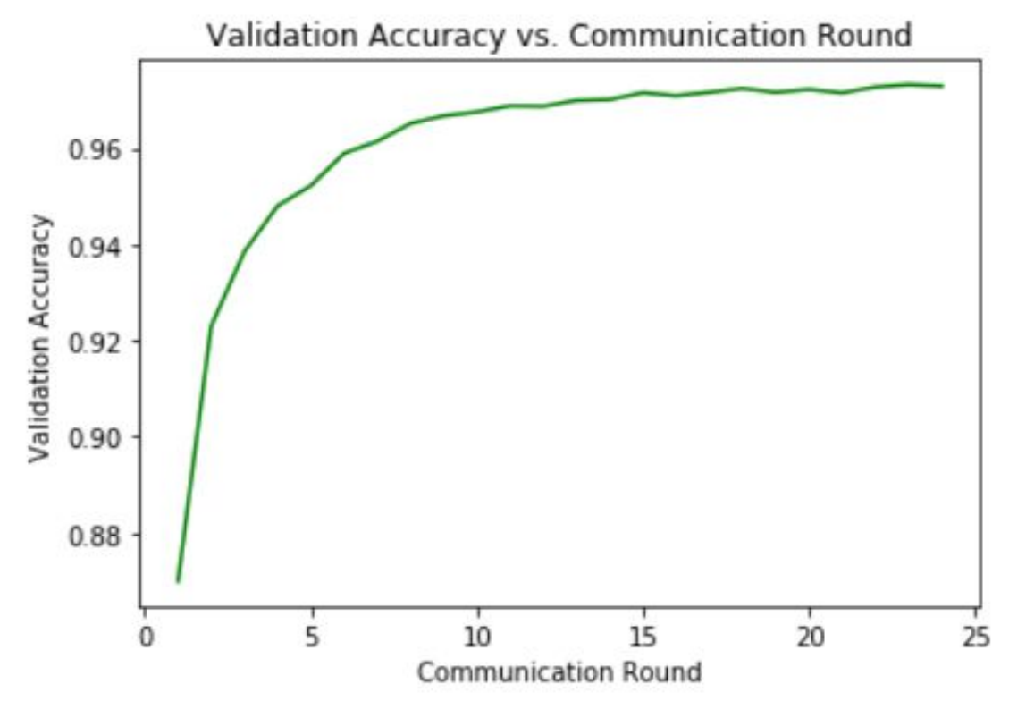


Experiments and Results

CNN | IID | Clients = .2 | Batch = 50 | Epochs = 5 | Learning = .1

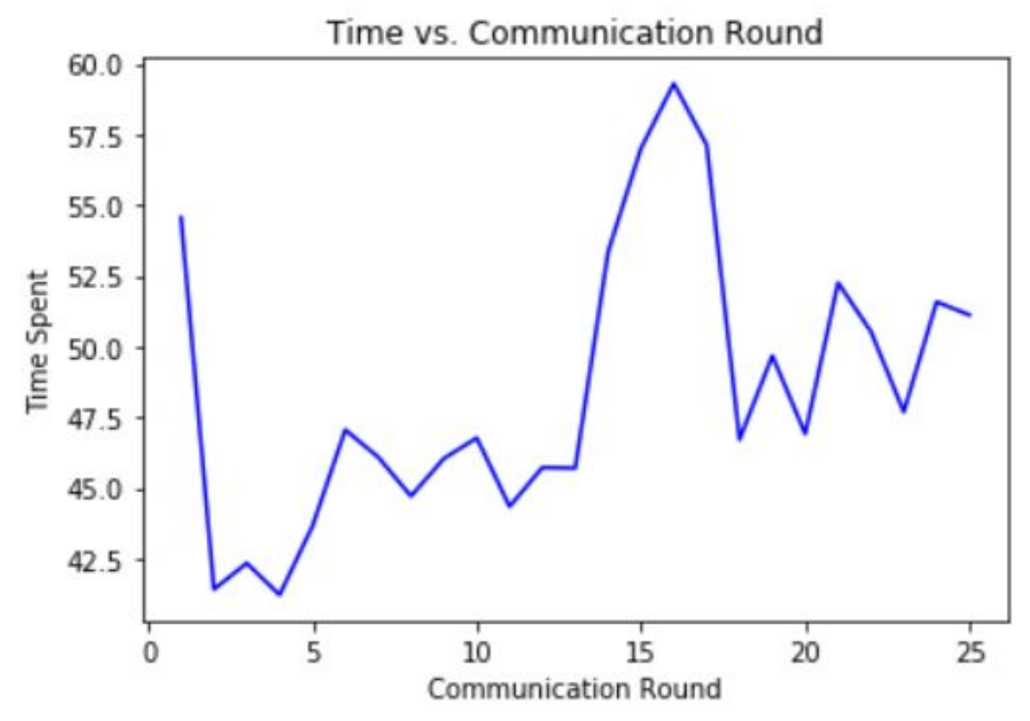


```
Total time spent: 2565.5246436595917 seconds.
Average time spent: 106.89686015248299 seconds.
```

