Compressive Differentially-Private Federated Learning Through Universal Vector Quantization

Saba Amiri[1*], **Adam Belloum**[1], **Sander Klous**[2], **Leon Gommans**[3]

[1]Multiscale Networked Systems (MNS) Research Group, University of Amsterdam, 1098 XH Amsterdam, The Netherlands
[2]Complex Cyber Infrastructure (CCI) Research Group, University of Amsterdam, 1098 XH Amsterdam, The Netherlands
[3]Air France KLM
{s.amiri,a.s.z.belloum,s.klous,leon.gommans}@uva.nl

Abstract

Collaborative and federated machine learning is an essential vehicle for achieving privacy preserving machine learning. By not forcing participants to share their private datasets, we can adhere to strict local as well as international privacy protection legal regimes. However, a federated learning mechanism is usually hindered by the overhead communication costs between the central server and participants when communication channels have constrained capacities. Furthermore, just refraining from sharing the data will not lead us to absolute privacy, e.g. in presence of privacy attacks such as membership inference. Differential Privacy has provided a set of rigorous privacy standards to protect individual records of a dataset being used by a randomized mechanism. As differential privacy has been widely accepted as the de facto privacy standard in machine learning, it could potentially mitigate privacy concerns. However, addition of differential privacy usually costs even more communication overhead, putting more pressure on uplink and downlink channels. In this work, we present a novel algorithm for achieving both differential privacy and reduced communication overhead through compression of client-server communication by means of quantization. Not only we show acceptable levels of differential privacy, we also show significant gains in terms of communication efficiency by compressing the data on the constrained uplink channel.

1 Introduction

Traditional machine learning paradigms usually require for all training data to be accumulated in one place. Nowadays, this practice poses several challenges, two of the most important instances of which being constrained capacity of communication channels for data transfer and privacy concerns. Collaborative and federated learning mechanisms solve this problem by instead bringing the computation to the data, making the learning process distributed. This negates the privacy-violating process of transferring private datasets from participants to a central authority, but doesn't solve the privacy challenges completely.

Aside from a more complicated risk model by introducing communication channels and usually a central authority to the machine learning pipeline, this collaborative learning scheme too suffers from the communication overhead similar to that of centralized learning. The difference is, instead of the overhead related to communicating big datasets once, the extra communication load is due to performing numerous rounds of training over a distributed setting. Smaller packets of model parameter are sent on the uplink and downlink channels between participants and the parameter server numerous times, making accumulative communication overhead high.

Regarding privacy aspects of federated machine learning, it has been shown that they can still leak critical information about their training dataset as well as the model parameters despite not forcing the participants to share their private data (Song, Ristenpart, and Shmatikov 2017; Nasr, Shokri, and Houmansadr 2018). Thus, we need to take additional measures to ensure the privacy of the training data is preserved. In recent years, the de facto standard of privacy for data access mechanisms has been Differential Privacy (Dwork, Roth et al. 2014) and its applications have been rigorously researched in machine learning (Abadi et al. 2016; Chamikara et al. 2019). Differential Privacy could theoretically be achieved via perturbation as long as the bounds of the introduced perturbation are calculable. The main idea behind this work stems from the necessity of compressing communications between participants and central authority and considering the inevitable transformation of communicated data as a form of perturbation. Thus with regards to the previously defined general problems of federated learning, namely communication overhead and privacy and considering that , one can pose the research question "Is is possible to use the perturbation resulting from compression of communication in a federated learning scenario to achieve Differential Privacy?".

In this work, we investigate the above research question and present a method for differentially private compression of the communicated parameters between participants and the parameter server through compression. Using this method we can achieve not only more efficient communication, but also an effective level of privacy in local domain, i.e. sample level privacy.

1.1 Our Contribution

- We design a novel method to achieve differential privacy through quantization of the federated communications.

*Corresponding author.

The novelty of this mechanism is to translate the perturbation by the compression method, namely universal lattice quantization, into measurable Gaussian noise independent of the source distribution using dithering mechanism. To the best of our knowledge, this is the first time the perturbation introduced by quantization has been employed as a source of noise to achieve differential privacy.

- We provide analysis of the quantization noise in different quantization schemes and connect their respective noise models to that of specific differentially-private noise adding mechanisms, making the connection between the two paradigms.
- We provide algorithms for compressive differentially private federated learning both to achieve local differential privacy. We design experiments and report preliminary results, proving the system can achieve compression while maintaining an acceptable level of privacy and utility.

