Mixing Gradients in Neural Networks as a Strategy to Enhance Privacy in Federated Learning

Shaltiel Eloul, Fran Silavong, Sanket Kamthe, Antonios Georgiadis, and Sean J. Moran
CTO, JPMorgan Chase
shaltiel.eloul@jpmchase.com

Abstract

Federated learning reduces the risk of information leakage, but remains vulnerable to attack. We show that well-mixed *gradients provide numerical resistance to gradient inversion in neural networks. For example, we can enhance mixing gradients in a batch by choosing an appropriate loss function and drawing identical labels, and we support this with an approximate solution of batch inversion for linear layers. These simple architecture choices show no degradation to classification performance as opposed to noise perturbation defense. To accurately assess data recovery, we propose to use a variation distance metric for information leakage in images, derived from total variation. In contrast to Mean Squared Error or Structural Similarity Index metrics, it provides a continuous metric for information recovery. Finally, our empirical results of information recovery from various inversion attacks and training performance supports our defense strategies. These simple architecture choices found to be also useful for practical size of convolutional neural networks but depends on their size. We hope this work will trigger further defense studies using gradient mixing, towards achieving a trustful federation policy.*

1. Introduction

Federated learning (FL) is a privacy preserving technique that enables distributed nodes to contribute to the training of a machine learning model [19, 13]. The promises of FL are significant and have wide applicability in industry [2]. For example, it is possible for hospitals to collaborate on training a centralised model around the globe, without sharing or moving the actual private patient information across institutions [22, 26]. As it potentially protects sensitive data, it can better align with data protection regulations such as GDPR [6]. For example, FL has been already applied to prediction of treatment side effects in medicine [12] or for deployment in smartphones devices [3, 14, 15, 18]. Given the potential impact of FL, its privacy has been critically studied and challenged [21]. A standard FL configuration is typically achieved with a central aggregator node which exchanges gradients for training a centralised model. At each training step (t), a client node receives neural network model weights, $F(W_t)$, from an aggregator server and calculates loss (l) with a local data $\mathbf{x}_{t,b}, \mathbf{y}_{t,b}$ in a batch, B, which generates gradients with respect to the model weights:

$$\Delta W_t = -\frac{\gamma}{B} \sum_{b<B} \frac{\partial l(F(\mathbf{x}_{t,b}, \mathbf{y}_{t,b}))}{\partial W_t} \tag{1}$$

The gradients are typically averaged in the server with a rate, γ. The gradients, ΔW_t, shared by the client can expose the client to a potential inversion attack instigated by a malicious eavesdropper. The inversion attacks have shown to be surprisingly successful in many pioneer studies [36, 8, 38]. This compromised privacy prevents federated learning from becoming a fully trustful framework for distributed training. Whilst differential privacy is extensively used for FL defense, in this work, we identify conditions under which privacy can be attained with confidence against inversion attacks, without introducing noise or masking gradients by pruning, which often leads to a degradation on model accuracy [33, 30, 21]. Through our analysis, we demonstrate how *mixing of gradients* within the batch is an effective defense strategy to counteract gradient inversion attacks of the vulnerable dense layer and without degradation of training performance. In more detail, our contributions in this paper are:

- **Inversion of batch, label distribution and loss function:** we revisit the linear dense layer, but as opposed to previous works [38, 36, 1], here we discuss it in a batch. We show the direct inversion of a full and large batch without an optimisation-based gradient attack at all. In this context, we show the effect of mixing gradients in comparison to noise injection.
- **Strategy for better privacy:** resulting from the above study, we show empirical evaluation of how simple design changes to NN are useful against recovery attack.

In contrast to existing defense mechanisms, our training requires no noise, and show no performance degradation in commonly used benchmark tests.

- **Absolute Variation Distance:** Metrics such as mean squared error between the ground-truth and recovered data are inadequate for measuring partial leakage of information from noisy images. We use a metric which is a variant of total variation metric [27]. This metric is shown to be effective for evaluating a defense policy using information leakage in FL.

2. Related Work

Despite the compelling promises for privacy in FL, there is a body of work that present eavesdropping attacks on distributed machine learning systems to compromise data privacy [1, 8, 34, 38, 20, 29, 31], necessitating a better understanding on defense mechanisms to generate a trustful federation policy. Our study hence, is mostly related to attacks on exchange gradients or weights to communicate training of a neural network model. Early works studied techniques for extracting metadata about the private data, for example, membership attacks have been proposed in which a classifier is trained to identify whether a specific data-point has been used to train a model [29]. Property attacks are another attack variant in this direction where properties of a batch are exposed [20], such as the presence of a person in a photograph or its age. In both cases, actual data-points are not extracted from the gradient information.

Later studies, led by [38] and followed by many others, e.g. [36, 8, 34, 37], showed how it is possible to extract the actual data by inverting the gradients communicated by clients in a federation. For example, it was shown that it is possible to extract data at a pixel-level granularity with remarkable clarity [34, 25]. Bayes framework enables improving priors to various input data distributions in order to pass several defenses [1]. Specifically here we support our proposed defense to the well approximated Bayes form for image reconstruction [25, 1, 8, 23]. Generative models priors attacks, such pre-trained GAN based attacks are shown to provide realistic and accurate images by construction. For example, 'GIAS' attack [35] trains a generative model prior interactively, showing improvements for several image databases. Here we consider even minimal information recovery as a successful attack, and the accuracy improvements are not in the scope of this work. Hence in this study, we assessed mixing gradients defense strategy on the general benchmark attacks mentioned above.