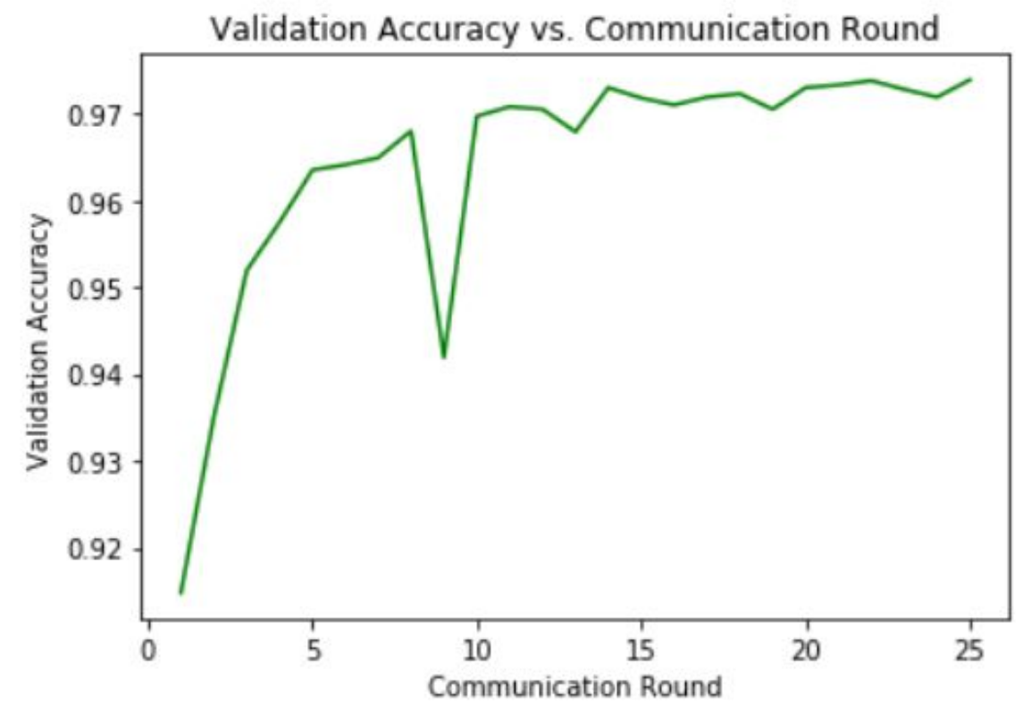


```
Best validation accuracy: 0.9732000231742859 (round: 23)
```