2 Preliminaries

This section provides preliminaries and background information on federated learning, differential privacy, select noise-adding mechanisms for differential privacy and universal lattice vector quantization as well as the noise models of the quantization techniques.

2.1 Horizontal Federated Learning

Federated learning (FL) (McMahan et al. 2017) is a collaborative learning scheme for distributed training of machine learning systems on multiple participants without them having to share their respective private datasets. In FL, the private datasets are accessed and processed locally, after which a central Parameter Server (PS), usually acting as orchestrator and aggregator, gathers and combines the local updates and returns the updated parameter set to the learning parties.

In the basic fusion algorithm proposed by McMahan et al., namely Federated Stochastic Gradient Descent (`FedSGD`), one step of gradient descent is done per learning round. Supposing N participants in total, in the first round the PS randomly generates the parameter set θ_0 and communicates it to all learning parties. Then at the round $r \geq 1$ the ith participant P_i will perform one round of the training, compute its local gradients $g_{i,r}$ and send them back to the PS. After receiving the parameters sets from $M \leq N$ participants, the mechanism F on parameters server attempts to compute the new weights, θ_{r+1}, using the M parameter sets as

$$F(g_{1,r}, ..., g_{M,r}) = \theta_r - \eta \sum_{i=1}^{M} w_i g_{i,r}$$

where η is the PS learning rate and w_i is the weight assigned to each participant, e.g. the percentage of the samples they are hosting.

2.2 Differential Privacy

Differential Privacy (DP) is a set of mathematical conditions to provide plausible deniability to every single member of the training set in the context of machine learning. Considering adjacent datasets D and D' which differ in only one element and the randomized mechanism M mapping the domain X to the range R, $M : X \to R$, the mechanism M is (ϵ, δ)-*differentially private* if for all possible outputs $S \subseteq R$

$$Pr[M(D) \in S] \leq e^{\epsilon}[Pr[M(D') \in S] + \delta$$

. ϵ (privacy budget) controls the trade-off of utility-privacy and δ allows for a small probability of failure for the privacy preservation mechanism. Simply put, DP caps the increase in information learned by changing a single element in the dataset.

2.3 Making Mechanism M Differentially Private

Methods used to make randomized mechanisms private usually involve some form of perturbation, e.g. noise addition. Noise addition mechanisms are usually applied to the output of query q. The amount of noise is controlled by the sensitivity of the q, defined as maximum change in the output of q for any adjacent datasets D and D':

$$S_q = max \, \|q(D) - q(D')\|$$

.

$(0, \delta)$-Differential Privacy with Uniform Noise Since the distribution of the ideal universal quantization noise is uniform, in this section we study DP in presence of uniform noise. According to (Geng and Viswanath 2015), in case of $(0, \delta)$-DP, the optimal noise mechanism for $S_q = 1$ has a uniform probability distribution

$$P(x) = \begin{cases} \frac{\delta}{S_q} & -\frac{S_q}{2\delta} \leq x \leq \frac{S_q}{2\delta} \\ 0 & otherwise \end{cases} \quad (1)$$

This result has been iterated in (Geng et al. 2019) as a special case of truncated Laplacian mechanism.

For $\epsilon \geq 0$, $0 \leq \delta \leq \frac{1}{2}$ and $S_q \geq 0$, when $\epsilon \to 0$, it will be reduced to a uniform distribution with the support of $\left[-\frac{S_q}{2\delta}, \frac{S_q}{2\delta}\right]$ and probability density of $\frac{\delta}{S_q}$.

Analytical Gaussian Mechanism The Gaussian noise adding mechanism introduced in (Dwork, Roth et al. 2014) has a limitation on ϵ, namely $0 \leq \epsilon \leq 1$. (Zhao et al. 2019) show how this mechanism has been misused with $\epsilon > 1$, leading to a loss of privacy. Balle and Wang in (Balle and Wang 2018) define analytical Gaussian noise adding mechanism $Z \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$ and show it to be (ϵ, δ)-DP for any $\epsilon \geq 0$, $\delta \in [0, 1]$ if and only if $\mathbf{\Phi}(\frac{S_q}{2\sigma} - \frac{\epsilon\sigma}{S_q}) - e^{\sigma}\mathbf{\Phi}(-\frac{S_q}{2\sigma} - \frac{\epsilon\sigma}{S_q}) \leq \sigma$, with $\mathbf{\Phi}$ being the Gaussian Comulative Distribution Function (CDF). They also provide a numerical algorithm to compute the optimal value of σ, which we will leverage to fine tune out quantization parameters.