Overall, less attention is dedicated for defense of those inversion attacks. However mechanisms to enhance privacy have been proposed with a range of effectiveness [23, 10], including gradient pruning, noise injection to the gradients [38, 33, 30] and blending methods of training images applied online by the client nodes [11, 5]. A very practical security protocol that minimise the risk of inversion is the secure aggregation protocols [4], for multi-party computing, where blending random vectors are added and subtracted between pair clients' gradients. This effective protocol is valid when large number of clients participating and sending valid gradients. Hence, exploring new privacy preserving approaches is still crucial for achieving trust-able FL setups. In this paper, we propose a defense strategy that utilises the properties of a batch in the client, to maximise gradient mixing. We analyse gradient mixing as a lightweight defense strategy for further counteracting gradient inversion attacks. The rest of the paper is structured as follows. We show for linear layers in a NN classifier, how several conditions in the architecture affect gradients mixing and data recovery. We investigate our approach using direct inversion for the batch and compare it to noise injection. Then, we describe an experimental framework for evaluating inversion attack by introducing the Absolute Variation Distance as a measure for successful attack. Lastly, we support the defense strategy of mixing gradients in the experimental results section, for linear-layers and typical CNN architecture with variable increasing size.

3. Gradient mixing as a batch defense strategy

The high vulnerability for recovering vector information from a dense linear layer is well known and described several time previously [36, 24, 8, 38], but here we show the direct inversion of gradients of information from the full batch, allowing to examine the conditions for well mixing as a defense, and compare it to noise injection.

3.1. Direct inversion of a full batch

We simplify our analysis of deep linear network to one hidden dense linear layer containing input and output vectors, $(\mathbf{x} = (o_1, o_2, ...o_n), \mathbf{o} = (o_1, o_2, ...o_C))$, where, $o_j = \sum_i^n w_{ij}x_i + b_j$. Note that it is enough to examine one dense linear layer as $\mathbf{x}$ can be inverted from a known o_j for any hidden linear layers using back-propagation. A typical classification architecture uses softmax, $p_k = \frac{e^{o_k}}{\sum_j e^{o_j}}$, followed by cross-entropy to obtain the loss:

$$l(p, y) = -\sum_k^C y_k \log p_k \tag{2}$$

where, C is the number of classes/categories. The derivative of p_k with respect to each o_j:

$$\frac{\partial p_k}{\partial o_j} = \begin{cases} p_k(1 - p_j), & k = j \\ -p_k p_j, & k \neq j. \end{cases} \tag{3}$$

where the set of the loss equations is then obtained:

$$\frac{\partial l}{\partial w_{i,j=k}} = \frac{\partial l}{\partial p_k}\frac{\partial p_k}{\partial o_j}x_i = (p_j - y_j)x_i \tag{4}$$

and:

$$\frac{\partial l}{\partial b_{j=k}} = p_j - y_j \tag{5}$$

Specifically, this is a case of a single input batch, $B = 1$, as also discussed by [24, 38]. The number of gradient equations are $nC + C$ with an extra C equations for weights (for each j) whilst the unknowns are $n + C$. For example x_i can be found immediately from any j, using $\frac{\partial l}{\partial w_{ij=k}} / \frac{\partial l}{\partial b_{j=k}} = x_i$. However, when the batch size is larger than one, the client would only share the averaged information:

$$\frac{\partial l^B}{\partial w^B_{i,j=k}} = \frac{1}{B} \sum_{m \in [1,B]} (p^m_j - y^m_j) x^m_i \tag{6}$$

$$\frac{\partial l^B}{\partial b^B_{j=k}} = \frac{1}{B} \sum_{m \in [1,B]} p^m_j - y^m_j \tag{7}$$

As no additional gradients are shared, the number of unknowns, $B(n + C)$ can exceed the gradients equations number $nC + C$, and there will be no unique solution to solve the set of equations. Even in the case that a unique solution exists, numerical optimisation can be challenging. However, in the scenario in which softmax is followed by cross entropy, we can show that an accurate direct solution is found in many cases even for $B \gg 1$, due to the de-mixing property across the batch. In an untrained, randomised weights model, the first order expected value of $\langle p^m_j - y^m_j \rangle$, is positive but close to zero for a non-target instance ($j \neq c$) and close to -1 for the instance target ($j = c$), where c is the target index. This is because we can show that the expected value, $\langle p_j(o_j) \rangle$ can be grossly estimated as $p_j(E(o_j))$ for the first order of Taylor expansion [7]. This results in $\langle p_j \rangle$ to be inversely proportional to the number of classes C. Subsequently, in a batch that contains unique labels, $c^m \neq c^{1 \dots B}$, Eq. 6-7 estimates:

$$\frac{\partial w^B_{i,j}}{\partial b^B_j} \approx \frac{(\langle p^m_{j=c} \rangle - 1) x^{m(j=c)}_i + \langle p^m_{j \neq c} \rangle \sum x^{m(j \neq c)}_i}{(\langle p^m_{j=c} \rangle - 1) + (B-1) \langle p^m_{j \neq c} \rangle}$$

$$\approx \frac{(\langle p^m_{j=c} \rangle - 1) x^{m(j=c)}_i}{(\langle p^m_{j=c} \rangle - 1)} = x^{m(j=c)}_i \tag{8}$$

This approximation shows that the de-mixing of the gradients for each vector enables a direct estimate of the input layer for any vector m in the batch once we pick $j = c$. The error of this estimation can be very low for large C. It is clearly seen here with the two most popular data-sets used for studying such inversion attacks, the MNIST with small number of equations C and LFW with large C ($C \gg B$).

