RECONSTRUCTING TRAINING DATA FROM MULTICLASS NEURAL NETWORKS

Gon Buzaglo[1]*, **Niv Haim**[1]*, **Gilad Yehudai**[1], **Gal Vardi**[2], **and Michal Irani**[1]
[1]Weizmann Institute of Science, Israel
[2]TTI Chicago and the Hebrew University of Jerusalem

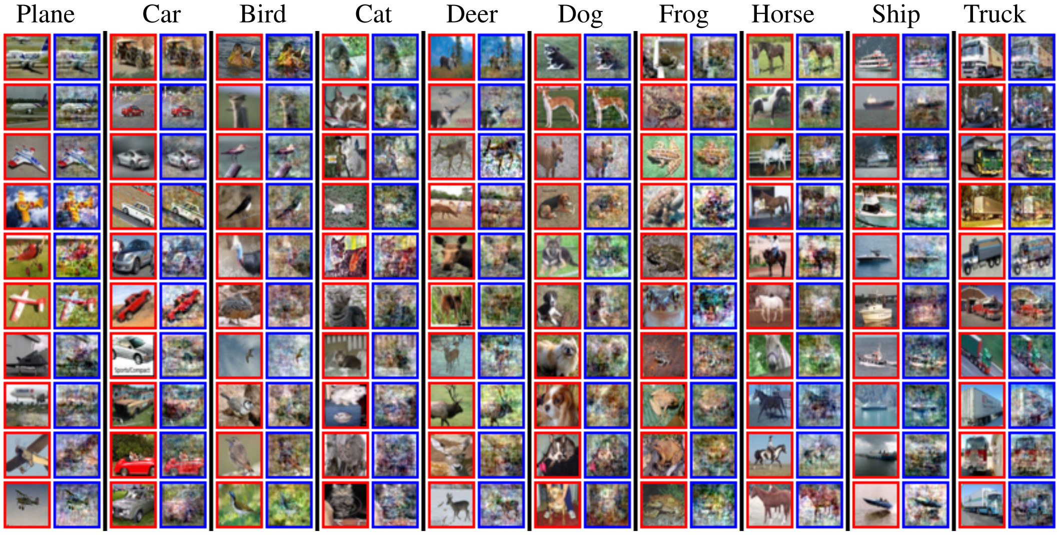


Figure 1: Reconstructed training samples from a multi-class MLP classifier that was trained on 500 CIFAR10 images. Each column corresponds to one class and shows the 10 training samples (*red*) that were best reconstructed from this class, along with their reconstructed result (*blue*).

ABSTRACT

Reconstructing samples from the training set of trained neural networks is a major privacy concern. Haim et al. (2022) recently showed that it is possible to reconstruct training samples from neural network binary classifiers, based on theoretical results about the implicit bias of gradient methods. In this work, we present several improvements and new insights over this previous work. As our main improvement, we show that training-data reconstruction is possible in the multi-class setting and that the reconstruction quality is even higher than in the case of binary classification. Moreover, we show that using weight-decay during training increases the vulnerability to sample reconstruction. Finally, while in the previous work the training set was of size at most 1000 from 10 classes, we show preliminary evidence of the ability to reconstruct from a model trained on 5000 samples from 100 classes.

1 INTRODUCTION

Understanding memorization in data-driven machine learning models is a fundamental question with implications on explainability, privacy, artistic synthesis and more. Haim et al. (2022) recently demonstrated that a large portion of the training samples are encoded in the parameters of trained neural network binary classifiers, by explicitly reconstructing samples from the training set of such

[*] Equal Contribution

models. The reconstruction method is based on consequences of the implicit bias of gradient descent, as presented by Lyu & Li (2019); Ji & Telgarsky (2020) – a homogeneous neural network trained with gradient descent will converge (in direction) to a solution of the KKT conditions of a maximum-margin problem. This dictates a set of equations that relates the parameters of the trained network and the training data. The key observation is that given a trained model (and its parameters), these relations can be leveraged to reconstruct training samples. Thus, a loss function is devised to show that by changing the inputs to the classifier (in order to minimize the loss), the inputs converge to true samples from the original training set.

The work of Haim et al. (2022) had several limitations, namely: (1) Their work only showed reconstructions from binary classifiers; (2) For their reconstruction method to succeed the trained network needed to be initialized with very small weights, smaller than standard Xavier initialization (Glorot & Bengio, 2010); and (3) Their experiments consisted of only small datasets with at most 1000 samples. In this work we extend their results in several directions, and overcome some of the limitations while gaining new insights on this reconstruction method.