2.4 Differential Privacy Domains

In a FL setting, the DP aspect of the mechanism, i.e. the machine learning algorithm, can be defined as either *Global* or *Local*. Local Differential Privacy (LDP) in a FL setting is applied on participant level. According to the definition of LDP provided by (Bebensee 2019; Kasiviswanathan et al.

2011), a randomized mechanism $M : X \rightarrow R$ adheres to LDP if for any $x, x' \in X$ and set of $y \in R$

$$Pr[M(x) = y] \leq e^{\epsilon}[Pr[M(x') = y] + \delta$$

Here the output value y is perturbed using a DP mechanism. Global Differential Privacy (GDP) in FL is applied on aggregation level, e.g. on PS. For the purpose of this work, the mechanism F defined in 2.1 needs to adhere to DP conditions defined in Section 2.2. This ensures that all participant parameter sets will be protected under DP assumptions.

3 Proposed Method

In this section we propose a (ϵ, δ)-*differentially private* compressive federated learning model to preserve local differential privacy. First, we provide two quantization schemes with subtractive and non-subtractive dither. We provide analysis of quantization noise for these methods. Next, we describe our differentially private federated lea

3.1 Lattice Quantization and Its Noise Model

One of the challenges introduced by the FL paradigm is the communication overhead between the participants and the PS, especially in the presence of constrained *uplink* and *downlink* channels. In this work we assume a constrained capacity for the uplink channel. Therefore, optimizing the communication component of the FL scheme is crucial. The desired optimization could be achieved through compression of the data packages being sent to PS by participants.

In the following section we will introduce *Lattice Vector Quantization* method as a way to achieve data compression. We define *Subtractive Dithered* and *Non-Subtractive Dithered Universal Vector Quantization* (SDUVQ, NSDUVQ respectively) and formalize them as randomized quantization methods.

Universal Lattice Vector Quantization Universal Lattice Vector Quantization (LVQ)(Gersho 1979) which comprises of Quantization followed by a lossless coding scheme is the most efficient entropy coded vector quantization method when the quantization rate goes to infinity(Gray and Neuhoff 1998). Introducing subtractive dither to the LVQ makes it *universal*(Zamir and Feder 1992), i.e. the quantization noise mechanism will be independent of the source distribution. This type of quantization noise is called a Pseudo Quantization Noise model (PQN). In case of non-subtractive dither, it has been shown that the Gaussian dither *essentially* can be modelled as PQN with quantization noise independent of input distribution if $\sigma \geq \frac{\Delta}{2}$. The two SDUVQ and NSDUVQ defined in this section are based on the uniform vector quantization scheme of (Dragotti and Gastpar 2009) with a simple lattice code design and introduction of dither. Each scheme will have an *encoding* and a *decoding* path. The encoding happens at source, i.e. participant, to compress the input data while the decoding happens at the destination to reconstruct the data.

Consider the N-dimensional vector $\vec{\mathbf{w}}$. We define the quantization step size Δ and lattice dimension L. The quantization parameters are known by both source and destination.