Figure 1(a)-(c) shows our estimates to direct invert the input of a dense layer from a vector batch, and the ability to infer all inputs of a batch in a dense layer. It supports that inverting the 2-8 vectors of a batch is possible with a very

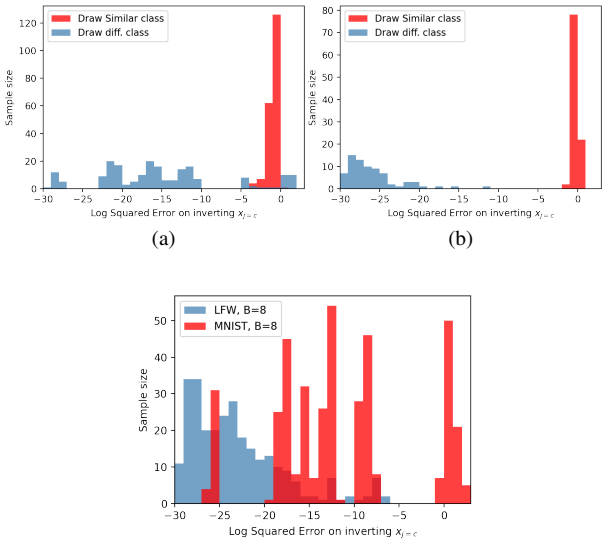


Figure 1: Log error on direct inversion for linear dense layer classification, calculate from Eq. (8) in 2 histograms for an untrained model. The case of drawing similar labels (red), and drawing unique labels in the batch (blue). (a) for MNIST (C=10), (b) for LFW dataset (C=5749), and (c) for batch size of 8 drawing unique labels, comparing LFW (blue) and MNIST (red).

small error, as long as labels are unique. In the case of the LFW dataset, due to the large C, the error is inversely proportional to C, so the two vectors in the batch are recovered for all random samples tested with a very low error. This low error is obtained so long as the vectors have a unique class, given $C \gg B$, which is the case for the LFW classification network with much larger batch sizes (e.g. $B \gg 8$).

The de-mixing property of cross-entropy is not only helpful for estimating input without numerical optimisation, but also allows simple numerical convergence as the set of equations to solve contains independent solution when a unique $x^{m(j=c)}_i$ contributes to the gradients, as inferred from Eq. (8). Yet, once we draw similar labels in the batch the recovery of data exhibits a large error, and the error is within the order of magnitude of the information. This mixing of gradients can serve as a strategy to increase privacy in FL against direct inversion without adding noise, and our empirical results later support that this strategy is also effective against numerical optimisation attacks.

Following this insight, we can also consider changing the objective function to mix the gradients. Instead of using cross-entropy loss (CEL), it is possible to use the mean squared loss (MSE), $l_2(o, y) = -\sum^C_k (y_k - o_k)^2$. This is not a typical choice for a classification task, but performance results show later that there an unnoticeable degradation in classification performance when using MSE in-

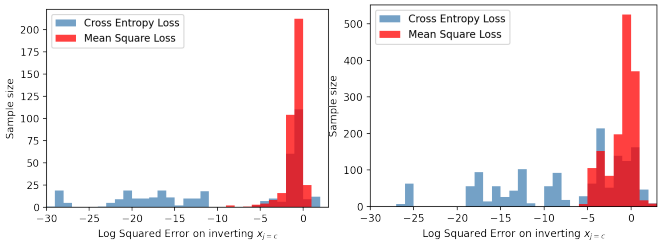
stead of CEL in typical benchmark setups used in inversion attacks studies. On the contrary, there is a large gain in privacy by the high mixing of gradients on the dense layer. The gradients are calculated on a dense layer using mean square estimation (for simplicity here, with no softmax):

$$\frac{\partial l_2}{\partial w_{i,j=k}} = \frac{\partial l_2}{\partial o_k}\frac{\partial o_k}{\partial w_i} = -2(o_j - y_j)x_i \tag{9}$$

and

$$\frac{\partial l_2}{\partial b_{j=k}} = -2(o_j - y_j) \tag{10}$$

Here o_j can take positive or negative values and generally within the order of y, for $j = c$ or $j \neq c$, hence the average of a batch will mix the gradients. Note that we can also use softmax, followed by MSE and the gradients will obtain a strong mixing, but with a less trivial gradient expression. We show in figure 2 a sample of results for the error on the direct inversion of a batch. We observe that the error on estimation of any vector using $\frac{\partial w}{\partial b}$ is not negligible, and sufficiently large even in batch size of 2 and more distinctive in batch size of 8.