CNN | IID | Clients = .1 | Batch = 50 | Epochs = 5 | Learning = .1

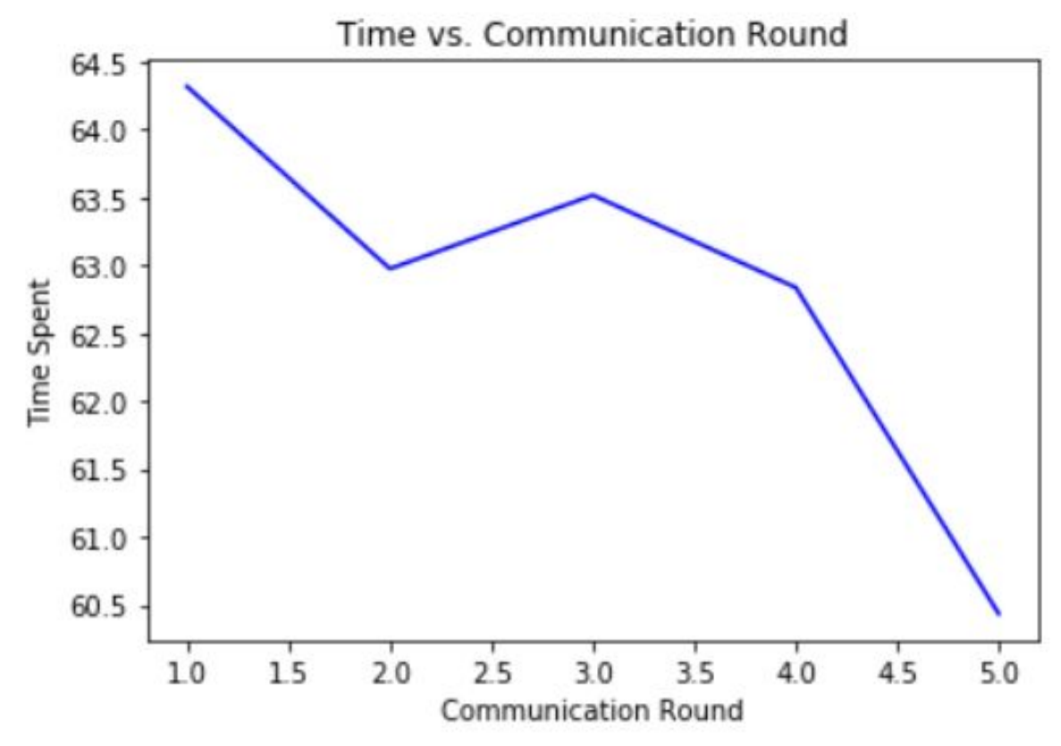


```
Total time spent: 1212.7709074020386 seconds.
Average time spent: 48.51083629608154 seconds.
```

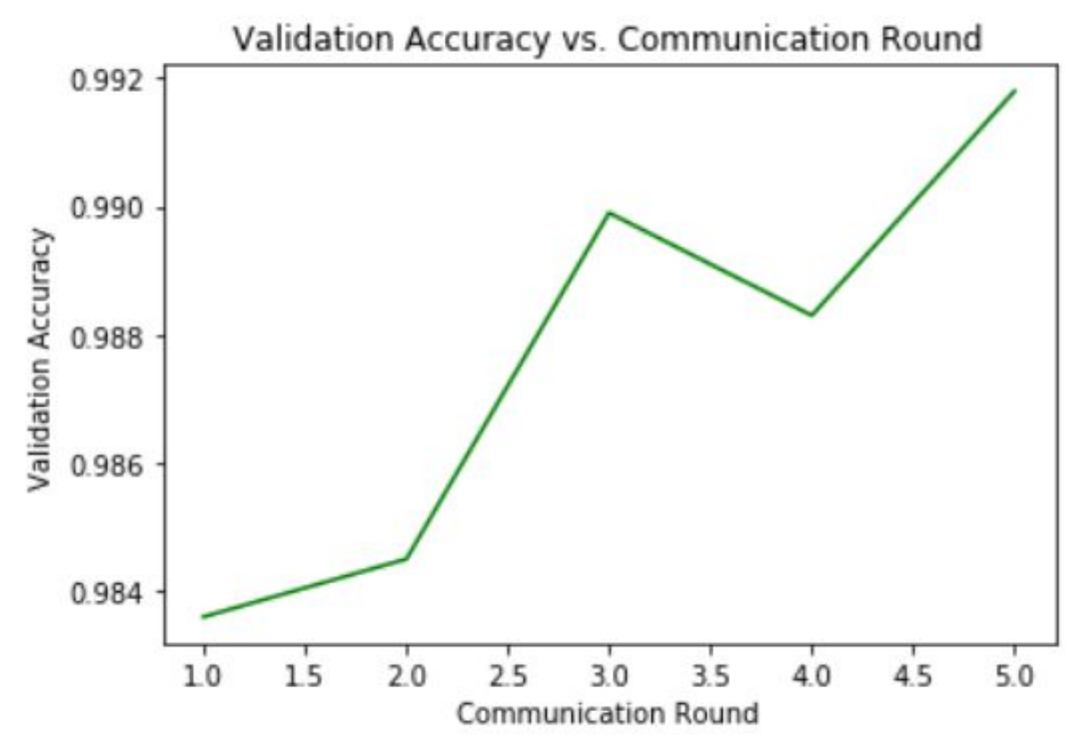


```
Best validation accuracy: 0.9739000201225281 (round: 25)
```