Our contributions: (1) We show reconstruction of large portions of actual training samples from a trained *multi-class* neural networks; (2) We show that the use of weight-decay enables reconstruction of samples from models trained with standard initialization schemes thus overcoming a major limitation of Haim et al. (2022); (3) We show reconstruction of models trained on datasets that are 10x larger than shown in Haim et al. (2022); (4) We empirically analyze the effect of weight-decay on sample reconstruction, showing that it increases the vulnerability to such attacks.

Related works. Several works have shown different privacy attacks in deep learning architectures, which aim to leak private information from trained models. For example, *Model inversion* attacks aim at reconstructing class representatives Fredrikson et al. (2015); He et al. (2015); Yang et al. (2019). Other types of attacks target specific models, such as extracting data from *language models* Carlini et al. (2019; 2021), which use crafted prompts; and information leakage from collaborative deep learning (federated learning) He et al. (2019); Melis et al. (2019); Huang et al. (2021); Hitaj et al. (2017). In Balle et al. (2022) a reconstruction attack is shown where the attacker knows all the training samples except for one. Recently, Carlini et al. (2023) showed extraction of actual training images from trained diffusion models. Their methods rely on generating many different images and using known membership inference attacks to determine which generated image was used as a training sample. We emphasize that their method is specific for generative models, whereas we focus on classifiers.

2 Preliminaries - Implicit Bias of Gradient Methods

Neural networks are commonly trained using gradient methods, and when large enough, they are expected to fit the training data well. However, it is empirically known that these models converge to solutions that also generalize well to unseen data, despite the risk of overfitting. Several works pointed to the "*implicit bias*" of gradient methods as a possible explanation. One of the most prominent results in this area is by Soudry et al. (2018), who showed that linear classifiers trained with gradient descent on the logistic loss converge to the same solution as that of a hard-SVM, meaning that they maximize the margins. This result was later extended to non-linear and homogeneous neural networks by Lyu & Li (2019); Ji & Telgarsky (2020). Based on these results, Haim et al. (2022) have devised a data reconstruction scheme from trained binary classifiers (see Section 3 in Haim et al. (2022)). Below we describe an extension of the theorem about the implicit bias of homogeneous neural networks to a multi-class setup, based on theoretical results from Appendix G in Lyu & Li (2019).

Formally, let $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^n \subseteq \mathbb{R}^d \times [C]$ be a multi-class classification training set where $C \in \mathbb{N}$ is any number of classes, and $[C] = \{1, \dots, C\}$. Let $\Phi(\boldsymbol{\theta}; \cdot) : \mathbb{R}^d \to \mathbb{R}^C$ be a neural network parameterized by $\boldsymbol{\theta} \in \mathbb{R}^p$. We denote the j-th output of Φ on an input $\mathbf{x}$ as $\Phi_j(\boldsymbol{\theta}; \mathbf{x}) \in \mathbb{R}$. Consider a homogeneous network, minimizing the standard cross-entropy loss and assume that after some number of iterations the model correctly classifies all the training examples. Then, gradient flow will converge to a KKT point of the following maximum-margin problem:

$$\min_{\boldsymbol{\theta}} \frac{1}{2} \|\boldsymbol{\theta}\|^2 \quad \text{s.t.} \quad \Phi_{y_i}(\boldsymbol{\theta}; \mathbf{x}_i) - \Phi_j(\boldsymbol{\theta}; \mathbf{x}_i) \geq 1 \quad \forall i \in [n], \forall j \in [C] \setminus \{y_i\} \quad . \tag{1}$$

This KKT point is characterized by the following set of conditions:

$$\boldsymbol{\theta} - \sum_{i=1}^{n} \sum_{j \neq y_i}^{c} \lambda_{i,j} \nabla_{\boldsymbol{\theta}} (\Phi_{y_i}(\boldsymbol{\theta}; \mathbf{x}_i) - \Phi_j(\boldsymbol{\theta}; \mathbf{x}_i)) = \mathbf{0} \tag{2}$$

$$\forall i \in [n], \forall j \in [C] \setminus \{y_i\} : \quad \Phi_{y_i}(\boldsymbol{\theta}; \mathbf{x}_i) - \Phi_j(\boldsymbol{\theta}; \mathbf{x}_i) \geq 1 \tag{3}$$

$$\forall i \in [n], \forall j \in [C] \setminus \{y_i\} : \quad \lambda_{i,j} \geq 0 \tag{4}$$

$$\forall i \in [n], \forall j \in [C] \setminus \{y_i\} : \quad \lambda_{i,j} = 0 \text{ if } \Phi_{y_i}(\boldsymbol{\theta}; \mathbf{x}_i) - \Phi_j(\boldsymbol{\theta}; \mathbf{x}_i) \neq 1 \tag{5}$$