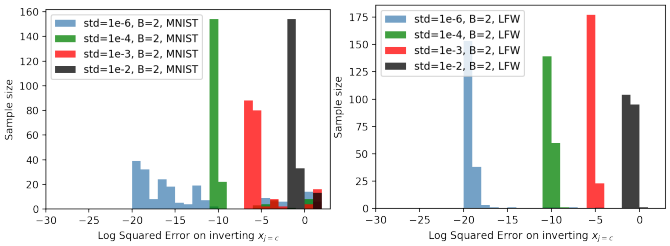
(a) (b)

Figure 2: The error on direct inversion of a linear dense layer in a batch, calculated from the approx. in Eq. (8) for two histograms of an untrained model. The case of MSE objective loss function (red histogram) compared to the CEL as generally used. (a) shows the result of batch size of 2, and (b) for batch size of 8.

We see here that potentially drawing input with similar labels and adjusting the loss function to increase mixing of gradients are strategies that counter the recovery of input from dense layers. In fact, we can compare our result to the effectiveness of injecting noise to gradients as a defense, given the wide application of differential privacy. We add a Gaussian noise term for gradients in a linear layer, $\frac{\partial l}{\partial w_{i,j}} + \zeta_{i,j}$, $\frac{\partial l}{\partial b_j} + \zeta_j$.

Figure 3 shows the results from addition of noise at various standard deviations. We find that small contamination of noise does not protect at all, against inversion and also the error is in the order of the noise, and given that, we are required the noise to be in the order of the weights ($std > 0.01$, when weights initialised uniformly between $(-0.5, 0.5)$). The addition of such a noise will affect the training drastically. In fact, this exercise shows that our gradient mixing strategies can be as effective as the addition of a large noise term, but without the loss of training performance as we show later.

Next, to further support mixing gradients strategy, we carry out an analysis of widely used, state of the art inversion attacks that uses numerical optimisation. We do this by firstly analysing inversion attack success in a dense layer model, and then show its validity for a typically explored CNN for inversion attack and the limitation of this strategy with increasing CNN size.



(a) (b)

Figure 3: The error from direct inversion by adding Gaussian noise to gradients and biases at various standard deviations. Histogram (a) for MNIST dataset and histogram (b) for LFW dataset.

4. Experimental framework

In this section we detail the experimental framework we use to evaluate our gradient de-mixing strategies. In our experiments we explore the privacy of the input data with two representative networks, the first, a single dense layer and a standard LeNET convolutional neural network (CNN) [16]. We analyse the impact of different loss functions and label distributions by varying the batch sizes and the number of filters. This empirical study explores the limit of the mixing gradient strategy for varying conditions, especially in larger convolutional network. For example, we demonstrate how it may help a practitioner to leverage a greater number of filters and a lower size of batch.

4.1. Inversion Attack Optimisation Algorithms

The gradient inversion attack is carried out by choosing $\mathbf{x}'_t, \mathbf{y}'_t$ on a proxy model, $F'(\mathbf{x}'_t, \mathbf{y}'_t)$, and finding $\Delta W'_t$ which minimises an objective function $M\Delta W_t$. A typical objective can be the norm of the gradients' difference:

$$g^{l2}(\mathbf{x}'_t, \mathbf{y}'_t) = \min||\Delta W'_t - \Delta W_t|| \tag{11}$$

This solution searches for a model $F'(\mathbf{x}'_t, \mathbf{y}'_t)$ that matches the size of the gradient vector observed by the client. Although further empirical studies have found the cosine dis-

Table 1: Types of gradient inversion attacks employed to evaluate our proposed defense strategy.

Attack Name	Main Objective Function	Description
2-norm	g^{l2} (11)	Euclidean distance and initial label determination [34].
Angle & var	g^{ang} + TV (12)	[8, 1] proposed to leverage cosine similarity, total variation (TV) and initial label determination.
Angle & var & Orth_regulators	g^{ang} + TV + Orth	Cosine distance with orthogonal regulator for the input + initial label determination [24, 34].

tance to provide better convergence results [8]:

$$g^{ang}(\mathbf{x}'_t, \mathbf{y}'_t) = \min 1 - \frac{\langle \Delta W'_t, \Delta W_t \rangle}{||\Delta W'_t|| \cdot ||\Delta W_t||} \tag{12}$$

Various regularisation terms were shown to improve convergence. For example, regularisation that penalises high variations in the input images and constrains the search to high-fidelity images and de-noised solutions [8, 34]. In batch, the orthogonality [24] between input vectors in the batch has been shown to bias the search towards different vectors in the batch. Additionally, it has been found that determining the label from the gradients is important for initialisation of the numerical optimisation [34]. We have also seen in the literature various type of attacks that provide improvements in image fidelity, or training convergence. Since no work so far is focused on enforcing the leakage of minimal information, here we apply various types of attacks and regularisation terms to provide a comprehensive analysis without any prior assumption on the performance of the attack. As summarised in Table 1, we utilise both the Euclidean distance and cosine similarity objective functions proposed by recent prior works [38, 8] with a selection of popular regularisation functions. We also determine labels from the negative gradients distributions prior to the optimisation as also previously suggested [36, 34]. However here we stress that this is only possible when $C \gg B$ but not necessarily true when B is comparable to C. This is due to the fact that p_j^m (see Eq. (4)) can obtain significant positive values for certain j when C is small, and can even result in average positive gradients when $j = C$. This for example occurs in the MNIST dataset where $C = 10$ but will not be observed in LFW with large C. In the case of the MSE loss, we also cannot initialise the labels distribution directly from the gradient sign, as o_j is proportional to y_j in Eq. (9) and Eq. (10). Hence, we determine the initial label distribution, but we still optimize the labels output using the optimizer scheme in order to maximise the recovery of an MNIST attack for a batch.

