DP-MAC: The Differentially Private Method of Auxiliary Coordinates for Deep Learning

Frederik Harder[1] **Jonas Köhler**[2] **Max Welling**[3] **Mijung Park**[4]

Abstract

Developing a differentially private deep learning algorithm is challenging, due to the difficulty in analyzing the *sensitivity* of objective functions that are typically used to train deep neural networks. Many existing methods resort to the stochastic gradient descent algorithm and apply a *pre-defined* sensitivity to the gradients for privatizing weights. However, their slow convergence typically yields a high cumulative privacy loss. Here, we take a different route by employing the *method of auxiliary coordinates*, which allows us to independently update the weights per layer by optimizing a *per-layer* objective function. This objective function can be well approximated by a low-order Taylor's expansion, in which sensitivity analysis becomes tractable. We perturb the coefficients of the expansion for privacy, which we optimize using more advanced optimization routines than SGD for faster convergence. We empirically show that our algorithm provides a decent trained model quality under a modest privacy budget.

1 Introduction

While providing outstanding performance, it has been shown that trained deep neural networks (DNNs) can expose sensitive information from the dataset they were trained on [1, 2, 3, 4]. In order to protect potentially sensitive training data, many existing methods adopt the notion of privacy, called *differential privacy* (DP) [5]. Differentially private algorithms often comprise of a noise injection step (e.g. during the training process), which is generally detrimental to performance and leads to a trade-off between privacy and utility. The amount of noise necessary for a desired level of privacy depends on the *sensitivity* of an algorithm, a maximum difference in its output depending on whether or not a single individual participates in the data. In DNNs, the sensitivity of an objective function is often intractable to quantify, since data appears in the function in a nested and complex way. In addition, such models have thousands to millions of parameters, one needs to saveguard, and require many passes over the dataset in training. As a result, providing meaningful privacy guarantees while maintaining reasonable performance remains a challenging task for DNNs.

One existing approach to this problem, DP-SGD [6, 7, 8], avoids complicated sensitivities, by applying a pre-defined sensitivity to the gradients, which are then perturbed with Gaussian noise before updating the weights to ensure DP. This work also introduces the *moments accountant* (MA) [6], a useful method for computing cumulative privacy loss when training for many epochs (a formal introduction to this method is found in Appendix A). In another line of recent work [9, 10, 11], DP training is achieved by approximating the nested objective function through Taylor approximation and perturbing each of the coefficients of the approximated loss before training.

In this paper, we combine the benefits of these two approaches. We modify the algorithm called the *method of auxiliary coordinates* (MAC), which allows independent weight updates per layer, by

1,2,4: Max Planck Institute for Intelligent Systems, 3: University of Amsterdam

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.
Workshop on Privacy Preserving Machine Learning

framing the interaction between layers as a local communication problem via introducing auxiliary coordinates [12]. This allows us to split the nested objective function into per-layer objective functions, which can be approximated by low-order Taylor's expansions. In this case the sensitivity analysis of the coefficients becomes tractable.

2 DP-MAC

2.1 The Method of Auxiliary Coordinates

Here we provide a short introduction on the MAC algorithm (see [12] for details). Under a fully connected neural net with K hidden layers, a typical mean squared error (MSE) objective is given by

$$E(\mathbf{W}) = \tfrac{1}{2N} \sum_{n=1}^{N} ||\mathbf{y}_n - \mathbf{f}(\mathbf{x}_n; \mathbf{W})||^2, \tag{1}$$

where $\mathbf{f}(\mathbf{x}_n; \mathbf{W}) = \mathbf{f}_{K+1}(\ldots \mathbf{f}_2(\mathbf{f}_1(\mathbf{x}_n; \mathbf{W}_1)\ldots); \mathbf{W}_{K+1})$. We denote $\mathbf{W}$ as a collection of weight matrices of $(K+1)$-layers, i.e., $\mathbf{W} = \{\mathbf{W}_k\}_{k=1}^{K+1}$, where the size of each weight matrix is given by $\mathbf{W}_k \in \mathbb{R}^{D^k_{in} \times D^k_{out}}$. Each layer activation function is given by $\mathbf{f}_k(\mathbf{x}_n; \mathbf{W}_k) = \mathbf{f}_k({\mathbf{W}_k}^\top \mathbf{x}_n)$, and $\mathbf{f}_k$ could be any type of element-wise activation functions. In the MAC framework [12], the objective function in eq. 1 is expanded by adding auxiliary variables $\{\mathbf{z}_n\}$ (one per datapoint) such that the optimization over many variables are decoupled:

$$E(\mathbf{W}, \mathbf{Z}; \mu) = E_o(\mathbf{W}, \mathbf{Z}) + \sum_{k=1}^{K} E_k(\mathbf{W}, \mathbf{Z}, \mu), \tag{2}$$

where the partial objective functions at the output layer and at the k-th layer are given by $E_o(\mathbf{W}_{K+1}, \mathbf{Z}_K) = \frac{1}{2N}\sum_{n=1}^{N} ||\mathbf{y}_n - \mathbf{f}_{K+1}(\mathbf{z}_{K,n}; \mathbf{W}_{K+1})||^2$, and $E_k(\mathbf{W}_k, \mathbf{Z}_k, \mathbf{Z}_{k-1}, \mu) = \frac{\mu}{2N}\sum_{n=1}^{N} ||\mathbf{z}_{k,n} - \mathbf{f}_k(\mathbf{z}_{k-1,n}; \mathbf{W}_k)||^2$. Alternating optimization of this objective function w.r.t. $\mathbf{W}$ and $\mathbf{Z}$ minimizes the objective function. In this paper, we set $\mu = 1$, as suggested in [12]. For obtaining differentially private estimates of $\mathbf{W}$, it turns out we need to privatize the $\mathbf{W}$ update steps only, while we can keep the $\mathbf{Z}$ update steps non-private, as has been studied in Expectation Maximization (EM) type algorithms before [13, 14].

To make this process DP, we first approximate each objective function as 1st or 2nd-order polynomials in the weights. Then, we perturb each approximate objective function by adding noise to the coefficients and optimize it for estimating $\mathbf{W}$. How much noise we need to add to these coefficients depends on the sensitivity of these coefficients as well as the privacy loss we allow in each training step. The final estimate $\mathbf{W}_T$ depends on estimates $\mathbf{W}_t$ and $\mathbf{Z}_t$ for all $t < T$ and so we keep the privacy loss per iteration fixed and compute the cumulative loss using the moments accountant.

2.2 DP-approximate to the per-layer objective function

First, we consider approximating the per-layer objective function via the 2nd-order Taylor expansion

$$E_k(\mathbf{W}_k) = \tfrac{1}{2N} \sum_{n=1}^{N} ||\mathbf{z}_{k,n} - \mathbf{f}_k(\mathbf{z}_{k-1,n}; \mathbf{W}_k)||^2 \approx a_k + \sum_{h=1}^{D^k_{out}} {\mathbf{w}_{kh}}^\top \mathbf{b}_{kh} + \sum_{h=1}^{D^k_{out}} {\mathbf{w}_{kh}}^\top \mathbf{C}_{kh} \mathbf{w}_{kh},$$

where $\mathbf{w}_{kh} \in \mathbb{R}^{D^k_{in}}$ is the h-th column of the matrix $\mathbf{W}_k$, and the derivation of each term $a_k, \mathbf{b}_{kh} \in \mathbb{R}^{D^k_{in}}, \mathbf{C}_{kh} \in \mathbb{R}^{D^k_{in} \times D^k_{in}}$ is given below. Here we choose to use the *softplus* function as an example activation function for f, but any twice differentiable function is valid. We introduce a new notation $T_n(\mathbf{w}_{kh})$

$$E_k(\mathbf{W}_k) = \tfrac{1}{2N} \sum_{n=1}^{N} \sum_{h=1}^{D^k_{out}} T_n(\mathbf{w}_{kh}) \tag{3}$$

where $T_n(\mathbf{w}_{kh}) = z_{kh,n}^2 - 2z_{kh,n} f({\mathbf{w}_{kh}}^\top \mathbf{z}_{k-1,n}) + \{f({\mathbf{w}_{kh}}^\top \mathbf{z}_{k-1,n})\}^2$. We then approximate $T_n(\mathbf{w}_{kh})$ by the 2nd-order Taylor expansion evaluated at $\hat{\mathbf{w}}_{kh}$. In the first optimization step, we approximate the loss function by the 2nd-order Taylor expansion evaluated at a randomly drawn $\hat{\mathbf{w}}_{kh} \sim \mathcal{N}(0, I)$. In the consecutive optimization step, we evaluate the loss function at the noised-up estimate $\hat{\mathbf{w}}_{kh}$ obtained from the previous optimization step.

