Deep Learning and its Applications
Project Presentation on
Federated Learning for Deep Neural Networks

Group-17
Debashis Sahoo, Ganesan P

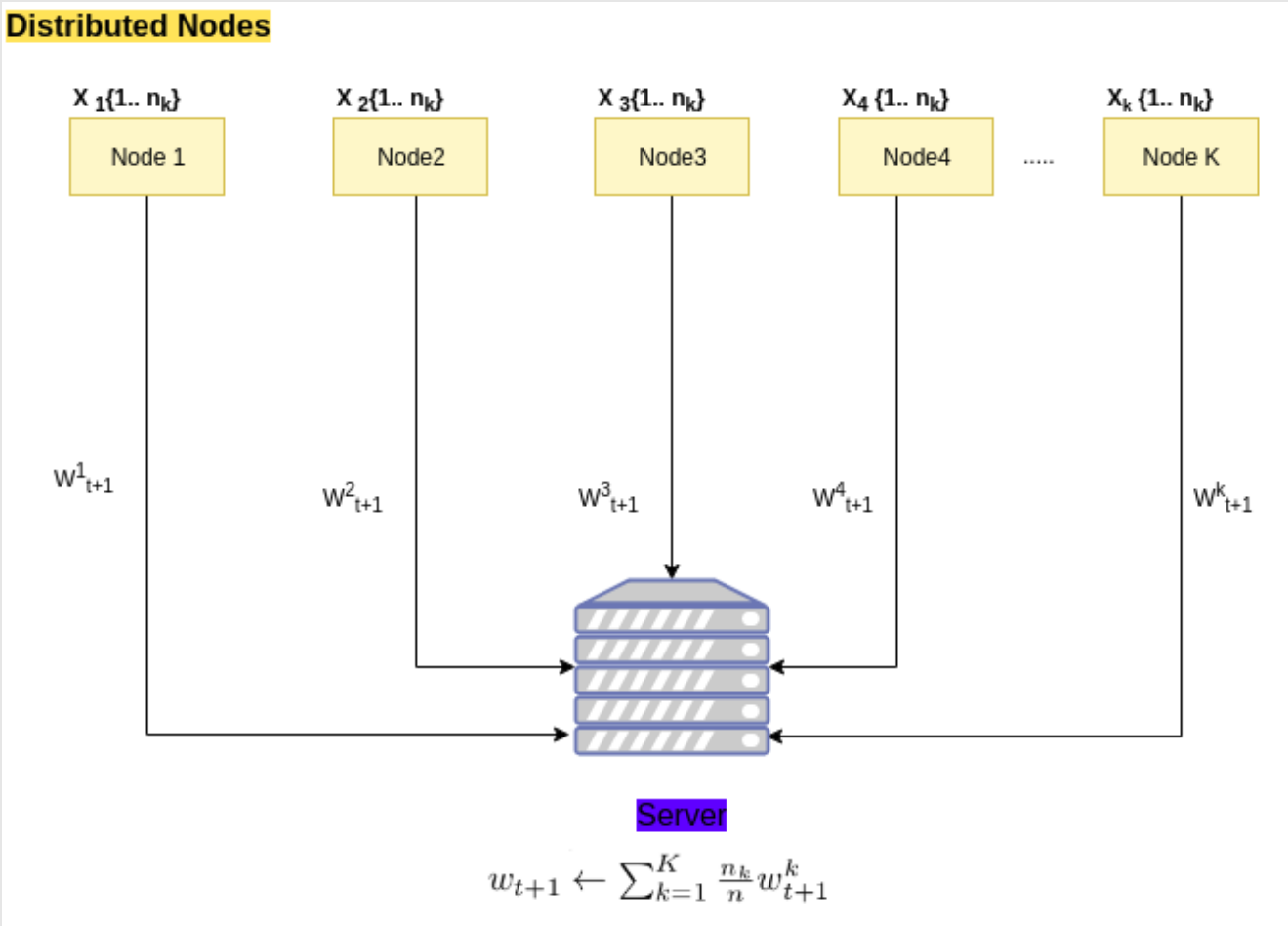
June 7, 2019

Problem Statement

The idea is to build Deep Neural Nets from distributed data sets and ensuring data privacy. Deep Learning models can be trained in the distributed nodes such that there is no need to collect any sensitive data to the central server inorder to train the model, instead the collective model can be built from the distributed models.