3 Multi-Class Reconstruction

We use a similar reconstruction method to Haim et al. (2022). Suppose we are given a trained classifier with parameters $\boldsymbol{\theta}$, our goal is to find the set of data samples $\{\mathbf{x}_i\}_{i=1}^n$ that the network trained on. A straightforward approach for such a loss would be to minimize the norm of the L.H.S of condition eq. (2). That is, we initialize $\{\mathbf{x}_i\}_{i=1}^m$ and $\{\lambda_{i,j}\}_{i \in [n], j \in [C] \setminus y_i}$ where m is a hyperparameter.

Note that from eqs. (3) and (5), most $\lambda_{i,j}$ zero out: the distance of a sample $\mathbf{x}_i$ to its nearest decision boundary, $\Phi_{y_i} - \max_{j \neq y_i} \Phi_j$, is usually achieved for a single class j and therefore (from eq. (5)) in this case at most one $\lambda_{i,j}$ will be non-zero. For some samples $\mathbf{x}_i$ it is also possible that all $\lambda_{i,j}$ will vanish. We therefore turn to only optimizing on the distance from the decision boundary. This implicitly includes eq. (5) into the summation in eq. (2), dramatically reducing the number of summands, and simplifying the overall optimization problem. We define the following loss:

$$L_{st}(\mathbf{x}_1, ..., \mathbf{x}_m, \lambda_1, ..., \lambda_m) = \left\| \boldsymbol{\theta} - \sum_{i=1}^{m} \lambda_i \ \nabla_{\boldsymbol{\theta}} [\Phi_{y_i}(\mathbf{x}_i; \boldsymbol{\theta}) - \max_{j \neq y_i} \Phi_j(\mathbf{x}_i; \boldsymbol{\theta})] \right\|_2^2 \tag{6}$$

Our reconstruction method is, given the parameters of a trained network $\boldsymbol{\theta}$, initialize $\mathbf{x}_i$ and λ_i for $i = 1, \ldots, m$, and minimize equation 6. In order to make sure that eq. (4) is satisfied we optimize for a_i and require that $\lambda_i = a_i^2$[1]. This further simplifies the optimization problem compared to Haim et al. (2022) that use a separate loss function.

Since n is unknown we set $m \geq n$ which represents the number of samples we want to reconstruct (thus, we only need to upper bound n). We can hypothetically set $m = C \cdot n$ and with balanced labels, this way there are enough reconstructed samples for any distribution of the labels. In practice, we set m to be slightly larger than n, which is enough to get good reconstructions. The rest of the hyperparameters (including $\lambda_{\min}$) are chosen using a hyper-parameter search.

4 Results

4.1 Multiclass Reconstruction

We compare between reconstruction from binary classifiers (as studied in Haim et al. (2022)) and multi-class classifiers. We conduct the following experiment: we train an MLP classifier with architecture D-1000-1000-C on 500 samples from the CIFAR10 (Krizhevsky et al., 2009) dataset. We use full-batch GD, and set the learning rate as 0.5. The model is trained to minimize the cross-entropy loss with full-batch gradient descent, once with two classes (250 samples per class) and once for the full 10 classes (50 samples per class). The test set accuracy of the models is 77%/32% respectively, which is far from random (50%/10% resp.).

To quantify the quality of our reconstructed samples, for each sample in the original training set we search for its nearest neighbour in the reconstructed images and measure the similarity using

[1] More precisely, $\lambda_i = a_i^2 + \lambda_{\text{min}}$, where λ_{min} is a hyperparameter that encourages the reconstructed samples to converge to margin-samples.