$$T_n(\mathbf{w}_{kh}) \approx T_n(\hat{\mathbf{w}}_{kh}) + (\mathbf{w}_{kh} - \hat{\mathbf{w}}_{kh})^\top \partial T_{nkh} + \frac{1}{2}(\mathbf{w}_{kh} - \hat{\mathbf{w}}_{kh})^\top \partial^2 T_{nkh} (\mathbf{w}_{kh} - \hat{\mathbf{w}}_{kh}), \tag{4}$$

where the derivative expressions of $T_n(\mathbf{w}_{kh})$ are given by

$$\begin{aligned}
\partial T_{nkh} &= [-2z_{kh,n} f'({\hat{\mathbf{w}}_{kh}}^\top \mathbf{z}_{k-1,n}) + 2f({\hat{\mathbf{w}}_{kh}}^\top \mathbf{z}_{k-1,n}) f'({\hat{\mathbf{w}}_{kh}}^\top \mathbf{z}_{k-1,n})]\mathbf{z}_{k-1,n}, \\
\partial^2 T_{nkh} &= [-2z_{kh,n} f''({\hat{\mathbf{w}}_{kh}}^\top \mathbf{z}_{k-1,n}) + 2f({\hat{\mathbf{w}}_{kh}}^\top \mathbf{z}_{k-1,n}) f''({\hat{\mathbf{w}}_{kh}}^\top \mathbf{z}_{k-1,n}) + 2\{f'({\hat{\mathbf{w}}_{kh}}^\top \mathbf{z}_{k-1,n})\}^2]\mathbf{z}_{k-1,n}{\mathbf{z}_{k-1,n}}^\top.
\end{aligned}$$

From this, we define the coefficients $a_k, \mathbf{b}_{kh}, \mathbf{C}_{kh}$ as: $a_k = \frac{1}{2N}\sum_{n=1}^{N}\sum_{h=1}^{D_{out}^k}[T_n(\hat{\mathbf{w}}_{kh}) - \hat{\mathbf{w}}_{kh}{}^{\top}\partial T_{nkh} + \frac{1}{2}\hat{\mathbf{w}}_{kh}{}^{\top}\partial^2 T_{nkh}\hat{\mathbf{w}}_{kh}], \mathbf{b}_{kh} = \frac{1}{2N}\sum_{n=1}^{N}\left[\partial T_{nkh} - \partial^2 T_{nkh}\hat{\mathbf{w}}_{kh}\right]$, $\mathbf{C}_{kh} = \frac{1}{2N}\sum_{n=1}^{N}\frac{1}{2}\partial^2 T_{nkh}$.

Adding Gaussian noise to these coefficients for privacy modifies the objective function by

$$\tilde{a}_k + \sum_{h=1}^{D_{out}^k}\mathbf{w}_{kh}{}^{\top}\tilde{\mathbf{b}}_{kh} + \sum_{h=1}^{D_{out}^k}\mathbf{w}_{kh}{}^{\top}\tilde{\mathbf{C}}_{kh}\mathbf{w}_{kh}, \tag{5}$$

where $\tilde{a}_k = a_k + \mathcal{N}(0, (\Delta a_k)^2\sigma^2)$, $\tilde{\mathbf{b}}_k = \mathbf{b}_k + \mathcal{N}(0, (\Delta \mathbf{b}_k)^2\sigma^2\mathbf{I})$, $\tilde{\mathbf{C}}_k = \mathbf{C}_k + \mathcal{N}(0, (\Delta \mathbf{C}_k)^2\sigma^2\mathbf{I})$ and the amount of additive Gaussian noise depends on the sensitivity $(\Delta a_k, \Delta \mathbf{b}_k, \Delta \mathbf{C}_k)$ of each term [1]. When using a purely gradient-based optimization routine (e.g. Adam, unlike Conjugate Gradient), we don't have to perturb a_k and in the case of first order approximation this leaves only $\mathbf{b}_k$. This method is not limited to MSE objectives and in the classification task we use a binary cross-entropy objective analogously in the output layer.

Note that on the first $\mathbf{W}$ step, unperturbed 1st and 2nd-order approximations provide the same gradient. In this case, if we use vanilla SGD to optimize this first order approximation, this boils down to a variant of DP-SGD, which optimizes each layer objective function separately. Empirically, however, we found that we get better results than DP-SGD, thanks to (a) the flexibility in choosing a better optimizer in our case and (b) the tighter sensitivity per layer than the pre-defined clipping norm for the entire weights in SGD, as illustrated in Sec. 3.

Analytic Sensitivities of the coefficients are given in the appendix. Depending on the architecture of a neural network and dataset at hand, these analytic sensitivity bounds can often be loose, in which case we propose to take a more direct approach and bound the losses directly by clipping the norms of $\mathbf{b}_k, \mathbf{C}_k$ to some fixed values T_{b_k}, T_{C_k}. Our method for selecting these norms is detailed in Sec. 2.4.

2.3 Calculation of cumulative privacy loss

Using the moments accountant and the theorem for subsampled Gaussian mechanism given in [6] for composition requires caution, since the log moments of privacy loss are linearly growing, only if we draw fresh noise per new subsampled data. Up to this point our algorithm, unfortunately, draws many noises for many losses given a subsampled data. This is fixed by treating the perturbed layerwise objectives as vector quantity, i.e., $\tilde{\mathbf{E}}(\mathbf{w}) = [\tilde{E}_1(\mathbf{w}_1), \cdots, \tilde{E}_K(\mathbf{w}_K), \tilde{E}_o(\mathbf{w}_{K+1})]$. Then, we scale down each objective function by its own sensitivity times $\sqrt{K}$, so that the concatenated vector's sensitivity is 1. Then, we add the standard Gaussian noise with standard deviation σ to the vectors and scale up each perturbed quantities by their own sensitivity times $\sqrt{K}$. Details are given in appendix H. The DP-MAC algorithm is summarized in Algorithm 1.

Algorithm 1 DP-MAC algorithm

```
Require: Dataset $\mathcal{D}$, total number of iterations $T$, privacy parameter $\sigma^2$, sampling rate $q$
Ensure: $(\varepsilon, \delta)$-DP weights $\{\mathbf{W}_k\}_{k=1}^{K+1}$
  for number of Iterations $\leq T$ do
    1. Optimize eq. 2 for $\mathbf{Z}$
    2. Optimize eq. 5 (noised-up objective) for $\mathbf{W}$
  end for
  Calculate the total privacy loss $(\varepsilon, \delta)$ using moments accountant
```

2.4 Layer-wise bounds for tighter sensitivity

In deep models, the magnitude of layer losses often differs significantly. As a result, it can be beneficial to select separate clipping thresholds T_{b_k} and T_{C_k} per layer. This can be achieved in a differentially private manner using pre-training and a histogram release, as we explain in detail for T_{b_k} in this section (and the same holds for T_{C_k}). While [15] uses a dynamically computed bound per batch, we found that doing so incurs a large privacy loss and similar performance can be achieved with static bounds computed prior to training.

Learning T_{b_k} per layer To determine individual T_{b_k}, we start by using an initial threshold T_b^{max} for all layers and training the model for a short time (e.g. 1 epoch) using DP-MAC. This pre-trained model is then used to record the norms of feature representations for each layer and each sample in the dataset. For each layer, these values are aggregated into a histogram over the interval $[0, T_b^{max}]$ which is released in a differentially private manner using the Gaussian mechanism [16] to perturb each bin. Each T_{b_k} is chosen as the upper border of the

[1] Note that due to the additivity of noise, we can rewrite the perturbed per-layer objective as a sum of original objective plus a single noise term that is a sum of all the noise terms for the coefficients.

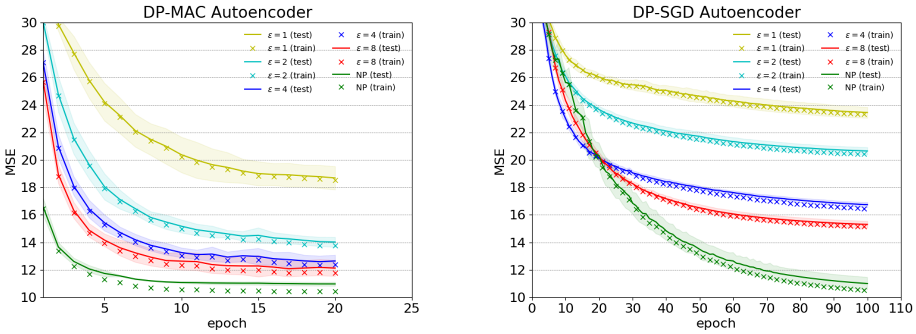


Figure 1: Autoencoder training and test errors. (± 1 stdev. of the latter) averaged over 10 runs each.

largest bin in the corresponding histogram, as in [15]. Afterwards, the pre-trained model is discarded and the proper training process starts using the new layer-wise bounds.

Privacy analysis for DP-MAC with learning T_{b_k} The privacy costs of each part (pre-training, histogram release and actual training) are computed individually and then summed using basic additive composition. As both training parts follow the same analysis, only the histogram release is left to be explained. Each histogram contains one count per sample, giving it sensitivity 1, as adding or removing a single sample only changes the total count of a single bin by one. Each bin is therefore perturbed using the Gaussian mechanism with a chosen privacy parameter σ_{hist}. Across layers, the privacy loss is aggregated using the moments accountant with the log moment of the Gaussian mechanism defined as $\alpha_G(\lambda) = (\lambda^2 + \lambda)/4\sigma^2$ [15]. As the total count of each histogram equals the size of the dataset, σ_{hist} can be chosen large enough that the privacy cost of the histogram release is negligible compared to the MAC-training without significantly affecting the results. A summary of the algorithm can be found in Appendix C.