4.2. Criterion for Successful Attack

Many studies for improving attacks focus on fidelity of recovery and rate of convergence, e.g. [25, 24, 34]. Our focus is the opposite, to determine if any information regarding the data can be recovered. Therefore our criterion for a successful attack is minimal information recovered for one input vector in a batch that is distinguished from noise. The

(a)

(b)

Figure 4: Random recovered vectors pupolated in a table by their recovery rates from MNIST (a) and LFW (b) datasets, column-wise sorted via the abs. variation distance measure. The values were scaled by dividing AVD (Eq. 13) with the distance between the initial uniform noise input image, to a black image.

mean square error (MSE) and structural similarity index measure (SSIM) [32] are typically used in inversion attacks recovery of high fidelity and small changes between images. We found that these metrics are not reliable and cannot be used as indicator for information leakage in datasets such

as MNIST where the information is sparse. We can show this visible information in a random sample of recovered vectors from attacks in figure 4(a). The MSE indicator is not sufficient in the intermediate range where information is visible but noisy, blended, or there are other patterns that can significantly skew the results. A more suitable indicator is to compare the spatial gradient of the recovered image and source image:

$$\mathrm{AVD}(v^{source}, v^{target}) = ||(|\nabla v^{source}_{x,y}| - |\nabla v^{target}_{x,y}|)|| \tag{13}$$

where $\nabla v = \frac{dv}{dx} + \frac{dv}{dy}$ is the pixel-wise gradient. The variation distance metric allows to consider boundaries and edges in images which are a common discriminator in visual recognition, whilst the gradient of noise remains as noise. A random sample of attack results after 550 iterations of optimisation are shown in figure 4. A qualitative assessment of the results shows that recovery of data is more visible as AVD decreases in a continuous manner. In contrast the MSE metric for MNIST fluctuates drastically when the image is not completely clear or a blend, and can obtain various values similar or higher than the MSE for the pure noise input. Using this qualitative observation, we can define a threshold range between 0.6, where numbers are starting to emerge, and beyond 0.8, where information, at least for human eyes, is not visible. For the LFW dataset (figure 4), we see the same trend of good correlation between AVD and the revealed information, but the MSE metric is also a reasonable indicator as the images contain highly dense information. We use a similar range of threshold of 0.6, and 0.8 to indicate a successful recovery in our experiments. [1]

4.3. Datasets and Attack Experiments

We conduct gradient inversion attack experiments on two representative datasets, MNIST Handwritten Digit [17] and Labelled Faces in the Wild (LFW) [9], to illustrate how our proposed defense strategies successfully minimise information leakage without performance degradation. These two dataset are commonly used among researchers to study attacks [38, 36, 20, 29]. For each experiment we carried out 10-20 trials for each of the 3 attacks presented in table 1. We analyse the recovery rate, which is the percentage of trials that lead to successful recovery. A successful recovery is determined by a threshold for the AVD metric (in MNIST) or MSE metric (in LFW) at the end of the every trial. In terms of the optimisation scheme, we utilized the standard optimisation scheme, LFBGS, with learning rate (lr) of 0.05 and 550 iterations for running a proxy model to attack. We also carried out complementary tests with 1200 iterations, and lr of 0.025 to further showcase the validity of our results. The analysis to support our results is available in the supporting information.

To evaluate the performance of the neural networks we trained the LeNET and dense layer models using an SGD optimiser on the MNIST dataset for 60 epochs. We will release the source code to reproduce these results upon acceptance of the paper.

5. Experimental Results

5.1. Single Dense Layer

A single layer attack is a valuable experiment to clearly demonstrate our strategy in single or multi-linear-layers model, such as in logistic regressions or NN models. As we have shown earlier we can recover the data directly from a batch without numerical optimisation. Here we explore the results from an optimiser attacks on a simplified single layer. In this case, we look at the MNIST dataset, with a linear regression that can provide a practical model for prediction. We explore different batch sizes, B=$1, 2, 4, 6$ using the AVD metric with two thresholds to determine a successful recovery. We calculate recovery rates for each experiment strategy, MSE vs. CEL as loss function, both followed by softmax and random drawing of labels vs. equal labels in a batch.