SSIM (Wang et al., 2004) (higher SSIM means better reconstruction). As a rule of thumb, we say that a sample was reconstructed well if its SSIM> 0.4 (see discussion in appendix A). In fig. 2 we plot the quality of reconstruction (in terms of SSIM) against the distance of the sample from the decision boundary $\Phi_{y_i}(\mathbf{x}_i; \boldsymbol{\theta}) - \max_{j \neq y_i} \Phi_j(\mathbf{x}_i; \boldsymbol{\theta})$. As seen, a multi-class classifier yields much more samples that are vulnerable to being reconstructed.

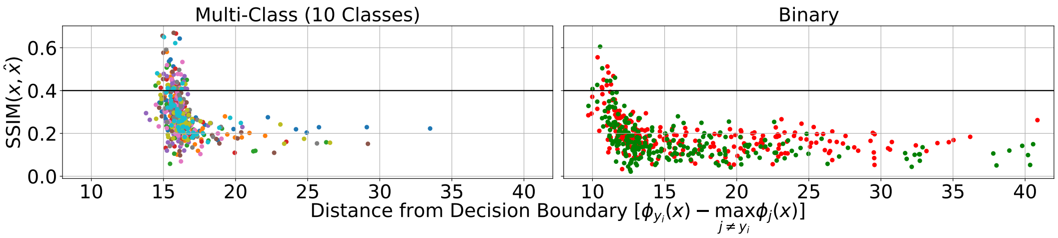


Figure 2: Multi-class classifiers are more vulnerable to training-set reconstruction. For a training set of size 500, a multi-class model (*left*) yields much more reconstructed samples with good quality (SSIM> 0.4), than a binary classification model (*right*).

Next, we examine the dependence between the ability to reconstruct from a model and the number of classes on which it was trained. Comparing between two models trained on different number of classes is not immediately clear, since we want to isolate the effect of the number of classes from the size of the dataset (it was observed by Haim et al. (2022) that the number of reconstructed samples decreases as the total size of the training set increases). We therefore train models on training sets with varying number of classes ($C \in \{2, 3, 4, 5, 10\}$) and varying number of samples per class ($1, 5, 10, 50$). The results are visualized in fig. 3a. Note that for models with same number of samples per class, the ability to reconstruct *increases* with the number of classes, even though the total size of the training set is larger. This further validates our hypothesis that multi-class models are more vulnerable to reconstruction. We continue this study in appendix B.

Another way to validate this hypothesis is by showing the dependency between the number of classes and the number of "good" reconstructions (SSIM> 0.4) – shown in fig. 3b. As can be seen, training on multiple classes yields more samples that are vulnerable to reconstruction. A possible intuitive explanation, is that multi-class classifiers have more "margin" samples. Since margin-samples are more vulnerable to reconstruction, this results in more samples being reconstructed from the model.

4.2 Weight Decay Increases Reconstructability

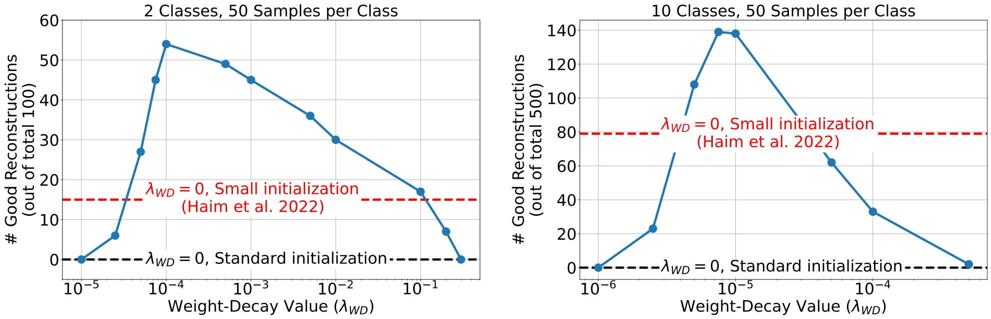


Figure 4: Using weight-decay during training increases vulnerability to sample reconstruction

Another drawback of Haim et al. (2022) is that reconstruction was only shown for models whose first fully-connected layer was initialized with small (non-standard) weights. Models with standard initialization, such as Kaiming He et al. (2015) or Xavier Glorot & Bengio (2010) where each weight vector is initialized with a variance of $\sim \frac{1}{d}$ (where d is the input's layer dimension) did not yield good reconstructed samples. Haim et al. (2022) only reconstructed from networks initialized with a variance of $\sim \frac{1}{d^{1.5}}$. Set to better understand this drawback, we observed an interesting phenomenon – for models with standard initialization, using weight-decay during training enabled samples reconstruction. Moreover, for some values of weight-decay the reconstructability is *significantly higher* than what was observed for models with small-initialized-first-layer models. In fig. 4 we show the number of good reconstructed samples for different choices of the value of the weight decay (λ_{WD}). We show results for two models trained on $C = 2$ classes (fig. 4*left*) and $C = 10$ classes (fig. 4*right*), both trained on 50 samples per class. We add two baselines trained without weight-decay: model trained with standard initialization (*black*) and model with small-initialized-first-layer (*red*).