3 Experiments

Autoencoder We highlight the advantages of DP-MAC, when training deeper models, in a reconstruction task with a fully connected autoencoder with 6 layers, as used in the original MAC paper [12] with the difference of using softplus activations in all layers, and not storing any **z** values. For this, we use the USPS dataset is a collection of 16x16 pixel grayscale handwritten digits, of with we use 5000 samples for training and 1000 for testing. We provide results for ε values of $1, 2, 4, 8$ with $\delta =$1e-5. For comparison, we train the same model using DP-SGD. As suggested in [6], we clipped the gradients using the mean gradient norms, although obtaining the clipping norm is *not* taken into account in the privacy analysis. For stability of training, we additionally introduce a maximum norm bound T_g.

In Fig 1, we observe that DP-MAC significantly outperforms DP-SGD on this task. We suspect that this is in part owed to the difference in norm clipping between the two methods. While DP-SGD normalizes the per-sample gradients across all layers as a whole, DP-MAC normalizes loss (and thus the resulting gradients) per layer, preventing layers with larger gradients from drowning out the rest. We also observe that DP-MAC using the Adam optimizer converges significantly faster than DP-SGD using the vanilla SGD optimizer. We show how the reconstructions of USPS data look in different private settings in Appendix C.

Classifier Due to the page limit, in Appendix B we show the performance of our method compared to other existing methods on a classification task, where our method achieves a comparable test accuracy under the same privacy constraint within a relatively small number of training epochs.

4 Conclusion

We present a novel differentially private deep learning paradigm, DP-MAC, which allows us to compute the sensitivity of the approximate objective functions analytically. For obtaining tighter sensitivities, we propose to use the DP-histogram mechanism to learn the sensitivity privately from the data. Since DP-MAC perturbs the objective function itself, we can decrease the cumulative privacy loss significantly by choosing to use a more efficient optimizer than vanilla SGD.

References

[1] Nicholas Carlini, Chang Liu, Jernej Kos, Úlfar Erlingsson, and Dawn Song. The secret sharer: Measuring unintended neural network memorization & extracting secrets. *CoRR*, abs/1802.08232, 2018.

[2] Congzheng Song, Thomas Ristenpart, and Vitaly Shmatikov. Machine learning models that remember too much. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, pages 587–601, New York, NY, USA, 2017. ACM.

[3] R. Shokri and V. Shmatikov. Privacy-preserving deep learning. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 909–910, Sept 2015.

[4] 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*, CCS '15, pages 1322–1333, New York, NY, USA, 2015. ACM.

[5] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9:211–407, August 2014.

[6] M. Abadi, A. Chu, I. Goodfellow, H. Brendan McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. *ArXiv e-prints*, July 2016.

[7] Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data. In *Proceedings of the International Conference on Learning Representations (ICLR)*, April 2017.

[8] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private language models without losing accuracy. *CoRR*, abs/1710.06963, 2017.

[9] J. Zhang, Z. Zhang, X. Xiao, Y. Yang, and M. Winslett. Functional Mechanism: Regression Analysis under Differential Privacy. *ArXiv e-prints*, August 2012.

[10] NhatHai Phan, Yue Wang, Xintao Wu, and Dejing Dou. Differential privacy preservation for deep auto-encoders: an application of human behavior prediction, 2016.

[11] N. Phan, X. Wu, and D. Dou. Preserving Differential Privacy in Convolutional Deep Belief Networks. *ArXiv e-prints*, June 2017.

[12] Miguel Carreira-Perpinan and Weiran Wang. Distributed optimization of deeply nested systems. In Samuel Kaski and Jukka Corander, editors, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pages 10–19, Reykjavik, Iceland, 22–25 Apr 2014. PMLR.

[13] Mijung Park, James Foulds, Kamalika Choudhary, and Max Welling. DP-EM: Differentially Private Expectation Maximization. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 896–904, Fort Lauderdale, FL, USA, 20–22 Apr 2017. PMLR.

[14] Mijung Park, James Foulds, Kamalika Chaudhuri, and Max Welling. Variational Bayes In Private Settings (VIPS). *ArXiv e-prints*, November 2016.

[15] Gergely Ács, Luca Melis, Claude Castelluccia, and Emiliano De Cristofaro. Differentially private mixture of generative neural networks. *CoRR*, abs/1709.04514, 2017.

[16] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503. Springer, 2006.

[17] Anand D. Sarwate and Kamalika Chaudhuri. Signal processing and machine learning with differential privacy: Algorithms and challenges for continuous data. *IEEE Signal Process. Mag.*, 30(5):86–94, 2013.

Appendix A: Differential Privacy Background

Here we provide background information on the definition of algorithmic privacy and a composition method that we will use in our algorithm, as well as the general formulation of the MAC algorithm.

Differential privacy

Differential privacy (DP) is a formal definition of the privacy properties of data analysis algorithms [5]. Given an algorithm $\mathcal{M}$ and neighbouring datasets $\mathcal{D}$, $\mathcal{D}'$ differing by a single entry. Here, we focus on the inclusion-exclusion[1] case, i.e., the dataset $\mathcal{D}'$ is obtained by excluding one datapoint from the dataset $\mathcal{D}$. The *privacy loss* random variable of an outcome o is $L^{(o)} = \log \frac{Pr(\mathcal{M}_{(\mathcal{D})}=o)}{Pr(\mathcal{M}_{(\mathcal{D}')}=o)}$. The mechanism $\mathcal{M}$ is called ε-DP if and only if $|L^{(o)}| \leq \varepsilon, \forall o$. A weaker version of the above is (ε, δ)-DP, if and only if $|L^{(o)}| \leq \varepsilon$, with probability at least $1 - \delta$. What the definition states is that a single individual's participation in the data do not change the output probabilities by much, which limits the amount of information that the algorithm reveals about any one individual.

The most common form of designing differentially private algorithms is by adding noise to a quantity of interest, e.g., a deterministic function $h : \mathcal{D} \mapsto \mathbb{R}^p$ computed on sensitive data $\mathcal{D}$. See [5] and [17] for more forms of designing differentially-private algorithms. For privatizing h, one could use the *Gaussian mechanism* [16] which adds noise to the function, where the noise is calibrated to h's *sensitivity*, S_h, defined by the maximum difference in terms of L2-norm, $||h(\mathcal{D}) - h(\mathcal{D}')||_2$, $\tilde{h}(\mathcal{D}) = h(\mathcal{D}) + \mathcal{N}(0, S_h^2\sigma^2\mathbf{I}_p)$, where $\mathcal{N}(0, S_h^2\sigma^2\mathbf{I}_p)$ means the Gaussian distribution with mean 0 and covariance $S_h^2\sigma^2\mathbf{I}_p$. The perturbed function $\tilde{h}(\mathcal{D})$ is (ε, δ)-DP, where $\sigma \geq \sqrt{2\log(1.25/\delta)}/\varepsilon$. In this paper, we use the Gaussian mechanism to achieve differentially private network weights. Next, we describe how the cumulative privacy loss is calculated when we use the Gaussian mechanism repeatedly during training.

The moments accountant

In the moments accountant, a cumulative privacy loss is calculated by bounding the moments of $L^{(o)}$, where the λ-th moment is defined as the log of the moment generating function evaluated at λ [6]: $\alpha_{\mathcal{M}}(\lambda; \mathcal{D}, \mathcal{D}') = \log \mathbb{E}_{o\sim\mathcal{M}(\mathcal{D})}\left[e^{\lambda L^{(o)}}\right]$. By taking the maximum over the neighbouring datasets, we obtain the worst case λ-th moment $\alpha_{\mathcal{M}}(\lambda) = \max_{\mathcal{D},\mathcal{D}'} \alpha_{\mathcal{M}}(\lambda; \mathcal{D}, \mathcal{D}')$, where the form of $\alpha_{\mathcal{M}}(\lambda)$ is determined by the mechanism of choice. The moments accountant compute $\alpha_{\mathcal{M}}(\lambda)$ at each step. Due to the composability theorem which states that the λ-th moment composes linearly (See the composability theorem: Theorem 2.1 in [6] when independent noise is added in each step, we can simply sum each upper bound on $\alpha_{\mathcal{M}_j}$ to obtain an upper bound on the total λ-th moment after T compositions, $\alpha_{\mathcal{M}}(\lambda) \leq \sum_{j=1}^{T} \alpha_{\mathcal{M}_j}(\lambda)$. Once the moment bound is computed, we can convert the λ-th moment to the (ε, δ)-DP, guarantee by, $\delta = \min_\lambda \exp\left[\alpha_{\mathcal{M}}(\lambda) - \lambda\varepsilon\right]$, for any $\varepsilon > 0$. See Appendix A in [6] for the proof.

[1] This is for using the moments accountant method when calculating the cumulative privacy loss.

