Communication-Efficient Learning of Deep Networks from Decentralized Data

Stefan Hofman (4887336), Mirza Mrahorović (4596536), Yen-Lin Wu (4848489)
Delft University of Technology

1) Introduction

- Enormous amounts of data on our mobile devices suitable for learning models, but private in nature.
- *Federated learning* allows users to collectively reap the benefits of shared models trained from such data without the need to centrally store it.
- Every *client* (k): trains on local data with SGD for E epochs with batch size B to obtain a gradient estimate (g_k) and sends updated weights (w^k) to the server, with learning rate η.
  $w^k \leftarrow w^k - \eta g_k$
- *Server*: combines the weights of the clients (k) to obtain a new model (w_{t+1}) and re-distributes the new model back to the clients for further training.
  $w_{t+1} \leftarrow \frac{n_k}{K} \sum_{k=1}^{K} w_{t+1}^k$

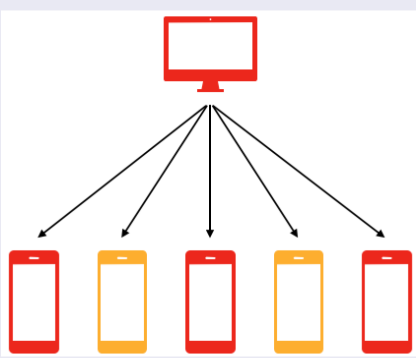
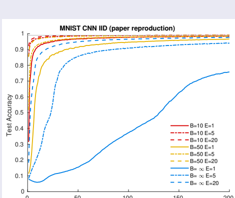


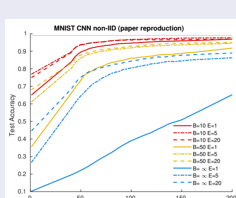
Figure: Sketch of a federated environment

2.1) Replication

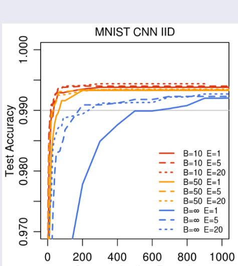
- reproduction of the original results from the paper



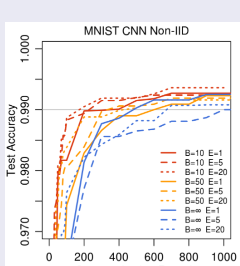
(a) replication IID



(b) replication non-IID



(c) Original IID



(d) Original non-IID

2.2) Unbalanced data distribution w/o weights

- unbalanced data distribution w/o weight contribution of clients; i.e. the partition P_k of the training data is independent but unevenly distributed, which affects the gradient estimates g_k

$$w^k \leftarrow w^k - \eta \frac{1}{n_k} \nabla \sum_{i \in P_k} L_i(w)$$

$$w_{t+1} \leftarrow \frac{1}{K} \sum_{k=1}^{K} w_{t+1}^k$$

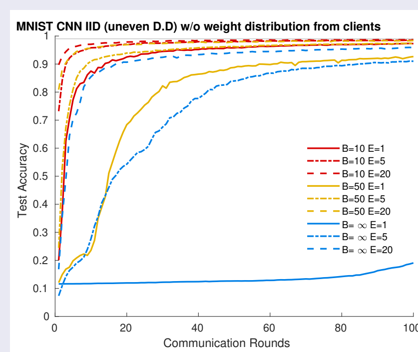


Figure: Uneven data distribution w/o weight contribution of clients

2.3) Unbalanced data distribution w/ weights

- unbalanced data distribution w/ weight contribution of clients; i.e. averaging model is based on fraction of local training data

$$w^k \leftarrow w^k - \eta \frac{1}{n_k} \nabla \sum_{i \in P_k} L_i(w)$$

$$w_{t+1} \leftarrow \frac{n_k}{K} \sum_{k=1}^{K} w_{t+1}^k$$

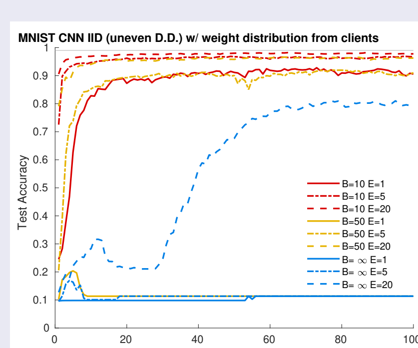


Figure: Uneven data distribution w/ weight contribution of clients

2.4) Original algorithm with Gaussian noise appended

- Gaussian noise added to weight updates before communication with clients

$$w^k \leftarrow w^k - \eta \frac{1}{n_k} \nabla \sum_{i \in P_k} L_i(w)$$

$$w_{t+1} \leftarrow \frac{n_k}{K} \sum_{k=1}^{K} \left(w_{t+1}^k + N(0, \sigma^2) \right)$$

Best performing model trains on batch size of 10 for 20 local epochs.

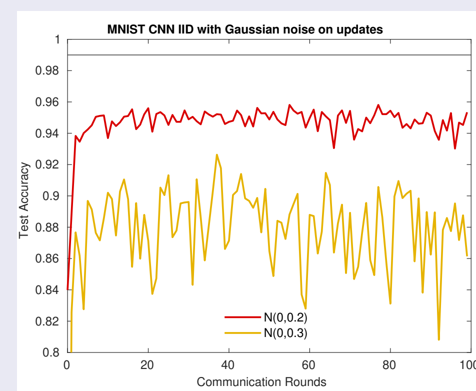


Figure: Learning curves of model with Gaussian noise on updates

3) Conclusion & Discussions

- replication has slightly worse results than original paper
- unbalanced data distribution led to worse performance
- unbalanced data distribution corrected with weights does not solve the problem
- adding Gaussian noise led to worse algorithm accuracy, yet the model converged

Table: Amount of rounds needed to reach 99% accuracy

E	B	IID	NON-IID	ud_IID	ud_w_IID
1	10	-	-	-	-
5	10	118	-	-	-
20	10	130	-	-	-
1	50	-	-	-	-
5	50	-	-	-	-
20	50	124	-	-	-
1	∞	-	-	-	-
5	∞	-	-	-	-
20	∞	-	-	-	-

References

H. Brendan McMahan and Eider Moore and Daniel Ramage and Seth Hampson and Blaise Agüera y Arcas. 1602.05629, 2016. *https://arxiv.org/abs/1602.05629*

https://github.com/shaoxiongji/federated-learning