CNN | IID | Batch = 50 | Learning = .1 | Non-Federated



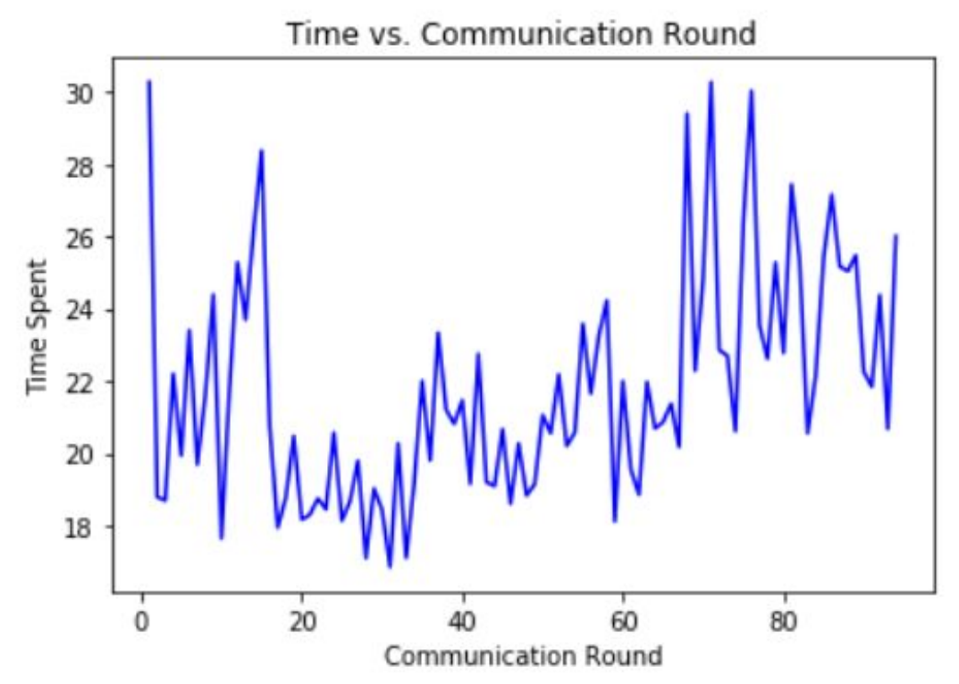
```
Total time spent: 314.0729789733887 seconds.
Average time spent: 62.814595794677736 seconds.
```



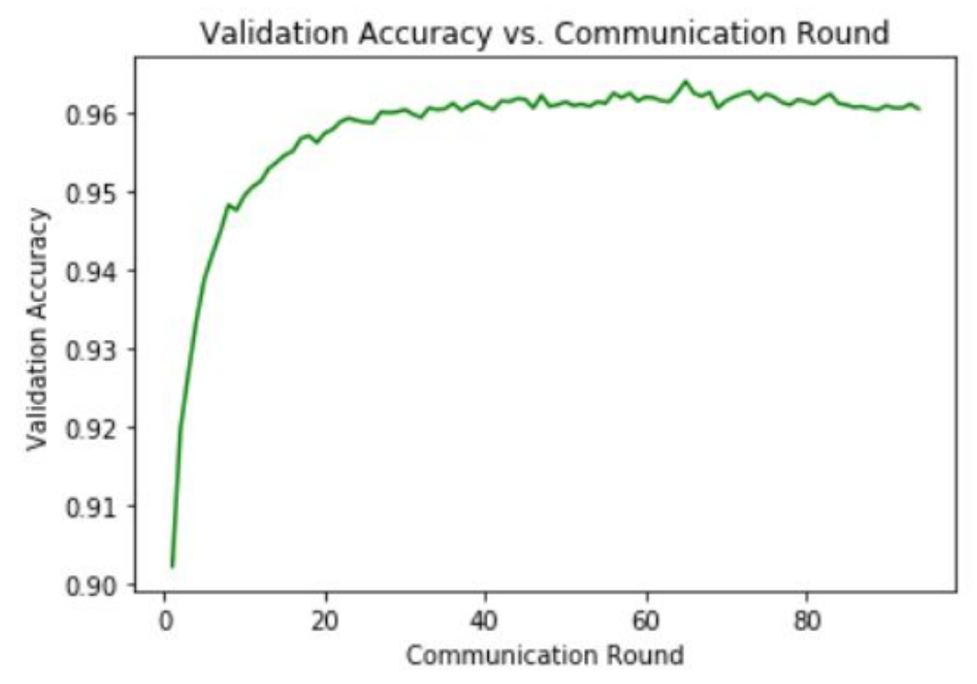
```
Best validation accuracy: 0.9918000102043152 (round: 5)
```