Appendix B: Experiment Results

	DP-SGD	DP-CDBN	**DP-MAC**
$\varepsilon = 0.5$	0.90	0.92	0.91
# epochs	16	162	20
$\varepsilon = 2$	0.95	0.95	0.95
# epochs	120	162	20

(a) Test classification accuracy on MNIST

	DP-SGD	**DP-MAC**
$\varepsilon = 1$	23.5	19
$\varepsilon = 2$	20.8	14
$\varepsilon = 4$	16.8	12.7
$\varepsilon = 8$	15.4	12

(b) Test reconstruction MSE on USPS

Table 1: Test performance of DP-MAC compared to [6] DP-SGD and DP-CDBN [11] at $\delta = 10^{-5}$

	DP-MAC Classifier	DP-MAC Autoencoder	DP-SGD Autoencoder
layer-sizes	300-100	300-100-20-100-300-256	300-100-20-100-300-256
batch size	50	50	100
train epochs	20	20	100
optimizer	Adam	Adam	SGD
$\mathbf{W}$ learning rate	0.003	0.001	0.03
$\mathbf{z}$ learning rate	0.01	0.02	
$\mathbf{W}$ lr-decay	0.9	0.9	0.97
$\mathbf{z}$-steps	20	30	
$\mathbf{W}$-steps	1	1	
T_b^{max}	0.1	0.01	
T_z	30	12.5	
T_w	30	20	
T_g			0.01 - 0.003
σ values	1.7, 5.8	2.2, 3, 5.2, 12.1	1.3, 2, 3.7, 7
σ_{hist}	20	20	
histogram bins	20	20	

Table 2: Training parameters choices for both DP-MAC experiments and the DP-SGD autoencoder comparison

Appendix C: Additional Figures

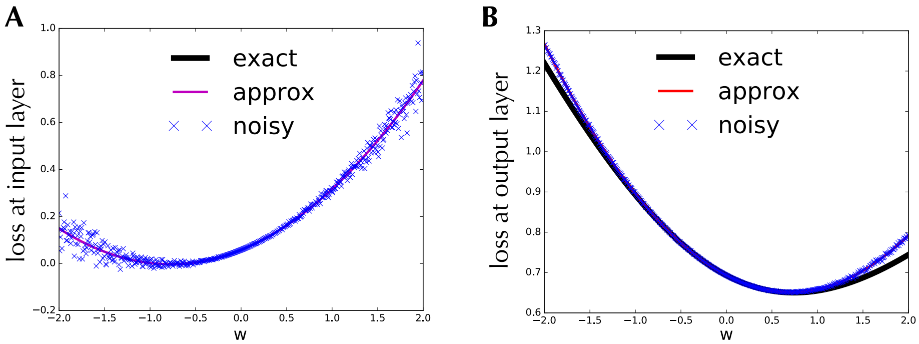


Figure 2: The input and output objective functions (black) are well approximated by the 2nd-order approximations (red). In both cases, approximation is made at 0, where the true w at the input layer is -0.7, and 0.7 at the output layer. The blue crosses depict additive noise centered around the approximated loss and the noise variance is determined by the sensitivities of the coefficients and privacy parameter σ^2.

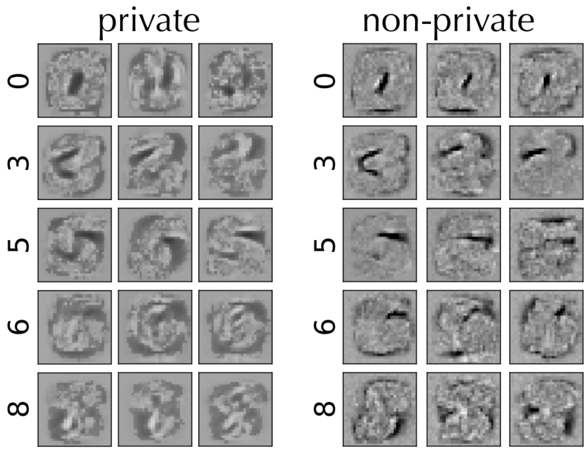
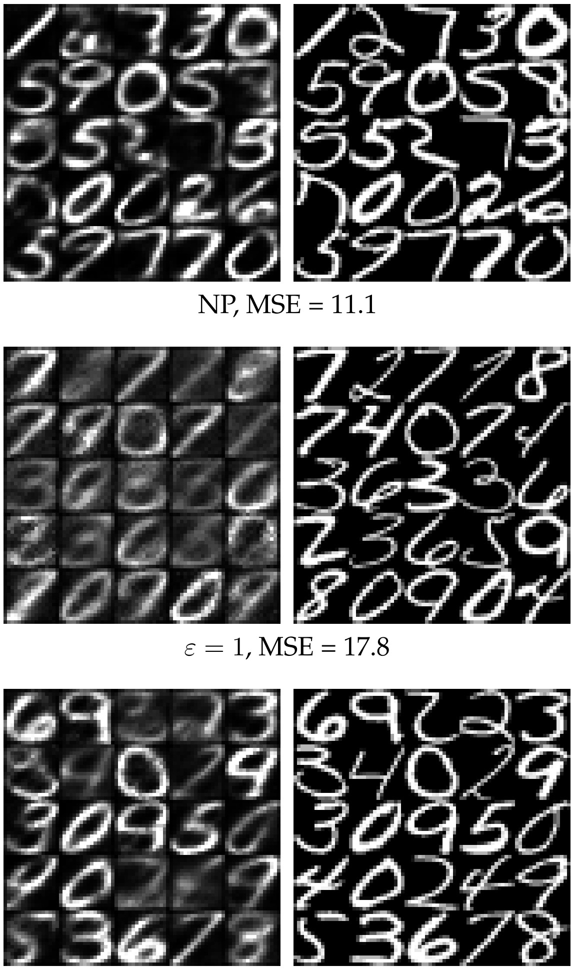


Figure 3: Learned significant features for labels 0,3,5,6,8 respectively. The non-private features show higher contrast and more characteristics in the high frequencies, whereas the private features become smoothed out and lose contrast.

Algorithm 2 DP-MAC with learning T_{b_k}

```
Require: $\mathcal{D}$, $T$, $\sigma^2$, $\sigma^2_{hist}$, $q$, initial threshold $T_{b_k}$
Ensure: $(\varepsilon, \delta)$-DP weights $\{\mathbf{W}_k\}_{k=1}^{K+1}$
  1. Pre-training using DP-MAC (Algo. 1)
  2. DP-histogram release which determines $T_{b_k}$
  3. DP-MAC (Algo. 1) training using learned $T_{b_k}$
```



NP, MSE = 11.1

$\varepsilon = 1$, MSE = 17.8

$\sigma = 4$, MSE = 12.7

Figure 4: Autoencoder reconstructions (left) of USPS test set samples (right) for average mean squared error rates after training in both non-private and private settings

Appendix D: sensitivity of a_k

We are using a few assumptions and facts to derive sensitivities below.

- $||\mathbf{z}_{k,n}||_2 \leq T_z$ for a predefined threshold T_z for all k, n.
- $||\mathbf{w}_{kh}||_2 \leq T_w$ for a predefined threshold T_w for all k and h.
- Due to Cauchy-Schwarz inequality: $\mathbf{w}_{kh}^T \mathbf{z}_{k-1,n} \leq ||\mathbf{w}_{kh}||_2 T_z$
- Using a monotonic nonlinearity (e.g., softplus): $f(\mathbf{w}_{kh}^T \mathbf{z}_{k-1,N}) \leq f(||\mathbf{w}_{kh}||_2 T_z)$ and $f'(\mathbf{w}_{kh}^T \mathbf{z}_{k-1,N}) \leq f'(||\mathbf{w}_{kh}||_2 T_z)$
- For softplus, $0 < f'(\mathbf{w}_{kh}^T \mathbf{z}_{k-1,N}) \leq f'(||\mathbf{w}_{kh}||_2 T_z) \leq 1$ and $0 < f^{''}(\mathbf{w}_{kh}^T \mathbf{z}_{k-1,n}) \leq \frac{1}{4}$
- $||\mathbf{a}||_2 \leq ||\mathbf{a}||_1 \leq ||\mathbf{a}||_2 \sqrt{D}$ for $\mathbf{a} \in \mathbb{R}^D$
- Direct application of above : $|\sum_{h=1}^{D_{out}^k} z_{kh,n}| \leq \sum_{h=1}^{D_{out}^k} |z_{kh,n}| = ||\mathbf{z}_{k,n}|^\top \mathbf{1}| \leq T_z \sqrt{D_{out}^k}$

Denote

$$\begin{aligned} \alpha_{\hat{\mathbf{w}}_{kh}} &:= f(||\hat{\mathbf{w}}_{kh}||_2 T_z) \\ \beta_{\hat{\mathbf{w}}_{kh}} &:= f'(||\hat{\mathbf{w}}_{kh}||_2 T_z) \end{aligned}$$

which we will further denote as vectors $\boldsymbol{\alpha}_{\hat{\mathbf{w}}_k}, \boldsymbol{\beta}_{\hat{\mathbf{w}}_k}$.