In the ***encoding path***, to quantize $\vec{\mathbf{w}}$ first we break it down into $v = \frac{N}{L}$ smaller vectors of the size L $\left\{w_{0..}\vec{L-1}, w_{L..}\vec{2L-1}, ..., w_{(v-1)}\vec{L..vL-1}\right\}$. For the sake of simplicity we assume $\frac{N}{L}$ is an integer, but the scheme can easily be modified by taking the $\lceil \frac{N}{L} \rceil$ instead if this condition doesn't hold true. Next, we will generate the v-dimensional dither vector $\vec{\mathbf{d}} = [d_1, d_2, ..., d_v]$ by generating in an i.i.d fashion v uniform random variables with the support $\left[-\frac{\Delta}{2}, \frac{\Delta}{2}\right]$. We can also generate Gaussian dither with $\mu = 0$ and $\sigma \geq \frac{\Delta}{2}$. Generating dither from uniform noise ensures source distribution-independent quantization error(Zamir and Feder 1996) in case of subtractive dither. For non-subtractive dither, it has been shown that the Gaussian dither comes extremely close to the PQN model, rendering the quantization noise independent of the source distribution and identified by the same Gaussian distribution(Widrow and Kollár 2008) (Chapter 19, Section 6.5). Next step would be applying the uniform quantization method of (Dragotti and Gastpar 2009) but before that, we need to randomize the source vector using our generated dither to compute $\vec{\mathbf{w_d}}$. To this end, each element of the dither vector $\vec{\mathbf{d}}$ will be added to the corresponding lattice cell, e.g. $\vec{w_{di}} = \{w_{iL} + d_i, w_{iL+1} + d_i, .., w_{iL-1} + d_i\}$. Then, we apply the uniform quantization to each dithered lattice cell $\vec{w_{di}}$ as $\vec{w_{qi}} = \Delta \texttt{round}(\frac{\vec{w_{di}}}{\Delta})$. In the last stage of the encoding path, as per (Zamir and Feder 1992), we will use a lossless entropy source encoder to compress $\vec{\mathbf{w_q}}$. There are a number of lossless source coding algorithms introduced in the literature. For the purpose of this work we chose the *bzip2*(Seward 1996) which uses the Burrows-Wheeler transformation(Kruse and Mukherjee 1999). Applying the bzip2 algorithm on $\vec{\mathbf{w_q}}$, we will end up with vector $\vec{\mathbf{w_e}}$ and codebook C

In the ***decoding path***, to reconstruct the vector $\vec{\mathbf{w}}'$ we need to decode the $\vec{\mathbf{w_e}}$ using codebook C, resulting in vector $\vec{\mathbf{w_q}}$. This will be the end of the encoding path for the NSDUVQ scheme.

The SDUVQ goes through additional steps of dither subtraction. First, the vector $\vec{\mathbf{w_q}}$ is divided into v lattice cells. Next, using the shared source of randomness the random vector $\vec{\mathbf{d}}$ is reconstructed. Finally, the corresponding dither values of $\vec{\mathbf{d}}$ are subtracted from vector $\vec{\mathbf{w_q}}$ leading to the reconstructed input vector $\vec{\mathbf{w}}'$.

The probability distribution of the noise of SDUVQ scheme is completely independent of the input distribution(Zamir and Feder 1992) and can be defined as

$$P(x) = \begin{cases} \frac{1}{2\Delta} & -\Delta \leq x \leq \Delta \\ 0 & otherwise \end{cases} \tag{2}$$

As mentioned earlier, the quantization noise of NSDUVQ with Gaussian dither can essentially behave as a PQN model when $\sigma \geq \frac{\Delta}{2}$, and we end up with a Gaussian noise pdf.

3.2 NSDUVQ-based Differentially Private Federated Learning Mechanism

Algorithm 1 and algorithm 2 provide the federated learning steps on participant and PS respectively for P participants.

We consider the case where the uplink from participant to the PS is constrained while the downlink has enough capacity to handle the traffic efficiently, thus we only need to increase efficiency in the uplink. The server is considered honest but curious and we need to maintain local differential privacy on sample level. Since uniform noise of the subtractive dither is inefficient for differential privacy without significant loss of utility, we use non-subtractive dither method with Gaussian noise adding mechanism. The value of σ is fixed at $\frac{\Delta}{2}$.

Algorithm 1: Compressive (ϵ, δ)-DP Federated Learning - Participant i, Round $t > 1$

Input: Training samples $D_i = \{x_{1i}, ..., x_{Ni}\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N_i} \sum_j \mathcal{L}(\theta, x_{ij})$

Parameters: gradient norm bound S_q, group size G, lattice vector size L, quantization step size Δ, noise scale N

1 Receive parameter set θ_t;
2 Take a random sample L_t with probability $\frac{L}{N_i}$;
3 Choose subset $D_t = \{x_{j \in L_t}\}$ from D_i using Poisson subsampling function $\mathcal{S}^{po}(.)$;
4 Compute gradient $\mathbf{g}_t(D_t, \theta_t)$;
5 Clip gradient $\bar{\mathbf{g}}_t \leftarrow \mathbf{g}_t / max(1, \frac{\|\mathbf{g}_t\|_2}{S_q})$;
6 Create Gaussian dither vector $\vec{\mathbf{d}}$ by drawing $\frac{N_i}{L}$ i.i.d samples from $\mathcal{N}(0, \sigma^2 C^2 \mathbf{I})$ with $\sigma = N\Delta$;
7 Split $\bar{\mathbf{g}}_t$ into $\frac{N_i}{L}$ lattice vectors $\{\bar{\mathbf{g}}_{t1}, ..., \bar{\mathbf{g}}_{tL}\}$;
8 Randomize lattice vector $\bar{\mathbf{g}}_{tj} = \bar{\mathbf{g}}_{tj} + \vec{\mathbf{d}}(j)$;
9 Quantize each lattice vector $\bar{\mathbf{g}}_{tqj} = \Delta \texttt{round}\left(\bar{\mathbf{g}}_{tqj} / \Delta\right)$;
10 Encode the concatenated quantized gradient vector $\bar{\mathbf{g}}_{tq}$ using universal source coding method `bzip`;
11 Send encoded gradient vector $\bar{\mathbf{g}}_{e_{ti}}$ and codebook CB_{t_i} to PS;