Experiments and Results

2NN | IID | Clients = .2 | Batch = 50 | Epochs = 10 | Learning = .1

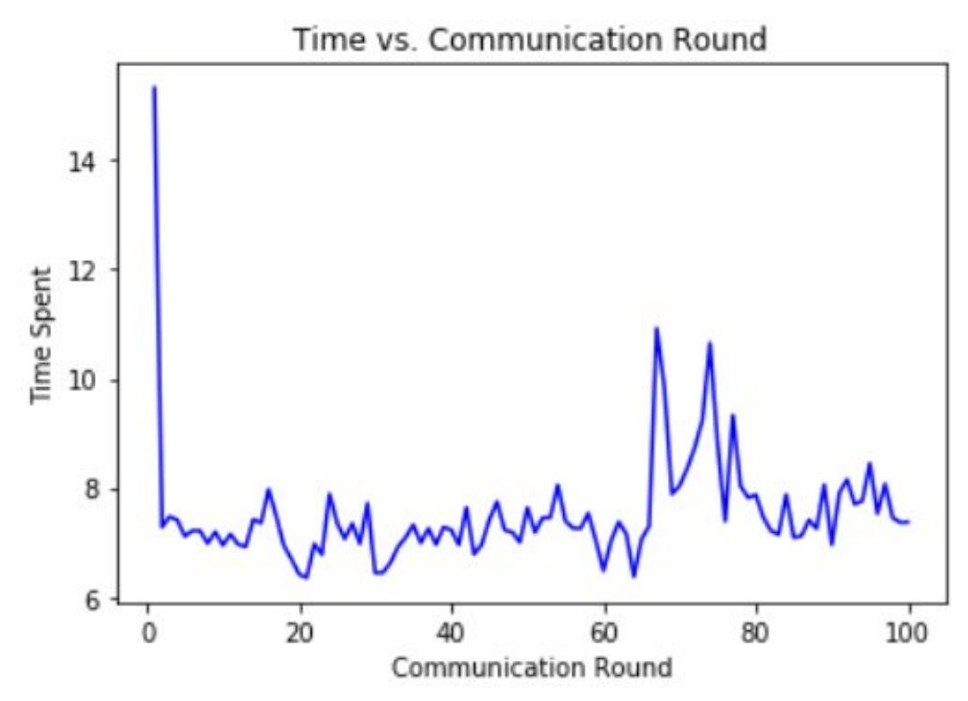


```
Total time spent: 2048.1862921714783 seconds.
Average time spent: 21.78921587416466 seconds.
```