Without loss of generality, we further assume that (1): the neighbouring datasets are in the form of $\mathcal{D} = \{\mathcal{D}', (\mathbf{x}_N, \mathbf{y}_N)\}$; and (2) the last entry maximizes the difference in a_k run on $\mathcal{D}, \mathcal{D}'$, i.e., the average over the first $(N-1)$ entries in a_k cannot exceed the value of the last entry in a_k.

$$\begin{aligned}
\Delta a_k &= \max_{|\mathcal{D} \backslash \mathcal{D}'|=1} |a_k(\mathcal{D}) - a_k(\mathcal{D}')|, \\
&= \max_{|\mathcal{D} \backslash \mathcal{D}'|=1} \tfrac{1}{2} | \sum_{h=1}^{D_{out}^k} \{ \tfrac{1}{N} \sum_{n=1}^{N} (T_n(\hat{\mathbf{w}}_{kh}) - \hat{\mathbf{w}}_{kh}{}^\top \partial T_{nkh} + \tfrac{1}{2} \hat{\mathbf{w}}_{kh}{}^\top \partial^2 T_{nkh} \hat{\mathbf{w}}_{kh}) - \tfrac{1}{N-1} \sum_{n=1}^{N-1} (T_n(\hat{\mathbf{w}}_{kh}) - \hat{\mathbf{w}}_{kh}{}^\top \partial T_{nkh} + \tfrac{1}{2} \hat{\mathbf{w}}_{kh}{}^\top \partial^2 T_{nkh} \hat{\mathbf{w}}_{kh}) \} | \\
&= \max_{|\mathcal{D} \backslash \mathcal{D}'|=1} \tfrac{1}{2} | \sum_{h=1}^{D_{out}^k} \{ \tfrac{N-1}{N} \tfrac{1}{N-1} \sum_{n=1}^{N-1} (T_n(\hat{\mathbf{w}}_{kh}) - \hat{\mathbf{w}}_{kh}{}^\top \partial T_{nkh} + \tfrac{1}{2} \hat{\mathbf{w}}_{kh}{}^\top \partial^2 T_{nkh} \hat{\mathbf{w}}_{kh}) + \tfrac{1}{N} (T_N(\hat{\mathbf{w}}_{kh}) - \hat{\mathbf{w}}_{kh}{}^\top \partial T_{Nkh} + \tfrac{1}{2} \hat{\mathbf{w}}_{kh}{}^\top \partial^2 T_{Nkh} \hat{\mathbf{w}}_{kh}) \\
&\qquad - \tfrac{1}{N-1} \sum_{n=1}^{N-1} (T_n(\hat{\mathbf{w}}_{kh}) - \hat{\mathbf{w}}_{kh}{}^\top \partial T_{nkh} + \tfrac{1}{2} \hat{\mathbf{w}}_{kh}{}^\top \partial^2 T_{nkh} \hat{\mathbf{w}}_{kh}) \} | \\
&= \max_{|\mathcal{D} \backslash \mathcal{D}'|=1} \tfrac{1}{2} | \sum_{h=1}^{D_{out}^k} \{ \tfrac{1}{N(N-1)} \sum_{n=1}^{N-1} (T_n(\hat{\mathbf{w}}_{kh}) - \hat{\mathbf{w}}_{kh}{}^\top \partial T_{nkh} + \tfrac{1}{2} \hat{\mathbf{w}}_{kh}{}^\top \partial^2 T_{nkh} \hat{\mathbf{w}}_{kh}) + \tfrac{1}{N} (T_N(\hat{\mathbf{w}}_{kh}) - \hat{\mathbf{w}}_{kh}{}^\top \partial T_{Nkh} + \tfrac{1}{2} \hat{\mathbf{w}}_{kh}{}^\top \partial^2 T_{Nkh} \hat{\mathbf{w}}_{kh}) \} | \\
&\leq \max_{|\mathcal{D} \backslash \mathcal{D}'|=1} \tfrac{1}{2} \tfrac{1}{N(N-1)} | \sum_{h=1}^{D_{out}^k} \sum_{n=1}^{N-1} (T_n(\hat{\mathbf{w}}_{kh}) - \hat{\mathbf{w}}_{kh}{}^\top \partial T_{nkh} + \tfrac{1}{2} \hat{\mathbf{w}}_{kh}{}^\top \partial^2 T_{nkh} \hat{\mathbf{w}}_{kh}) | \\
&\qquad + \tfrac{1}{2} \tfrac{1}{N} | \sum_{h=1}^{D_{out}^k} (T_N(\hat{\mathbf{w}}_{kh}) - \hat{\mathbf{w}}_{kh}{}^\top \partial T_{Nkh} + \tfrac{1}{2} \hat{\mathbf{w}}_{kh}{}^\top \partial^2 T_{Nkh} \hat{\mathbf{w}}_{kh}) \} |, \text{ due to Triangle inequality} \\
&\leq \max_{|\mathcal{D} \backslash \mathcal{D}'|=1} \tfrac{2}{2N} | \sum_{h=1}^{D_{out}^k} (T_N(\hat{\mathbf{w}}_{kh}) - \hat{\mathbf{w}}_{kh}{}^\top \partial T_{Nkh} + \tfrac{1}{2} \hat{\mathbf{w}}_{kh}{}^\top \partial^2 T_{Nkh} \hat{\mathbf{w}}_{kh}) \} |, \text{ due to assumption (2).}
\end{aligned}$$

Now the sensitivity can be divided into three terms due to triangle inequality as

$$\Delta a_k \quad \leq \underbrace{\max_{|\mathcal{D}\backslash\mathcal{D}'|=1} \left| \frac{2}{2N} \sum_{h=1}^{D_{out}^k} T_N(\hat{\mathbf{w}}_{kh}) \right|}_{\Delta a_{k_1}} + \underbrace{\max_{|\mathcal{D}\backslash\mathcal{D}'|=1} \left| \frac{2}{2N} \sum_{h=1}^{D_{out}^k} \hat{\mathbf{w}}_{kh}{}^\top \partial T_{Nkh} \right|}_{\Delta a_{k_2}} + \underbrace{\max_{|\mathcal{D}\backslash\mathcal{D}'|=1} \left| \frac{2}{2N} \sum_{h=1}^{D_{out}^k} \frac{1}{2} \hat{\mathbf{w}}_{kh}{}^\top \partial^2 T_{Nkh} \hat{\mathbf{w}}_{kh} \right|}_{\Delta a_{k_3}}.$$

We compute the sensitivity of each of these terms below. The sensitivity of a_{k_1} is given by

$$\begin{aligned}
\Delta a_{k_1} &= \frac{1}{N} \max_{\mathbf{z}_{k,N}, \mathbf{z}_{k-1,N}} \left| \sum_{h=1}^{D_{out}^k} z_{kh,N}^2 - 2z_{kh,N} f(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N}) + \big(f(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N})\big)^2 \right| \\
&\leq \frac{1}{N} \max_{\mathbf{z}_{k,N}, \mathbf{z}_{k-1,N}} \left| \sum_{h=1}^{D_{out}^k} z_{kh,N}^2 \right| + \left| \sum_{h=1}^{D_{out}^k} 2z_{kh,N} f(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N}) \right| + \left| \sum_{h=1}^{D_{out}^k} \big(f(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N})\big)^2 \right| \\
&\leq \frac{1}{N} \left(T_z^2 + 2T_z ||\boldsymbol{\alpha}_{\hat{\mathbf{w}}_k}||_2 + ||\boldsymbol{\beta}_{\hat{\mathbf{w}}_k}||_2^2\right)
\end{aligned}$$

The sensitivity of a_{k_2} is given by

$$\Delta a_{k_2} = \frac{1}{N} \max_{\mathbf{z}_{k,N}, \mathbf{z}_{k-1,N}} \left| \sum_{h=1}^{D_{out}^k} \big(-2z_{kh,N} f'(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N}) + 2f(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N}) f'(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N})\big) \hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N} \right| \tag{1}$$

$$\leq \frac{1}{N} \max_{\mathbf{z}_{k,N}, \mathbf{z}_{k-1,N}} \left| \sum_{h=1}^{D_{out}^k} \big(2z_{kh,N} f'(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N})\big) \hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N} \right| + \left| \sum_{h=1}^{D_{out}^k} \big(2f(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N}) f'(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N})\big) \hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N} \right| \tag{2}$$

$$\leq \frac{1}{N} \max_{\mathbf{z}_{k,N}, \mathbf{z}_{k-1,N}} \left| \sum_{h=1}^{D_{out}^k} ||2z_{kh,N} f'(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N}) \hat{\mathbf{w}}_{kh}||_2 \cdot ||\mathbf{z}_{k-1,N}||_2 \right| + \left| \sum_{h=1}^{D_{out}^k} ||2f(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N}) f'(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N}) \hat{\mathbf{w}}_{kh}||_2 \cdot ||\mathbf{z}_{k-1,N}||_2 \right|$$

$$\leq \frac{2T_z}{N} \max_{\mathbf{z}_{k,N}, \mathbf{z}_{k-1,N}} \left(\left| \sum_{h=1}^{D_{out}^k} |z_{kh,N}| \cdot ||f'(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N}) \hat{\mathbf{w}}_{kh}||_2 \right| + \left| \sum_{h=1}^{D_{out}^k} ||f(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N}) f'(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N}) \hat{\mathbf{w}}_{kh}||_2 \right| \right) \tag{3}$$

$$\leq \frac{2T_z}{N} \left(T_z \cdot \left(\sum_{h=1}^{D_{out}^k} \beta_{\hat{\mathbf{w}}_{kh}}^2 ||\mathbf{w}_{kh}||_2^2 \right)^{1/2} + \sum_{h=1}^{D_{out}^k} |\alpha_{\hat{\mathbf{w}}_{kh}} \beta_{\hat{\mathbf{w}}_{kh}}| \cdot ||\hat{\mathbf{w}}_{kh}||_2 \right), \tag{4}$$