By looking at the exact distribution of reconstruction quality to the distance from the margin, we observe that weight-decay (for some values) results in more training samples being on the margin of the trained classifier, thus being more vulnerable to reconstruction. This observation is shown in fig. 5 where we show the plots for all experiments from fig. 4*left*. We also provide the train and test errors for each model. It seems that the generalization (test error) does not change significantly. However, an interesting observation is that reconstruction is possible even for models with non-zero training errors (for which, the assumptions of Lyu & Li (2019) do not hold).

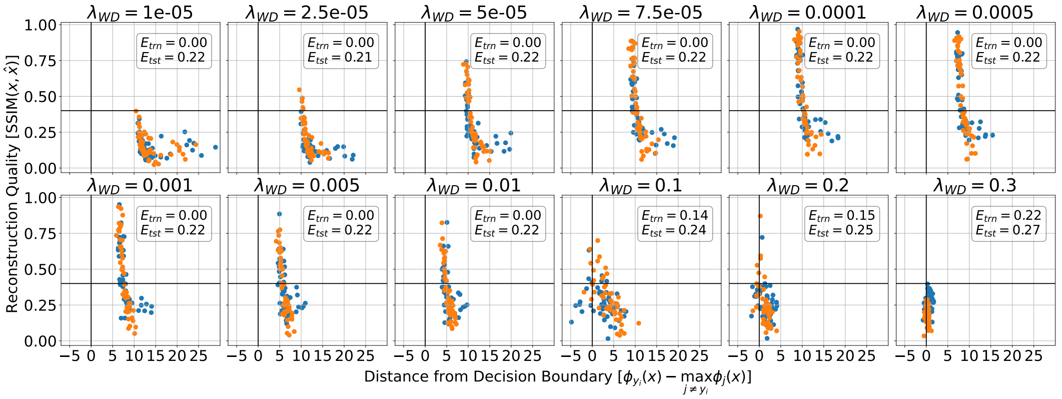
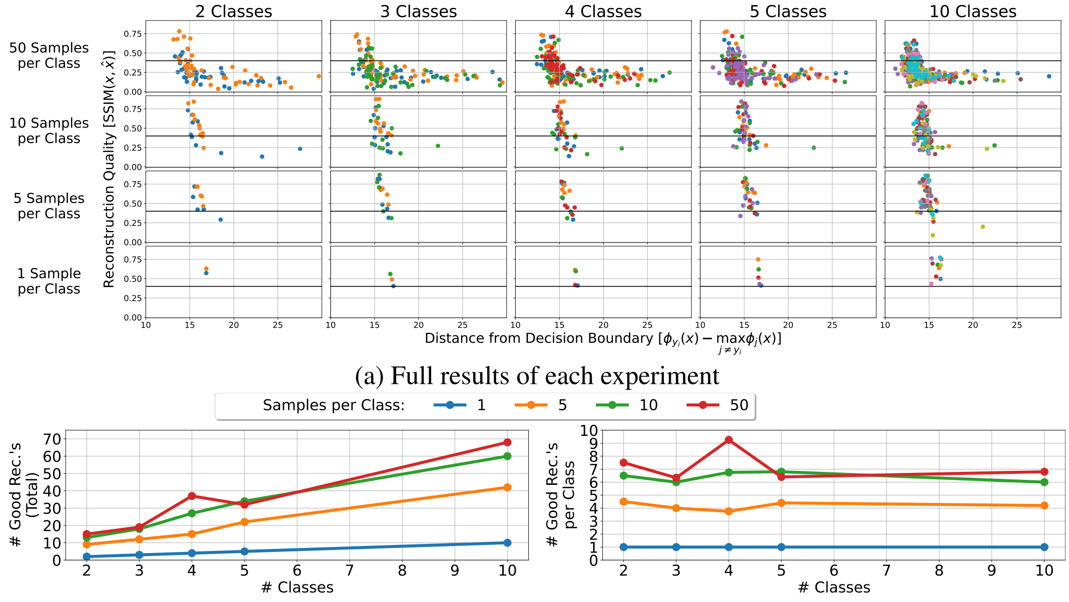