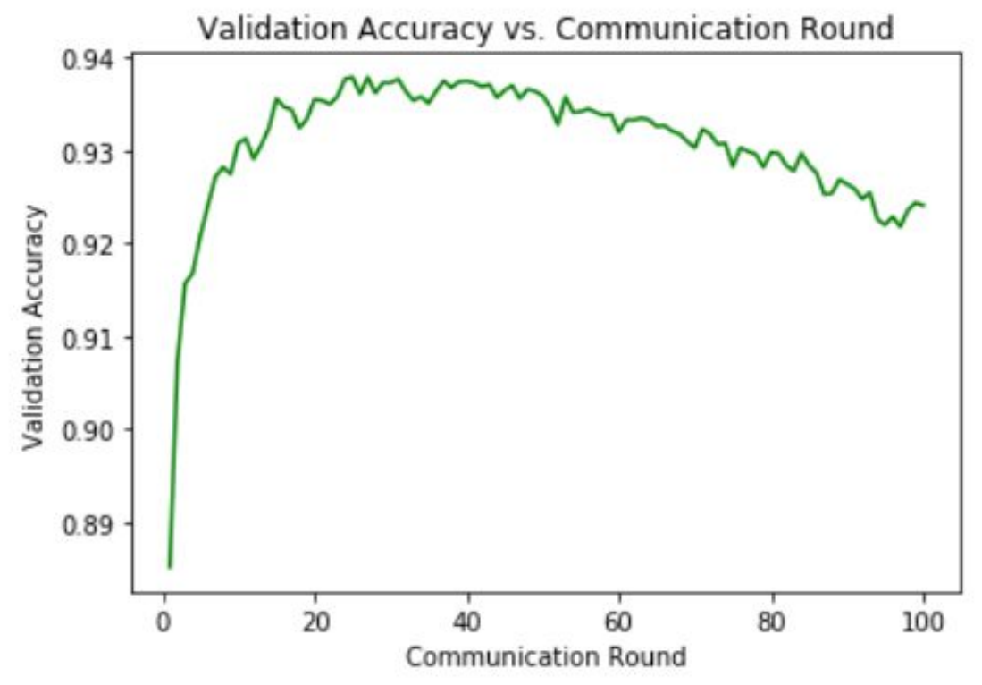


```
Best validation accuracy: 0.9639000296592712 (round: 65)
```

2NN | IID | Clients = .1 | Batch = 50 | Epochs = 10 | Learning = .1

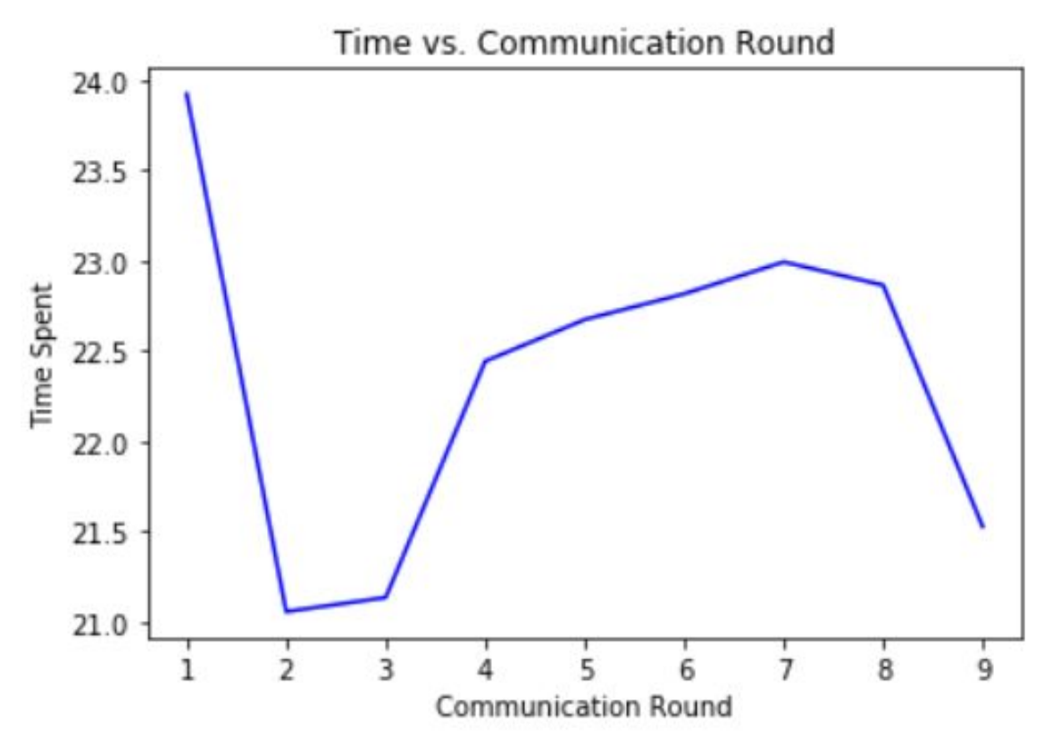


```
Total time spent: 754.5301134586334 seconds.
Average time spent: 7.545301134586334 seconds.
```