In round $t = 1$ the PS randomly initializes parameters set θ_0 and sends it to participants. We applying the quantization after the calculation of the gradients using the quantization noise to achieve a certain level of differential privacy. The descent on noisy gradients is performed by the PS.

To calculate the accumulative privacy costs over a total of T steps, first we need to fix the parameters of our noise mechanism. Consider the ideal quantization scheme of SDUVQ, in which the noise is independent of the source distribution and can be modelled as uniform noise. This will be equivalent of achieving $(0, \delta)$-DP. This special case is not desirable for us, since it will lead to very low levels of privacy, e.g. considering $S_q = 1$, setting $\Delta = 1/128$ will lead to $\delta = 64$. In NSDUVQ scheme, although the quantization noise can not be proven to be independent of the source distribution, it is shown that for $\sigma \geq \frac{\Delta}{2}$ it essentially follows a PQN model and behaves independent of the input distribution and can be modeled by the Gaussian dither. Furthermore, it is shown by Gariby and Erez (Gariby and Erez 2008) that the lattice quantization noise can approach a desired distribution, e.g. Gaussian distribution, with correct design of the lattice itself. We use this principle to shape the quantization noise to be Gaussian.

Fixing the values of ϵ, δ and Δ and clipping the gradients so that $S_q = 1$, we can then calculate the optimal value of σ according to the numeric algorithm of analytical Gaussian method (Balle and Wang 2018). Finally with the calculated σ, we can define the lattice parameters and the noise scale N of Algorithm 1 so that the quantization noise would be $\sim \mathcal{N}(0, \sigma^2 \mathbf{I})$. We use advanced composition theorem and privacy magnification through Poisson subsampling on our privacy bounds while calculating the system parameters.

Algorithm 2: Compressive (ϵ, δ)-DP Federated Learning - Parameter server (PS), Round t

Input: Encoded gradients of P participants $\{Bar\mathbf{g}_{e_{t1}}, ..., Bar\mathbf{g}_{e_{tP}}\}$ and codebooks $\{CB_{t_1}, ..., CB_{t_P}\}$

Parameters: PS learning rate η

1 Decode gradient vectors $\{Bar\mathbf{g}_{q_{t1}}, ..., Bar\mathbf{g}_{q_{tP}}\}$ using codebooks $\{CB_{t_1}, ..., CB_{t_P}\}$ for P clients;
2 Descend and calculate new model parameters $\theta_{t+1} = \theta_t - \eta \sum_{j=1}^{P} \bar{\mathbf{g}}_{q_{tj}}$;
3 Send updated model parameters θ_{t+1} to P clients for round t+1;

4 Experiments

4.1 Dataset and Experiment Design

We evaluate our method on EMNIST dataset. We simulate a FL setup with 10 participants. We experiment in severe and mild non-i.i.d data distribution cases. In severe non-i.i.d distribution, each participant only holds data related to one writer. In the mild non-i.i.d case, each client hold data related to 50 authors. To manage the trade-off between privacy and utility, we fix the privacy budget $\epsilon \leq 10$ and set the quantization system setting so that we would achieve the lowest σ value possible. This process has a trade-off in itself in form of quantization accuracy which directly affects the quantization noise while having an inverse effect on utility of the system in a fixed number of iterations. This also affects the compression level achieved through the quantization process.

4.2 Results

We applied the experiments above on a simple model with a dense layer of the size 10. For $\epsilon = 9.94$, $\delta = 0.001$, $\Delta = 1/64$, $L = 32$, $\eta = 0.01$, $T = 1000$, $G = 20$, $P = 10$, $N = 74$, $\sigma = 1.15625$ and with 4650 samples per client, we report a sparse categorical accuracy of 92.2% after $T = 1000$ iterations without early stopping, with the algorithm converging around $T = 650$. The achieved average compression ratio over the 1000 iterations for all participants is 2.97.