Figure 5: Weight-Decay “pushes” more samples to the margin, thus enabling them to be reconstructed



(a) Full results of each experiment

(b) Number of “good” reconstructions increases with number of classes and the samples per class

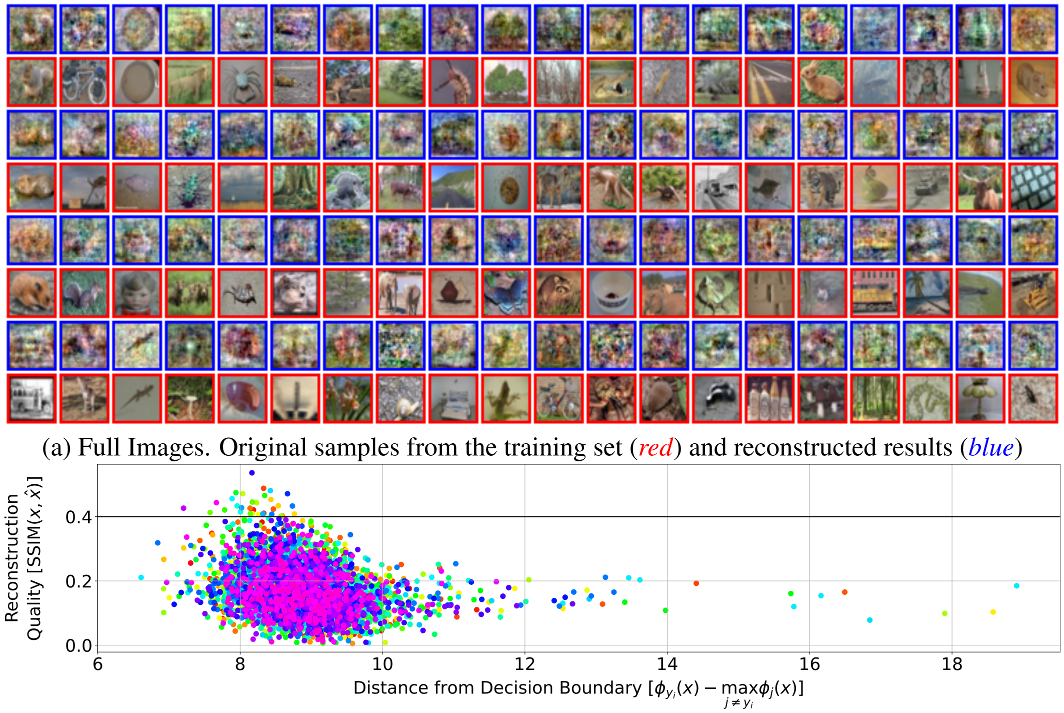
Figure 3: Evaluating effect of multiple classes on the ability to reconstruct. We show reconstructions from models trained with different number of classes and different number of samples per class. As seen, multiple classes result in more reconstructed samples.

4.3 Reconstruction From a Larger Number of Samples

One of the major limitations of Haim et al. (2022) is that they reconstruct from models that trained on a relatively small number of samples. Specifically, in their largest experiment, a model is trained with only 1000 samples. Here we take a step further, and apply our reconstruction scheme for a model trained on 5000 data samples.

To this end, we trained a 3-layer MLP, where the number of neurons in each hidden layer is $10,000$. Note that the size of the hidden layer is 10 times larger than in any other model we used. Increasing the number of neurons seems to be one of the major reasons for which we are able to reconstruct from such large datasets, although we believe it could be done with smaller models, which we leave for future research. We used the CIFAR100 dataset, with 50 samples in each class, for a total of 5000 samples.

In fig. 6a we give the best reconstructions of the model. Note that although there is a degradation in the quality of the reconstruction w.r.t a model trained on less samples, it is still clear that our scheme can reconstruct some of the training samples to some extent. In fig. 6b we show a scatter plot of the SSIM score w.r.t the distance from the boundary, similar to fig. 3a. Although most of the samples are on or close to the margin, only a few dozens achieve an SSIM> 0.4. This may indicate that there is a potential for much more images to reconstruct, and possibly with better quality.