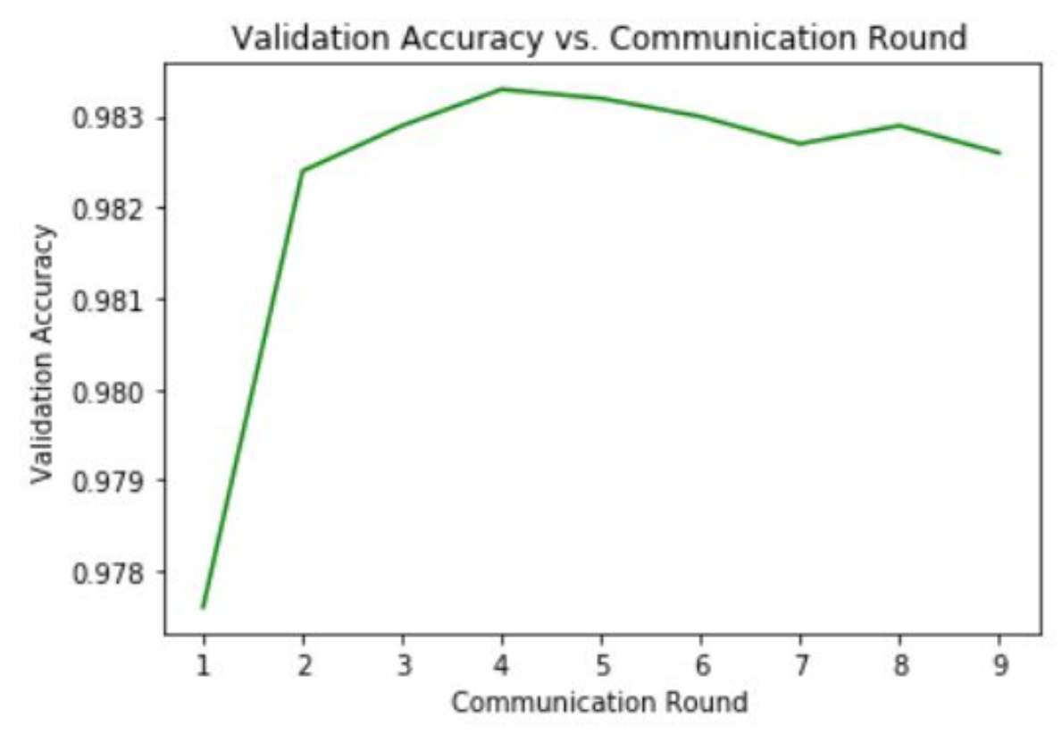


```
Best validation accuracy: 0.9379000067710876 (round: 25)
```

2NN | IID | Batch = 50 | Learning = .1 | Non-Federated



```
Total time spent: 201.42905640602112 seconds.
Average time spent: 22.381006267335678 seconds.
```



```
Best validation accuracy: 0.983299970626831 (round: 4)
```

Extension

Prior implementations used Keras so any benchmarks of speed would be inaccurate

We implemented model-agnostic federated learning in Tensorflow

Used Ray for concurrency/speedup and ADAM for faster convergence

Discussion and Analysis

Unable to replicate fast convergence to target accuracies of paper

Found B=50 better than paper's B=10

Paper emphasized importance of learning rate as hyperparameter but did not provide learning rate, so we did not use a gridsearched learning rate

Times inaccurate due to thread limitation vs num clients (4 vs 10-20)

Further Work

Need to gridsearch over learning rate hyperparameter field specified by paper and use optimal rate to redo experiments for 2NN and CNN

The next step would be implementing model update compression for speedup

References

"[1602.05629] Communication-Efficient Learning of Deep ... - arXiv." 17 Feb. 2016, https://arxiv.org/abs/1602.05629. Accessed 19 Mar. 2018.

"Federated Learning: Strategies for Improving Communication Efficiency." 18 Oct. 2016, https://arxiv.org/abs/1610.05492. Accessed 19 Mar. 2018.

"Distributed Mean Estimation with Limited ... - Research at Google." https://research.google.com/pubs/pub45672.html. Accessed 19 Mar. 2018.

“Research Blog: Federated Learning…”06 Apr. 2017 https://research.googleblog.com/2017/04/federated-learning-collaborative.html Accessed 19 Mar. 2018.