Figure 5 shows that MSE and a batch of equal labels provides very low recovery rates for batches of size 4 and 6. This result is observed for both threshold values 0.6 and 0.8. The additional value of using MSE with equal labeling is minor. The regular approach of using random data in the batch with CEL is observed to be the most vulnerable. The importance of these results can be justified by looking at the training performance of these networks as presented in figure 6. It is shown that performance has in fact remained intact, even for MSE as an objective function followed by softmax, which is not typically applied in classification tasks, and also for equal labels despite the possible diversification issue that this may raise [2]. We note that in the aggregation of a central model, we update the model only after aggregation of the gradients from clients, so the diversification of labels in the batch of equal labels happens naturally. It therefore enables similar performance to the random label. We also observe that the MSE loss without softmax results in lower performance for a single dense layer. However we obtained the opposite behaviour for LeNET as we show next, so this discrepancy may be addressed by further optimisation through tuning the network hyper-parameters.

[1] We note that using the variation by measuring the entropy, also yields in a very compatible metric, but was not used in this study. In that case, the relative information can be measured as:

$$\Delta S_{av} = -p_0 \log |\nabla v^{source}_{x,y}|/|\nabla v^{target}_{x,y}| \tag{14}$$

Here p_0 is used here to be the expected value of the initial input vector to the attack, which in our case is uniform noise (0,1), so $p_0 = 0.5$.

[2] Hyper-parameters were not optimised and similar to all configurations.

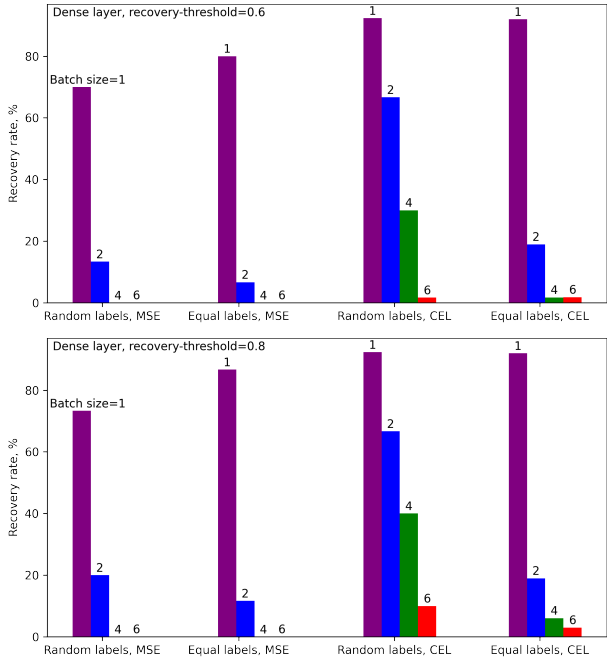


Figure 5: Recovery rates (in percents) for MNIST dataset using one linear dense layer. We preset the rate of recovery using the AVD metric with threshold 0.6 (a), and 0.8 (b). The bars show different mini-batch sizes, and the x-axis differentiates between MSE loss function, CEL function and the cases of drawing random and equal labels in CEL function.

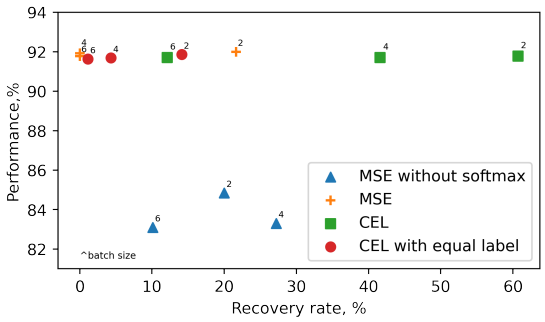


Figure 6: Performance of MNIST in single net architecture sizes at various strategies and at batch sizes 2, 4 and 6. The y-axis represents classification accuracy and the x-axis shows fraction of recovered images. The figure shows that recovery rate is high for cross entropy, and either mixing the labels or using a different loss function can reduce recovery rate without sacrificing the classification accuracy.

5.2. Convolutional Neural Network

Results for the linear layer show the clear advantage of maximising gradient mixing. Here we explore how this affects attack success rate in a widely used convolutional neural network, LeNET [16] for image recognition and widely used for testing inversion attacks [28, 36, 38, 8]. We show a 2D map of the recovery rates for attack experiments on different batch sizes and increasing number of channels to explore larger CNNs (up to double the filter size of LeNETs). The results are presented in figure 7 for the MNIST and the LFW dataset. The maps also show the boundaries for zero recovery rates.

The trends in figure 7 show that gradient mixing overall provides a useful decrease in recovery rates that permit a practitioner to deploy a larger model with data protection supported by the mixing of gradients. However it is shown that with a much larger number of filters, the data can be recovered by the convolutional layers due to large number of equations compare to unknowns with increasing number of filters. Hence defenses applied on wider models such as large Residual neural networks (a ResNet with 64-128 channels), will be more vulnerable. In general, the network size effect has a similar challenge for all defense strategies [24, 10]. In a further study, we intend to explore the combination of mixing gradients and injecting minimal noise as a combined strategy for enhancing privacy in large models.