Problem Statement

Federated Learning for Deep Neural Networks



Motivation and challenges

- The large volume of data required for building deep learning models are present with the edge devices such as mobiles, laptops, etc. In order to build deep learning models, it is necessary to bring the data to the central server and should prevent from any data leakage.
- The General Data Protection Regulation (GDPR) enforces the users personal privacy and data security when the data is used by companies for technical purposes [1].
- An approach called Federated Learning can train the deep neural nets in a distributed environment such that data remains isolated with the users and ensures the data privacy [2].
- However, the federated learning requires more number of communication rounds to build the models with the predefined accuracy.
- Although, the federated learning provides data security, it is possible that the honest and curious server learns the data during the model building process.

Literature Survey

Communication-Efficient Learning of Deep Networks from Decentralized Data [2]

- Independent and Identically Distributed Data $\{1 \ldots n_k\}$ or Non-i.i.d $\{1 \ldots n_k\}$.
- Number of distributed nodes: K
- Local epoch: E
- Batchsize: B
- # Parallel nodes is defined by: C
- # local update per client per round $u = E * n/(K * B)$
- Models trained : 2NN, CNN and LSTM.
- Dataset used: MNIST handwritten data, CIFAR-10, and The complete work of William-Shakespeare.

Literature Survey

Federated Averaging Algorithm

Algorithm 1 `FederatedAveraging`. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

Server executes:

initialize w_0

for each round $t = 1, 2, \dots$ **do**

$m \leftarrow \max(C \cdot K, 1)$

$S_t \leftarrow$ (random set of m clients)

for each client $k \in S_t$ **in parallel do**

$w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$

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

ClientUpdate(k, w): *// Run on client k*

$\mathcal{B} \leftarrow$ (split $\mathcal{P}_k$ into batches of size B)