The sensitivity of a_{k_3} is given by

$$
\begin{aligned}
\Delta a_{k_3} &= \frac{1}{N} \max_{\mathbf{z}_{k,N}, \mathbf{z}_{k-1,N}} \left| \sum_{h=1}^{D_{out}^k} \left(-z_{kh,N} f''(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N}) + \left(f'(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N}) \right)^2 + f(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N}) f''(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N}) \right) \left(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N} \right)^2 \right| \\
&\leq \frac{1}{N} \max_{\mathbf{z}_{k,N}, \mathbf{z}_{k-1,N}} \left| \sum_{h=1}^{D_{out}^k} \left(z_{kh,N} f''(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N}) \right) \left(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N} \right)^2 \right| + \left| \sum_{h=1}^{D_{out}^k} \left(\left(f'(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N}) \right)^2 \right) \left(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N} \right)^2 \right| \\
&\qquad\qquad + \left| \sum_{h=1}^{D_{out}^k} \left(f(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N}) f''(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N}) \right) \left(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N} \right)^2 \right| \\
&\leq \frac{1}{N} \max_{\mathbf{z}_{k,N}, \mathbf{z}_{k-1,N}} \left| \sum_{h=1}^{D_{out}^k} z_{kh,N} \cdot f''(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N}) \cdot ||\hat{\mathbf{w}}_{kh}||_2^2 \cdot ||\mathbf{z}_{k-1,N}||_2^2 \right| + \left| \sum_{h=1}^{D_{out}^k} \left(f'(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N}) \right)^2 \cdot ||\hat{\mathbf{w}}_{kh}||_2^2 \cdot ||\mathbf{z}_{k-1,N}||_2^2 \right| \\
&\qquad\qquad + \left| \sum_{h=1}^{D_{out}^k} f(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N}) f''(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N}) \cdot ||\hat{\mathbf{w}}_{kh}||_2^2 \cdot ||\mathbf{z}_{k-1,N}||_2^2 \right| \\
&\leq \frac{T_z^2}{N} \max_{\mathbf{z}_{k,N}} \left(\sum_{h=1}^{D_{out}^k} 1/4 \cdot |z_{kh,N}| \cdot ||\hat{\mathbf{w}}_{kh}||_2^2 + \sum_{h=1}^{D_{out}^k} \left(\beta_{\hat{\mathbf{w}}_{kh}} \right)^2 \cdot ||\hat{\mathbf{w}}_{kh}||_2^2 + \sum_{h=1}^{D_{out}^k} 1/4 \alpha_{\hat{\mathbf{w}}_{kh}} \cdot ||\hat{\mathbf{w}}_{kh}||_2^2 \right) \\
&\leq \frac{T_z^2}{4N} \left(T_z \left(\sum_{h=1}^{D_{out}^k} \left(||\mathbf{w}_{kh}||_2^2 \right)^2 \right)^{1/2} + \sum_{h=1}^{D_{out}^k} \left(4 \left(\beta_{\hat{\mathbf{w}}_{kh}} \right)^2 + \alpha_{\hat{\mathbf{w}}_{kh}} \right) \cdot ||\hat{\mathbf{w}}_{kh}||_2^2 \right)
\end{aligned}
$$

Appendix E: sensitivity of b_k

$$
\begin{aligned}
\Delta \mathbf{b}_k = \max_{|\mathcal{D} \setminus \mathcal{D}'|=1} ||\mathbf{b}_k(\mathcal{D}) - \mathbf{b}_k(\mathcal{D}')||_F &= \max_{|\mathcal{D} \setminus \mathcal{D}'|=1} \left(\sum_{h=1}^{D_{out}^k} ||\mathbf{b}_{kh}(\mathcal{D}) - \mathbf{b}_{kh}(\mathcal{D}')||_2^2 \right)^{\frac{1}{2}}, \\
&\leq \max_{\mathbf{z}_{k,N} \mathbf{z}_{k-1,N}, \mathbf{z}'_{k,N}, \mathbf{z}'_{k-1,N}} \tfrac{1}{N} \left(\sum_{h=1}^{D_{out}^k} ||(\partial T_N(\hat{\mathbf{w}}_{kh}) - \partial^2 T_N(\hat{\mathbf{w}}_{kh}) \hat{\mathbf{w}}_{kh}) - (\partial T'_N(\hat{\mathbf{w}}_{kh}) - \partial^2 T'_N(\hat{\mathbf{w}}_{kh}) \hat{\mathbf{w}}_{kh})||_2^2 \right)^{\frac{1}{2}} \\
&\leq \tfrac{1}{N} (\Delta b_{k_1} + \Delta b_{k_2})^{\frac{1}{2}}, \qquad (5)
\end{aligned}
$$

where

$$\Delta b_{k_1} = \max_{\mathbf{z}_{k,N}\mathbf{z}_{k-1,N},\mathbf{z}'_{k,N},\mathbf{z}'_{k-1,N}} \sum_{h=1}^{D^k_{out}} ||\partial T_N(\hat{\mathbf{w}}_{kh}) - \partial T'_N(\hat{\mathbf{w}}_{kh})||_2^2 \tag{6}$$

$$\Delta b_{k_2} = \max_{\mathbf{z}_{k,N}\mathbf{z}_{k-1,N},\mathbf{z}'_{k,N},\mathbf{z}'_{k-1,N}} \sum_{h=1}^{D^k_{out}} ||\partial^2 T_N(\hat{\mathbf{w}}_{kh})\hat{\mathbf{w}}_{kh} - \partial^2 T'_N(\hat{\mathbf{w}}_{kh})\hat{\mathbf{w}}_{kh}||_2^2, \tag{7}$$

$$\begin{aligned}
\boldsymbol{\Delta}\mathbf{b}_{k_1} &= \max_{\mathbf{z}_{k,N},\mathbf{z}_{k-1,N}} \sum_{h=1}^{D^k_{out}} || \left(-2z_{kh,N} f'(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N}) + 2 f(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N}) f'(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N})\right) \mathbf{z}_{k-1,N}||_2^2 \\
&\leq 2 \max_{\mathbf{z}_{k,N},\mathbf{z}_{k-1,N}} \sum_{h=1}^{D^k_{out}} \left|z_{kh,N} f'(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N})\right|^2 ||\mathbf{z}_{k-1,N}||_2^2 + \sum_{h=1}^{D^k_{out}} \left|f(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N}) f'(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N})\right|^2 ||\mathbf{z}_{k-1,N}||_2^2 \\
&\leq 2T_z^2 \max_{\mathbf{z}_{k,N}} \sum_{h=1}^{D^k_{out}} \left|z_{kh,N}\right|^2 + \sum_{h=1}^{D^k_{out}} \left|\alpha_{\hat{\mathbf{w}}_{kh}} \beta_{\hat{\mathbf{w}}_{kh}}\right|^2, \text{ since } \beta_{\hat{\mathbf{w}}_{kh}} \leq 1 \\
&\leq 2T_z^2 \left(T_z^2 + ||\boldsymbol{\alpha}_{\hat{\mathbf{w}}_k} \odot \boldsymbol{\beta}_{\hat{\mathbf{w}}_k}||_2^2\right)
\end{aligned}$$

$$\boldsymbol{\Delta}\mathbf{b}_{k_2} = \max_{\mathbf{z}_{k,N},\mathbf{z}_{k-1,N}} \sum_{h=1}^{D^k_{out}} ||\Big(\Big(- 2z_{kh,N} f''(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N}) + 2\left(f'(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N})\right)^2 \tag{8}$$

$$+ 2 f(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N}) f''(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N})\Big)\mathbf{z}_{k-1,N}\mathbf{z}_{k-1,N}^T\Big)\hat{\mathbf{w}}_{kh}||_2^2 \tag{9}$$

$$\leq 2 \max_{\mathbf{z}_{k,N},\mathbf{z}_{k-1,N}} \sum_{h=1}^{D^k_{out}} ||z_{kh,N} f''(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N})\mathbf{z}_{k-1,N}\mathbf{z}_{k-1,N}^T\hat{\mathbf{w}}_{kh}||_2^2 + || \left(f'(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N})\right)^2 \mathbf{z}_{k-1,N}\mathbf{z}_{k-1,N}^T\hat{\mathbf{w}}_{kh}||_2^2 \tag{10}$$

$$+ ||f(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N}) f''(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N})\mathbf{z}_{k-1,N}\mathbf{z}_{k-1,N}^T\hat{\mathbf{w}}_{kh}||_2^2 \tag{11}$$