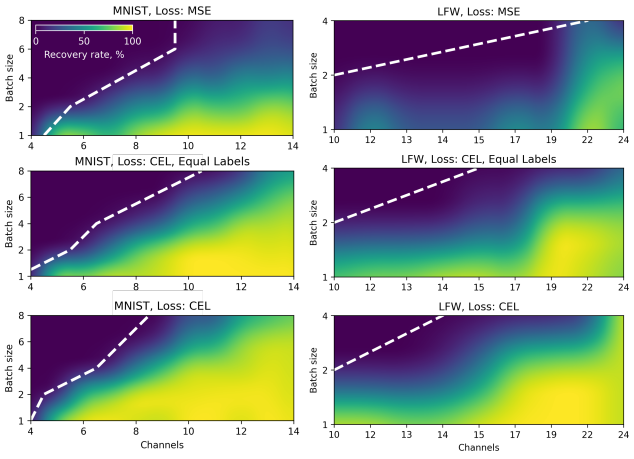


Figure 7: Analysis of recovery attack success rate in MNIST and LFW with CNN model, LeNET architecture. The white dashed line represents the boundary of zero rate success recovery.

Finally, we carry out a performance test for LeNET in each strategy and with varying batch sizes and number of channels for the MNIST dataset. We show the comparison of the network performance against the recovery rates in figure 8. Results show that performance is also kept relatively intact allowing a clear benefit of privacy protection in comparison to the typical CEL and random label selection. Interestingly, in contrast to the single dense layer case, here MSE with softmax performs less well than other networks. Our results for CNN show the benefit of choosing strategies for mixing gradients as, in many cases, the maximum batch

size that can be used is limited (e.g. in distributed training over clients with sparse data).

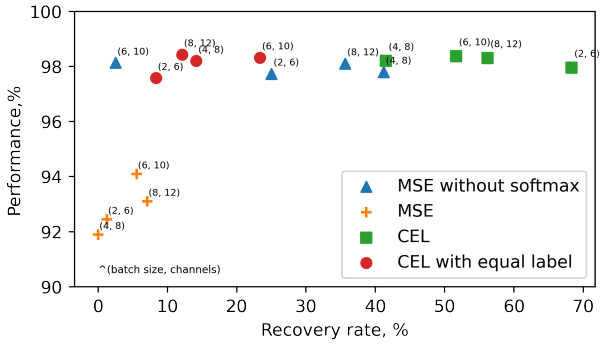


Figure 8: Performance of MNIST dataset on LeNET network architecture with different channels and batch sizes and for various gradient mixing strategies.

6. Conclusions

We have shown that by simple architecture choices, we can prevent the recovery of data from widely used gradient inversion attacks. The choice of loss function and the drawing of equal labels in a batch results in mixing of the gradients in practical neural networks architectures. In fact, without mixing gradients, it is possible to recover directly all batch vectors due to the de-mixing nature of cross entropy loss function. Our suggested strategies for mixing gradients maintain network performance in certain setups, which is in contrast to common methods that apply noise to the gradients. Additionally, in practice, one could combine the mixed gradients strategies further with noise or other defense methods for better privacy. Finally, we have shown that an absolute variation distance (AVD) metric is able to measure the relative information recovered by gradient inversion attacks. The metric, which is derived from total variation, can distinguish information from noise for datasets that have sparse information such as in the MNIST dataset and will be explored further in future studies. We hope that this work prompts the development of new strategies towards achieving more trustful federated learning platforms. Further work will also study the effect of more complex architectures and larger models which are more challenging area of privacy preserving in distributed learning.

7. Acknowledgment & Disclaimer

This paper was prepared for informational purposes by the Applied Innovation of AI team and the Global Technology Applied Research Center of JPMorgan Chase & Co. This paper is not a product of the Research Department of JPMorgan Chase & Co. or its affiliates. Neither JPMorgan Chase & Co. nor any of its affiliates makes any explicit or implied representation or warranty and none of them accept any liability in connection with this paper, including, without limitation, with respect to the completeness, accuracy, or reliability of the information contained herein and the potential legal, compliance, tax, or accounting effects thereof. This document is not intended as investment research or investment advice, or as a recommendation, offer, or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction.

References

[1] Mislav Balunović, Dimitar I Dimitrov, Robin Staab, and Martin Vechev. Bayesian framework for gradient leakage. *arXiv preprint arXiv:2111.04706*, 2021.

[2] Syreen Banabilah, Moayad Aloqaily, Eitaa Alsayed, Nida Malik, and Yaser Jararweh. Federated learning review: Fundamentals, enabling technologies, and future applications. *Information Processing & Management*, 59(6):103061, 2022.

[3] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloé Kiddon, Jakub Konečný, Stefano Mazzocchi, Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. Towards federated learning at scale: System design. In *Proceedings of Machine Learning and Systems*, volume 1, pages 374–388, 2019.

[4] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191, 2017.

[5] Nicholas Carlini, Samuel Deng, Sanjam Garg, Somesh Jha, Saeed Mahloujifar, Mohammad Mahmoody, Shuang Song, Abhradeep Thakurta, and Florian Tramèr. An attack on instahide: Is private learning possible with instance encoding? *CoRR*, abs/2011.05315, 2020.

[6] European Commission. 2018 reform of eu data protection rules. In *General Data Protection Regulation*, 2018.

[7] Jean Daunizeau. Semi-analytical Approximations to Statistical Moments of Sigmoid and Softmax Mappings of Normal Variables, 2017.