(a) Full Images. Original samples from the training set (*red*) and reconstructed results (*blue*)

(b) Scatter plot (similar to fig. 3).

Figure 6: Reconstruction from a model trained on 50 images per class from the CIFAR100 dataset (100 classes, total of 5000 datapoints)

5 Conclusions and Future Work

In this paper we have shown several improvements over Haim et al. (2022). Most notably, we have shown that it is possible to reconstruct training data in a multi-class setting, compared to only a binary classification setting in the previous work. Additionally, Haim et al. (2022) showed reconstructions only from networks trained with small initialization. Here we show reconstructions from networks trained with weight decay, which is much more standard than a small initialization scale. Finally, we show it is possible to reconstruct from models trained on 5000 data samples, which is 5 times more than the largest trained model in Haim et al. (2022)

There are a couple of future research directions that we think might be interesting. First, to extend our reconstruction scheme to more practical models such as CNN and ResNets. Second, to reconstruct from models trained on more data, such as the entire CIFAR 10/100 datasets, or different types of data such as text, time-series or tabular data. Finally, it would be interesting to find privacy schemes which could protect from reconstruction attacks by specifically protecting samples which lie on the margin.

ACKNOWLEDGMENTS

This project received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 788535), and ERC grant 754705, and from the D. Dan and Betty Kahn Foundation. GV acknowledges the support of the NSF and the Simons Foundation for the Collaboration on the Theoretical Foundations of Deep Learning.

REFERENCES

Borja Balle, Giovanni Cherubin, and Jamie Hayes. Reconstructing training data with informed adversaries. *arXiv preprint arXiv:2201.04845*, 2022.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium (USENIX Security 19)*, pp. 267–284, 2019.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2633–2650, 2021.

Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. *arXiv preprint arXiv:2301.13188*, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pp. 1322–1333, 2015.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

Niv Haim, Gal Vardi, Gilad Yehudai, Ohad Shamir, and Michal Irani. Reconstructing training data from trained neural networks. *NeurIPS*, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Zecheng He, Tianwei Zhang, and Ruby B Lee. Model inversion attacks against collaborative inference. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pp. 148–162, 2019.

Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. Deep models under the gan: information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pp. 603–618, 2017.

Yangsibo Huang, Samyak Gupta, Zhao Song, Kai Li, and Sanjeev Arora. Evaluating gradient inversion attacks and defenses in federated learning. *Advances in Neural Information Processing Systems*, 34:7232–7241, 2021.

Ziwei Ji and Matus Telgarsky. Directional convergence and alignment in deep learning. *Advances in Neural Information Processing Systems*, 33:17176–17186, 2020.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Kaifeng Lyu and Jian Li. Gradient descent maximizes the margin of homogeneous neural networks. *arXiv preprint arXiv:1906.05890*, 2019.

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)*, pp. 691–706. IEEE, 2019.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19 (1):2822–2878, 2018.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Ziqi Yang, Jiyi Zhang, Ee-Chien Chang, and Zhenkai Liang. Neural network inversion in adversarial setting via background knowledge alignment. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 225–240, 2019.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