$$\leq 2 \max_{\mathbf{z}_{k,N},\mathbf{z}_{k-1,N}} \sum_{h=1}^{D^k_{out}} ||1/4 \cdot z_{kh,N}\mathbf{z}_{k-1,N}\mathbf{z}_{k-1,N}^T\hat{\mathbf{w}}_{kh}||_2^2 \tag{12}$$

$$+ || \left(\left(f'(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N})\right)^2 + 1/4 \cdot f(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N})\right) \mathbf{z}_{k-1,N}\mathbf{z}_{k-1,N}^T\hat{\mathbf{w}}_{kh}||_2^2 \tag{13}$$

$$\leq 2 \max_{\mathbf{z}_{k,N},\mathbf{z}_{k-1,N}} \sum_{h=1}^{D^k_{out}} |1/4 \cdot z_{kh,N}|^2 \cdot T_z^2 \cdot T_z^2 ||\hat{\mathbf{w}}_{kh}||_2^2 \tag{14}$$

$$+ || \left(\left(f'(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N})\right)^2 + 1/4 \cdot f(\hat{\mathbf{w}}_{kh}^T \mathbf{z}_{k-1,N})\right) \hat{\mathbf{w}}_{kh}||_2^2 \cdot T_z^2 \cdot T_z^2 \tag{15}$$

$$\leq \frac{T_z^4}{8} \left(T_z^2 ||\mathbf{w}_k||_F^2 + \sum_{h=1}^{D^k_{out}} || \left(4\beta^2_{\hat{\mathbf{w}}_{kh}} + \alpha_{\hat{\mathbf{w}}_{kh}}\right) \hat{\mathbf{w}}_{kh}||_2^2 \right) \tag{16}$$

$$\tag{17}$$

Appendix F: sensitivity of $\mathbf{C}_k$

The sensitivity of $\Delta\mathbf{C}_k$ is given by

$$\begin{aligned}\Delta\mathbf{C}_k &= \max_{|\mathcal{D}\backslash\mathcal{D}'|=1} ||\mathbf{C}_k(\mathcal{D}) - \mathbf{C}_k(\mathcal{D}')||_F, \\ &= \max_{|\mathcal{D}\backslash\mathcal{D}'|=1} \left(\sum_{h=1}^{D_{out}^k} ||\mathbf{C}_{kh}(\mathcal{D}) - \mathbf{C}_{kh}(\mathcal{D}')||_F^2 \right)^{\frac{1}{2}} \end{aligned} \tag{18}$$

where

$$\begin{aligned}\mathbf{C}_{kh}(\mathcal{D}) &= \tfrac{1}{2N} \sum_{n=1}^{N} \tfrac{1}{2} \partial^2 T_{nkh}, \\ &= \tfrac{1}{2N} \sum_{n=1}^{N} \left[-2z_{kh,n} f''(\hat{\mathbf{w}}_{kh}{}^{\top}\mathbf{z}_{k-1,n}) + 2\{f'(\hat{\mathbf{w}}_{kh}{}^{\top}\mathbf{z}_{k-1,n})\}^2 + 2f(\hat{\mathbf{w}}_{kh}{}^{\top}\mathbf{z}_{k-1,n}) f''(\hat{\mathbf{w}}_{kh}{}^{\top}\mathbf{z}_{k-1,n}) \right] \mathbf{z}_{k-1,n}\mathbf{z}_{k-1,n}{}^{\top}.\end{aligned}$$

Due to the triangle inequality,

$$\begin{aligned}\Delta\mathbf{C}_k &\le \frac{1}{2N} \max_{\mathbf{z}_{k,N},\mathbf{z}_{k-1,N}} \left(\sum_{h=1}^{D_{out}^k} ||\Delta\mathbf{C}_{kh}||_F^2 \right)^{1/2} \\ &= \frac{1}{2N} \max_{\mathbf{z}_{k,N},\mathbf{z}_{k-1,N}} \left(\sum_{h=1}^{D_{out}^k} ||\Big(-2z_{kh,N} f''(\hat{\mathbf{w}}_{kh}^T\mathbf{z}_{k-1,N}) + 2\left(f'(\hat{\mathbf{w}}_{kh}^T\mathbf{z}_{k-1,N})\right)^2 + 2f(\hat{\mathbf{w}}_{kh}^T\mathbf{z}_{k-1,N}) f''(\hat{\mathbf{w}}_{kh}^T\mathbf{z}_{k-1,N}) \Big) \mathbf{z}_{k-1,N}\mathbf{z}_{k-1,N}^T||_F^2 \right)^{1/2} \\ &\le \frac{T_z^2}{N} \max_{\mathbf{z}_{k,N},\mathbf{z}_{k-1,N}} \left(\sum_{h=1}^{D_{out}^k} |z_{kh,N} f''(\hat{\mathbf{w}}_{kh}^T\mathbf{z}_{k-1,N})|^2 + |\left(f'(\hat{\mathbf{w}}_{kh}^T\mathbf{z}_{k-1,N})\right)^2 + f(\hat{\mathbf{w}}_{kh}^T\mathbf{z}_{k-1,N}) f''(\hat{\mathbf{w}}_{kh}^T\mathbf{z}_{k-1,N})|^2 \right)^{1/2} \\ &\le \frac{T_z^2}{4N} \left(T_z^2 + ||4\left(\boldsymbol{\beta}_{\hat{\mathbf{w}}_k}\right)^2 + \boldsymbol{\alpha}_{\hat{\mathbf{w}}_k}||_2^2 \right)^{1/2}\end{aligned}$$

Appendix G: sensitivity of coefficients in the output layer objective function

$$\begin{aligned}
\Delta a_o &\leq \frac{1}{2N} \max_{\mathbf{z}_{K,N}} \left| \sum_{h=1}^{D_{out}^o} f(\hat{\mathbf{w}}_{K+1h}^T \mathbf{z}_{K,N}) - \mathbf{w}_{K+1h}^T f'(\hat{\mathbf{w}}_{K+1h}^T \mathbf{z}_{K,N}) \mathbf{z}_{K,N} + 1/2 \mathbf{w}_{K+1h}^T f''(\hat{\mathbf{w}}_{K+1h}^T \mathbf{z}_{K,N}) \mathbf{z}_{K,N} \mathbf{z}_{K,N}^T \mathbf{w}_{K+1h} \right| \\
&\leq \frac{1}{2N} \max_{\mathbf{z}_{K,N}} \sum_{h=1}^{D_{out}^o} \left| f(\hat{\mathbf{w}}_{K+1h}^T \mathbf{z}_{K,N}) \right| + \sum_{h=1}^{D_{out}^o} \left| \mathbf{w}_{K+1h}^T f'(\hat{\mathbf{w}}_{K+1h}^T \mathbf{z}_{K,N}) \mathbf{z}_{K,N} \right| + \sum_{h=1}^{D_{out}^o} \left| 1/2 \mathbf{w}_{K+1h}^T f''(\hat{\mathbf{w}}_{K+1h}^T \mathbf{z}_{K,N}) \mathbf{z}_{K,N} \mathbf{z}_{K,N}^T \mathbf{w}_{K+1h} \right| \\
&\leq \frac{1}{2N} \left(||\boldsymbol{\alpha}_{\mathbf{w}_{K+1}}||_1 + T_z \sum_{h=1}^{D_{out}^o} ||\mathbf{w}_{K+1h} \cdot \beta_{\mathbf{w}_{K+1h}}||_2 + \frac{T_z^2}{8} ||\mathbf{w}_{K+1}||_F^2 \right)
\end{aligned}$$

$$\begin{aligned}
\Delta \mathbf{b}_o &\leq \frac{1}{2N} \max_{\mathbf{y}, \mathbf{z}_{K,N}} \left(\sum_{h=1}^{D_{out}^o} || - y_h \mathbf{z}_{K,N} + f'(\hat{\mathbf{w}}_{K+1h}^T \mathbf{z}_{K,N}) \mathbf{z}_{K,N} - f''(\hat{\mathbf{w}}_{K+1h}^T \mathbf{z}_{K,N}) \mathbf{z}_{K,N} \mathbf{z}_{K,N}^T \mathbf{w}_{K+1h} ||_2^2 \right)^{1/2} \\
&\leq \frac{1}{2N} \max_{\mathbf{y}, \mathbf{z}_{K,N}} \left(\sum_{h=1}^{D_{out}^o} ||y_h \mathbf{z}_{K,N}||_2^2 + \sum_{h=1}^{D_{out}^o} ||f'(\hat{\mathbf{w}}_{K+1h}^T \mathbf{z}_{K,N}) \mathbf{z}_{K,N}||_2^2 + \sum_{h=1}^{D_{out}^o} ||f''(\hat{\mathbf{w}}_{K+1h}^T \mathbf{z}_{K,N}) \mathbf{z}_{K,N} \mathbf{z}_{K,N}^T \mathbf{w}_{K+1h}||_2^2 \right)^{1/2} \\
&\leq \frac{1}{2N} \left(D_{out}^o T_z^2 + T_z^2 \cdot ||\boldsymbol{\beta}_{\mathbf{w}_{K+1}}||_2^2 + 1/16 \cdot T_z^4 \cdot ||W||_F^2 \right)^{1/2}
\end{aligned}$$

$$\begin{aligned}
\Delta \mathbf{C}_o &\leq \frac{1}{2N} \max_{\mathbf{z}_{K,N}} \left(\sum_{h=1}^{D_{out}^o} ||1/2 f''(\hat{\mathbf{w}}_{K+1h}^T \mathbf{z}_{K,N}) \mathbf{z}_{K,N} \mathbf{z}_{K,N}^T||_F^2 \right)^{1/2} \\
&\leq \frac{1}{2N} \max_{\mathbf{z}_{K,N}} \left(\sum_{h=1}^{D_{out}^o} ||1/8 \mathbf{z}_{K,N} \mathbf{z}_{K,N}^T||_F^2 \right)^{1/2} \\
&\leq \frac{1}{16N} \sqrt{D_{out}^o} T_z^2
\end{aligned}$$