[8] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting Gradients - How easy is it to break privacy in federated learning? In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[9] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[10] Yangsibo Huang, Samyak Gupta, Zhao Song, Kai Li, and Sanjeev Arora. Evaluating gradient inversion attacks and defenses in federated learning. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[11] Yangsibo Huang, Zhao Song, Kai Li, and Sanjeev Arora. InstaHide: Instance-hiding schemes for private distributed learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4507–4518. PMLR, 13–18 Jul 2020.

[12] Arthur Jochems, Timo M. Deist, Johan van Soest, Michael J. Eble, Paul Bulens, Philippe Coucke, Wim J F Dries, Philippe Lambin, and Andre Dekker. Distributed learning: Developing a predictive model based on data from multiple hospitals without data leaving the hospital - a real life proof of concept. *Radiotherapy and oncology : journal of the European Society for Therapeutic Radiology and Oncology*, 121 3:459–467, 2016.

[13] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, K. A. Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G.L. D'Oliveira, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and Open Problems in Federated Learning. *Foundations and Trends® in Machine Learning*, 14:1–210, 2021.

[14] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtarik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. In *NIPS Workshop on Private Multi-Party Machine Learning*, 2016.

[15] Fan Lai, Yinwei Dai, Xiangfeng Zhu, and Mosharaf Chowdhury. Fedscale: Benchmarking model and system performance of federated learning. *CoRR*, abs/2105.11367, 2021.

[16] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, R. Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten Digit Recognition with a Back-Propagation Network. In *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann, 1990.

[17] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[18] Bing Luo, Xiang Li, Shiqiang Wang, Jianwei Huang, and Leandros Tassiulas. Cost-effective federated learning in mobile edge networks. *CoRR*, abs/2109.05411, 2021.

[19] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR, 20–22 Apr 2017.

[20] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, pages 691–706. IEEE, 2019.

[21] Viraaji Mothukuri, Reza M Parizi, Seyedamin Pouriyeh, Yan Huang, Ali Dehghantanha, and Gautam Srivastava. A survey on security and privacy of federated learning. *Future Generation Computer Systems*, 115:619–640, 2021.

[22] Dinh C Nguyen, Quoc-Viet Pham, Pubudu N Pathirana, Ming Ding, Aruna Seneviratne, Zihuai Lin, Octavia Dobre, and Won-Joo Hwang. Federated learning for smart healthcare: A survey. *ACM Computing Surveys (CSUR)*, 55(3):1–37, 2022.

[23] Attia Qammar, Jianguo Ding, and Huansheng Ning. Federated learning attack surface: taxonomy, cyber defences, challenges, and future directions. *Artificial Intelligence Review*, 55(5):3569–3606, 2022.

[24] Jia Qian, Hiba Nassar, and Lars Kai Hansen. Minimal model structure analysis for input reconstruction in federated learning, 2021.

[25] Hanchi Ren, Jingjing Deng, and Xianghua Xie. Grnn: generative regression neural network—a data leakage attack for federated learning. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(4):1–24, 2022.

[26] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletarì, Holger R. Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N. Galtier, Bennett A. Landman, Klaus Maier-Hein, Sébastien Ourselin, Micah Sheller, Ronald M. Summers, Andrew Trask, Daguang Xu, Maximilian Baust, and M. Jorge Cardoso. The future of Digital Health with Federated Learning. *npj Digital Medicine*, 3(1), Dec. 2020.

[27] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. In *Proceedings of the Eleventh Annual International Conference of the Center for Nonlinear Studies on Experimental Mathematics: Computational Issues in Nonlinear Science: Computational Issues in Nonlinear Science*, page 259–268, USA, 1992. Elsevier North-Holland, Inc.

[28] Virat Shejwalkar, Amir Houmansadr, Peter Kairouz, and Daniel Ramage. Back to the drawing board: A critical evaluation of poisoning attacks on production federated learning. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1354–1371. IEEE, 2022.

[29] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pages 3–18. IEEE Computer Society, 2017.

[30] Florian Tramèr and Dan Boneh. Differentially Private Learning Needs Better Features (or Much More Data). In *International Conference on Learning Representations (ICLR)*, 2021.

[31] Binghui Wang and Neil Zhenqiang Gong. Stealing hyperparameters in machine learning. In *2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings, 21-23 May 2018, San Francisco, California, USA*, pages 36–52. IEEE Computer Society, 2018.

[32] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[33] Wenqi Wei, Ling Liu, Margaret Loper, Ka-Ho Chow, Mehmet Emre Gursoy, Stacey Truex, and Yanzhao Wu. A Framework for Evaluating Gradient Leakage Attacks in Federated Learning, 2020.

[34] Hongxu Yin, Arun Mallya, Arash Vahdat, Jose Alvarez, Jan Kautz, and Pavlo Molchanov. See through Gradients: Image Batch Recovery via GradInversion. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16332–16341, 2021.

[35] Xianglong Zhang and Xinjian Luo. Exploiting defenses against gan-based feature inference attacks in federated learning. *arXiv preprint arXiv:2004.12571*, 2020.

[36] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. idlg: Improved deep leakage from gradients, 2020.

[37] Junyi Zhu and Matthew B. Blaschko. R-GAP: recursive gradient attack on privacy. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[38] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.