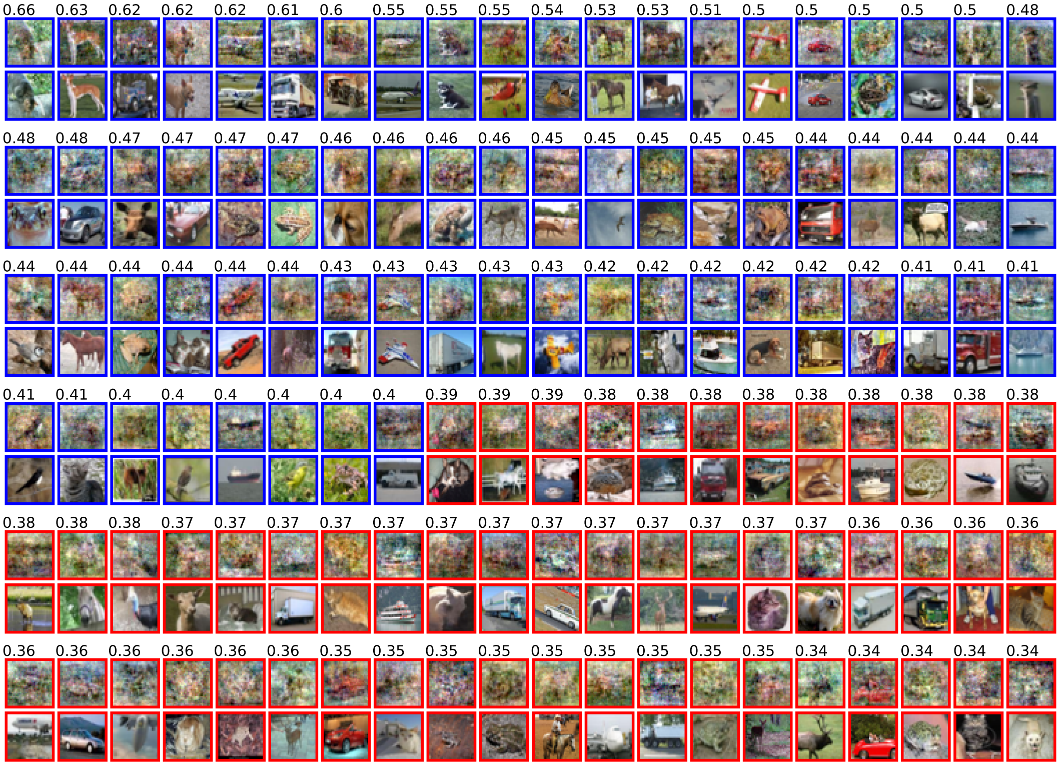


Figure 7: Justifying the threshold of SSIM$= 0.4$ as good rule-of-thumb for a threshold for a "good" reconstruction. Note that samples with SSIM> 0.4 (blue) are visually similar. Also some of the samples with SSIM< 0.4 (red) are similar. In general deciding whether a reconstruction is "good" is an open question beyond the scope of this paper. The SSIM values are shown above each train-reconstruction pair.

A Deciding whether a Reconstruction is "Good"

Here we justify our selection for SSIM$= 0.4$ as the threshold for what we consider as a "good" reconstruction. In general, the problem of deciding whether a reconstruction is the correct match to a given sample, or whether a reconstruction is a "good" reconstruction is equivalent to the problem of comparing between images. No "synthetic" metric (like SSIM, $l2$ etc.) will be aligned with human perception. A common metric for this purpose is LPIPS Zhang et al. (2018) that uses a classifier trained on Imagenet Deng et al. (2009), but since CIFAR images are much smaller than Imagenet images (32×32 vs. 224×224) it is not clear that this metric will be better. As a simple rule of thumb, we use SSIM> 0.4 for deciding that a given reconstruction is "good". To justify, we plot the best reconstructions (in terms of SSIM) in fig. 7. Note that almost all samples with SSIM> 0.4 are also visually similar (for a human). Also note that some of the samples with SSIM< 0.4 are visually similar, so in this sense we are "missing" some good reconstructions. In general, determining whether a reconstruction is "good" is an open question which cannot be dealt in the scope of this paper, and is a future direction for our work.

B Experiments with Different Number of Classes and Fixed Training Set Size

To complete the experiment shown in fig. 3, we also perform experiments on models trained on various number of classes ($C \in \{2, 3, 4, 5, 10\}$) but this time with a fixed training set size of 500 samples (distributed equally between classes). It seems from fig. 8 that the results are not much different from those for 50 samples per class, and we hypothesize that the model only needs a certain amount of "support vectors" to support its parameters, and this number is also achieved

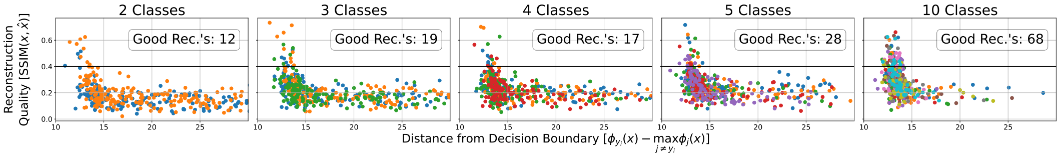


Figure 8: Experiments of reconstruction from models trained on a a fixed training set size (500 samples) for different number of classes. Number of "good" reconstruction is shown for each model.

by 50 samples per class, and are not harmed if more data is added for each class. Also note that for models with less classes, not only the number of good reconstruction decreases, but also the quality of reconstruction (lower SSIM). Since we don't have a good heuristic for aggregating the reconstruction quality score of a given model this observation is hard to quantify, but evident from the plot itself.