Appendix H: Computing a cumulative privacy loss

Preliminary

We first address how the level of perturbation in the coefficients affects the level of privacy in the resulting estimate. Suppose we have an objective function that's quadratic in w, i.e.,

$$E(w) = a + bw + cw^2, \tag{19}$$

where only the coefficients a, b, c contain the information on the data (not anything else in the objective function is relevant to data). We perturb the coefficients to ensure the coefficients are collectively (ϵ, δ)-differentially private.

$$\tilde{a} = a + n_a, \text{ where } n_a \sim \mathcal{N}(0, \Delta_a^2 \sigma^2), \tag{20}$$

$$\tilde{b} = b + n_b, \text{ where } n_b \sim \mathcal{N}(0, \Delta_b^2 \sigma^2), \tag{21}$$

$$\tilde{c} = c + n_c, \text{ where } n_c \sim \mathcal{N}(0, \Delta_c^2 \sigma^2), \tag{22}$$

where $\Delta_a, \Delta_b, \Delta_c$ are the sensitivities of each term, and σ is a function of ϵ and δ. Here "collectively" means composing the perturbed $\tilde{a}, \tilde{b}, \tilde{c}$ results in (ϵ, δ)-DP. For instance, if one uses the linear composition method (privacy degrades with the number of compositions), and perturbs each of these with ϵ_a, ϵ_b, and ϵ_c, then the total privacy loss should match the sum of these losses, i.e., $\epsilon = \epsilon_a + \epsilon_b + \epsilon_c$. In this case, if one allocates the same privacy budget to perturb each of these coefficients, then $\epsilon_a = \epsilon_b = \epsilon_c = \epsilon/3$. The same holds for δ.

However, if one uses more advanced composition methods and allocates the same privacy budget for each perturbation, per-perturbation budget becomes some function (denoted by g) of total privacy budget ϵ, i.e., $\epsilon_a = \epsilon_b = \epsilon_c = g(\epsilon)$, where $g(\epsilon) \geq \epsilon/3$. So, per-perturbation for a, b, c has a higher privacy budget to spend, resulting in adding less amount of noise.

Whatever composition methods one uses to allocate the privacy budget in each perturbation of those coefficients, since the objective function is a simple quadratic form in w, the resulting estimate of w is some function of those perturbed coefficients, i.e., $\hat{w} = h(\tilde{a}, \tilde{b}, \tilde{c})$. Since the data are summarized in the coefficients and the coefficients are (ϵ, δ)-differentially private, the function of these coefficients is also (ϵ, δ)-differentially private.

One could write the perturbed objective as

$$\tilde{E}(w) = \tilde{a} + \tilde{b}w + \tilde{c}w^2, \tag{23}$$

$$= (a + bw + cw^2) + (n_a + n_b w + n_c w^2), \tag{24}$$

$$= E(w) + n(w). \tag{25}$$

Note that we write down the noise term as $n(w)$ to emphasize that when we optimize this objective function, the noise term also contributes to the gradient with respect to w (not just the term $E(w)$).

If we denote some standard normal noise $\alpha \sim \mathcal{N}(0, 1)$, we can rewrite the noise term as

$$n(w) = (\Delta_a + \Delta_b w + \Delta_c w^2)\sigma\alpha, \tag{26}$$

which is equivalent to

$$n(w) \sim \mathcal{N}(0, (\Delta_a + \Delta_b w + \Delta_c w^2)^2 \sigma^2), \tag{27}$$

$$\sim \mathcal{N}(0, \Delta_{E(w)}^2 \sigma^2) \tag{28}$$

where we denote $\Delta_{E(w)} = \Delta_a + \Delta_b w + \Delta_c w^2$.

Extending the preliminary to DP-MAC

In the framework of DP-MAC, given a mini-batch of data $\mathcal{D}_q$ with a sampling rate q, the DP-mechanism we introduce first computes layer-wise objective functions (k layer-wise objective functions for a model with k layers, including the output layer), then noise up the coefficients of each of the layer-wise objective functions using Gaussian noise, and then outputs the vector of perturbed objective function, given by:

$$\mathcal{M}(\mathcal{D}_q) = \begin{bmatrix} E_1(w_1) \\ \cdots \\ E_K(w_K) \end{bmatrix} + \begin{bmatrix} n_1^*(w_1) \\ \cdots \\ n_K^*(w_K) \end{bmatrix}. \tag{29}$$

We denote the noise terms by $n_k^*(w_k)$ and the sensitivities of each objective by $\Delta_{E_1(w_1)}, \cdots, \Delta_{E_k(w_k)}$, where each of these is a function of $\Delta_{a_k}, \Delta_{b_k}, \Delta_{c_k}$, and w_k.

Here the question is, if we decide to use an advanced composition method such as moments accountant, how the log-moment of the privacy loss random variable composes in this case. To directly use the composition theorem of Abadi et al, we need to draw a fresh noise whenever we have a new subsampled data. This means, there should be an instance of Gaussian mechanism that affects the these noise terms simultaneously.

To achieve this, we rewrite the vector of perturbed objectives as $\tilde{\mathbf{E}}(\mathbf{w})$ as below. Then, we scale down each objective function by its own sensitivity times $\sqrt{K}$, so that the concatenated vector's sensitivity becomes just 1. Then, add the standard normal noise to the vectors with scaled standard deviation, σ. Then, scale up each perturbed quantities by its own sensitivity times $\sqrt{K}$.

$$\tilde{\mathbf{E}}(\mathbf{w}) = \begin{bmatrix} \tilde{E}_1(w_1) \\ \cdots \\ \tilde{E}_K(w_K) \end{bmatrix}, \tag{30}$$

$$= \begin{bmatrix} E_1(w_1) \\ \cdots \\ E_K(w_K) \end{bmatrix} + \begin{bmatrix} n_1^*(w_1) \\ \cdots \\ n_K^*(w_K) \end{bmatrix}, \tag{31}$$

$$= \begin{bmatrix} \sqrt{K}\Delta_{E_1(w_1)} \cdot \left\{ \frac{E_1(w_1)}{\sqrt{K}\Delta_{E_1(w_1)}} + \sigma\mathcal{N}(0,1) \right\} \\ \cdots \\ \sqrt{K}\Delta_{E_K(w_K)} \cdot \left\{ \frac{E_K(w_K)}{\sqrt{K}\Delta_{E_K(w_K)}} + \sigma\mathcal{N}(0,1) \right\} \end{bmatrix}, \tag{32}$$

$$= \begin{bmatrix} \sqrt{K}\Delta_{E_1(w_1)} \\ \cdots \\ \sqrt{K}\Delta_{E_K(w_K)} \end{bmatrix} \cdot \left(\begin{bmatrix} \frac{E_1(w_1)}{\sqrt{K}\Delta_{E_1(w_1)}} \\ \cdots \\ \frac{E_K(w_K)}{\sqrt{K}\Delta_{E_K(w_K)}} \end{bmatrix} + \mathcal{N}(0, \sigma^2 I) \right) \tag{33}$$

Since we're adding independent Gaussian noise under each subsampled data, the privacy loss after T steps, is simply following the composibility theorem in the Abadi et al paper.

So compared to the sensitivity for $n(w)$ in the first section, the new noise $n^*(w)$ has a higher sensitivity due to the factor $\sqrt{K}$.

Moments Calculations

In this case, with a subsampling with rate q, we re-do the calculations in Abadi et al. First, let:

$$\mu_0 = \mathcal{N}(\mathbf{0}_K, \sigma^2 I), \mu_1 = \mathcal{N}(\mathbf{1}_K, \sigma^2 I) \tag{34}$$

and let μ as a mixture of the two Gaussians,

$$\mu = (1-q)\mathcal{N}(\mathbf{0}_K, \sigma^2 I) + q\mathcal{N}(\mathbf{1}_K, \sigma^2 I). \tag{35}$$

Here $\mathbf{0}_K$ is the K-dimensional 0 vector, and $\mathbf{1}_K$ is the K-dimensional all ones vector. Here $\alpha_M(\lambda)$ should be $\log\max(E_1, E_2)$ where

$$E_1 = \mathbb{E}_{z\sim\mu}[(\mu(z)/\mu_0(z))^\lambda], E_2 = \mathbb{E}_{z\sim\mu_0}[(\mu_0(z)/\mu(z))^\lambda]$$

Then, we can compose further mechanisms using this particular $\alpha_M(\lambda)$, which follows the same analysis as in Abadi et al.