for each local epoch i from 1 to E **do**

for batch $b \in \mathcal{B}$ **do**

$w \leftarrow w - \eta \nabla \ell(w; b)$

return w to server

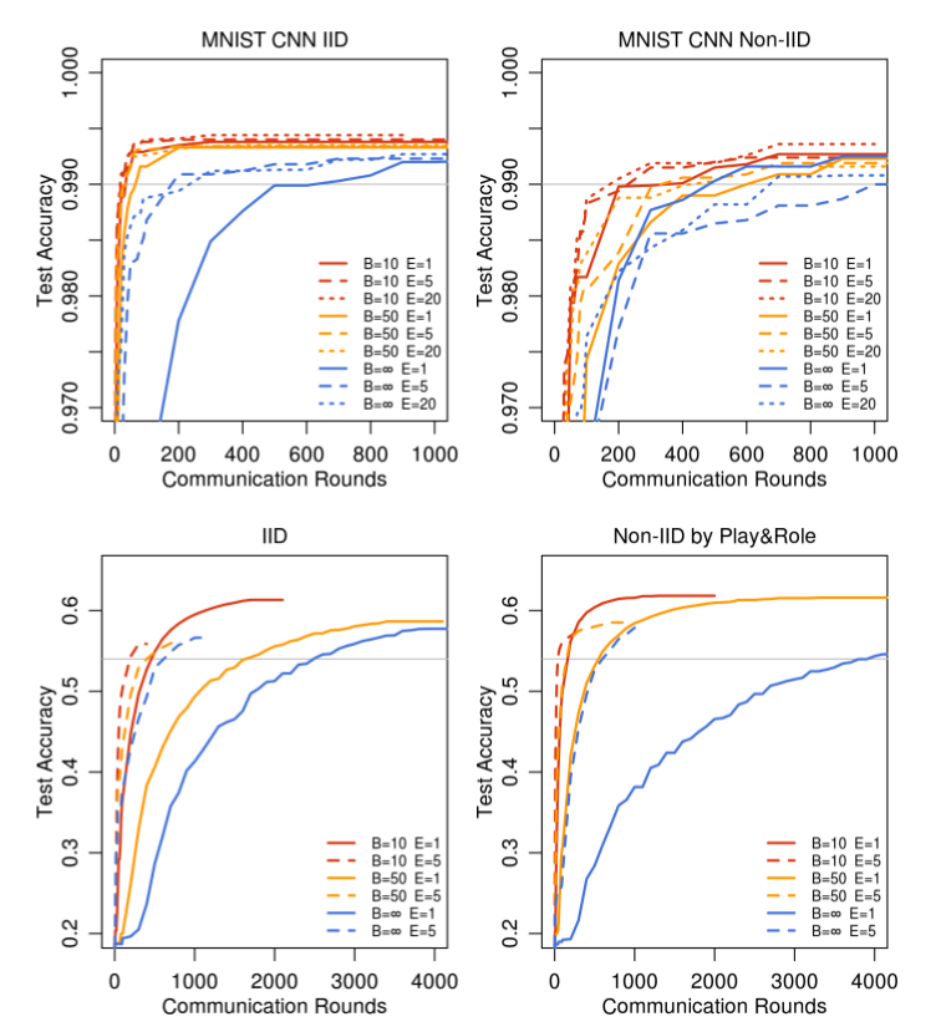
Literature Survey

Number of Communication rounds to reach a target accuracy

MNIST CNN, 99% ACCURACY					
CNN	E	B	u	IID	NON-IID
FEDSGD	1	∞	1	626	483
FEDAVG	5	∞	5	179 (3.5×)	1000 (0.5×)
FEDAVG	1	50	12	65 (9.6×)	600 (0.8×)
FEDAVG	20	∞	20	234 (2.7×)	672 (0.7×)
FEDAVG	1	10	60	34 (18.4×)	350 (1.4×)
FEDAVG	5	50	60	29 (21.6×)	334 (1.4×)
FEDAVG	20	50	240	32 (19.6×)	426 (1.1×)
FEDAVG	5	10	300	20 (31.3×)	229 (2.1×)
FEDAVG	20	10	1200	18 (34.8×)	173 (2.8×)
SHAKESPEARE LSTM, 54% ACCURACY					
LSTM	E	B	u	IID	NON-IID
FEDSGD	1	∞	1.0	2488	3906
FEDAVG	1	50	1.5	1635 (1.5×)	549 (7.1×)
FEDAVG	5	∞	5.0	613 (4.1×)	597 (6.5×)
FEDAVG	1	10	7.4	460 (5.4×)	164 (23.8×)
FEDAVG	5	50	7.4	401 (6.2×)	152 (25.7×)
FEDAVG	5	10	37.1	192 (13.0×)	41 (95.3×)

Literature Survey

Test set accuracy vs Communication rounds



Literature Survey

- Communication efficient distributed algorithms such as Federated Averaging(Fed-Avg), Federated Stochastic Variance Reduced Gradient(FSVRG) and CO-OP are the state of the art approaches for model aggregation. The performance evaluation of these algorithms is carried out and the comparison suggest that the Fed-Avg performs better than other algorithms [3]. CO-OP algorithm has been used in asynchronous communication settings, but the Fed-Avg and FSVRG are used in the synchronous settings [3].

Datasets

1. MNIST handwritten data
2. CIFAR-10 [4].
3. The complete work of William-Shakespeare [5].
4. NMSWorks Performance Data.

Proposed Methodology

- TensorFlow supports federated learning of Deep Neural Nets using TFF API. The existing implementation is done using TFF.
- Federated Learning methodology can be applied with NMSWorks Performance management data such that the BNG_auth_queue,In-Bandwidth and Out-bandwidth target values are predicted properly.
- LSTM model is being built with the performance data created and the federated learning setup can be implemented on the existing model.

References

Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong,

"Federated machine learning: Concept and applications,"
ACM Trans. Intell. Syst. Technol., vol. 10, no. 2, pp. 12:1–12:19, Jan. 2019.

H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas,

"Communication-efficient learning of deep networks from decentralized data,"
in *AISTATS*, 2017.

Adrian Nilsson, Simon Smith, Gregor Ulm, Emil Gustavsson, and Mats Jirstrand,

"A performance evaluation of federated learning algorithms,"
in *Proceedings of the Second Workshop on Distributed Infrastructures for Deep Learning, DIDL@Middleware 2018, Rennes, France, December 10, 2018*, 2018, pp. 1–8.

Alex Krizhevsky,

"Learning multiple layers of features from tiny images,"
2009.

William Shakespeare,

"The complete works of william shakespeare," https://www.gutenberg.org/ebooks/100.