Based on the early results we have of non-concluded experiments, the privacy level, utility and compression ratio can certainly be massively improved by fine tuning the lattice, the quantization accuracy and the target privacy budget. Also, by experimenting with different architectures, we are able to see varying levels of success, depending on the architecture of the network and the training data.

5 Discussion

In this work, we explored the possibility of achieving differential privacy in a federated learning setting through quantization. We have presented an end to end compressive federated protocol and provided an analysis of the quantization noise. We integrated the quantization scheme with the federated learning setup and were able to achieve both compression and differential privacy in our collaborative setup. Although our experiments are still ongoing, early reported results support our claim of privacy and compression.

6 Acknowledgments

This research has been performed as part of the *Enabling Personalized Intervention* (EPI) project. The EPI project is funded by the Dutch Science Foundation in the Commit2Data program (Grant Number: 628.011.028).

References

Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H. B.; Mironov, I.; Talwar, K.; and Zhang, L. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 308–318.

Balle, B.; and Wang, Y.-X. 2018. Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. *arXiv preprint arXiv:1805.06530* .

Bebensee, B. 2019. Local differential privacy: a tutorial. *arXiv preprint arXiv:1907.11908* .

Chamikara, M. A. P.; Bertok, P.; Khalil, I.; Liu, D.; Camtepe, S.; and Atiquzzaman, M. 2019. Local differential privacy for deep learning. *arXiv preprint arXiv:1908.02997* .

Dragotti, P. L.; and Gastpar, M. 2009. *Distributed source coding: theory, algorithms and applications*. Academic Press.

Dwork, C.; Roth, A.; et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science* 9(3-4): 211–407.

Gariby, T.; and Erez, U. 2008. On general lattice quantization noise. In *2008 IEEE International Symposium on Information Theory*, 2717–2721. IEEE.

Geng, Q.; Ding, W.; Guo, R.; and Kumar, S. 2019. Optimal noise-adding mechanism in additive differential privacy. In *The 22nd International Conference on Artificial Intelligence and Statistics*, 11–20. PMLR.

Geng, Q.; and Viswanath, P. 2015. Optimal noise adding mechanisms for approximate differential privacy. *IEEE Transactions on Information Theory* 62(2): 952–969.

Gersho, A. 1979. Asymptotically optimal block quantization. *IEEE Transactions on information theory* 25(4): 373–380.

Gray, R. M.; and Neuhoff, D. L. 1998. Quantization. *IEEE transactions on information theory* 44(6): 2325–2383.

Kasiviswanathan, S. P.; Lee, H. K.; Nissim, K.; Raskhodnikova, S.; and Smith, A. 2011. What can we learn privately? *SIAM Journal on Computing* 40(3): 793–826.

Kruse, H.; and Mukherjee, A. 1999. Improving text compression ratios with the Burrows-Wheeler transform. In *Proceedings DCC'99 Data Compression Conference (Cat. No. PR00096)*, 536. IEEE.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, 1273–1282. PMLR.

Nasr, M.; Shokri, R.; and Houmansadr, A. 2018. Comprehensive privacy analysis of deep learning: Stand-alone and federated learning under passive and active white-box inference attacks. *arXiv preprint arXiv:1812.00910* .

Seward, J. 1996. bzip2 and libbzip2. *avaliable at http://www. bzip. org* .

Song, C.; Ristenpart, T.; and Shmatikov, V. 2017. Machine learning models that remember too much. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 587–601.

Widrow, B.; and Kollár, I. 2008. Quantization noise. *Cambridge University Press* 2: 5.

Zamir, R.; and Feder, M. 1992. On universal quantization by randomized uniform/lattice quantizers. *IEEE Transactions on Information Theory* 38(2): 428–436.

Zamir, R.; and Feder, M. 1996. On lattice quantization noise. *IEEE Transactions on Information Theory* 42(4): 1152–1159.

Zhao, J.; Wang, T.; Bai, T.; Lam, K.-Y.; Ren, X.; Yang, X.; Shi, S.; Liu, Y.; and Yu, H. 2019. Reviewing and improving the Gaussian mechanism for differential privacy. *arXiv preprint arXiv:1911.12060* .