Machine Learning with Feature Differential Privacy

Saeed Mahloujifar [1] **Chuan Guo** [1] **Edward Suh** [1] **Kamalika Chaudhuri** [1]

Abstract

Machine learning applications incorporating differential privacy frequently face significant utility degradation. One prevalent solution involves enhancing utility through the use of publicly accessible information. Public data-points, well-known for their utility-enhancing capabilities in private training, have received considerable attention. However, it is worth noting that these public sources can vary substantially in their nature. In this work, we explore the feasibility of leveraging public features from the private dataset. For instance, consider a tabular dataset in which some features are publicly accessible while others need to be kept private. We delve into this scenario, defining a concept we refer to as *feature-DP*. We examine feature DP in the context of private optimization, and propose a solution based the widely used DP-SGD framework. Notably, our framework maintains the advantage of privacy amplification through sub-sampling, even while some features are disclosed. We analyze our algorithm for Lipschitz and convex loss functions and we establish privacy and excess empirical risk bounds. Importantly, due to our strategy's ability to harness privacy amplification via sub-sampling, our excess risk bounds converge to zero as the number of data points increases. This enables us to improve upon previously understood excess risk bounds for label differential privacy, and provides a response to an open question proposed by (Ghazi et al., 2021). We applied our methodology to the Purchase100 dataset, finding that the public features facilitated by our framework can indeed improve the balance between utility and privacy.

1. Introduction

A principal catalyst for the advancement of machine learning resides in the availability of high-quality data, serving as the basis for training models. In certain circumstances the data for training these models contain private attributes requiring protection. Differential privacy has emerged as a standard strategy for handling such situations. Differential privacy mandates the outcome of the training mechanism to exhibit statistical "smoothness" with respect to changes in the training set. A mechanism M is said to be (ϵ, δ)-DP if, for all neighboring datasets D and D' differing by a single data point, and for all events S defined on the output of the mechanism, the following inequality holds:

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

As inferred from the definition, this approach aims to regulate an adversary's capability to distinguish whether a particular data-point was incorporated in the dataset or not s(Dwork, 2006). The objective of our work is to relax this definition and control the adversary's information access up to a certain level of permissible leakage. Specifically, rather than permitting the neighboring datasets D and D' to be chosen in the worst-case scenario, we necessitate the differing points to be identical for some of their features. This requirement imposes an increased challenge for the adversary to differentiate between D and D', but simultaneously obligates us to willingly expose those features. This concept is formalized in the following definition:

Definition 1.1 (Feature Differential Privacy)**.** Let $f : X \to F$ be a feature that could be inferred from X. We say a mechanism M is (ϵ, δ) feature DP relative to f if for all datasets D and D' where

$$\exists x, x'; D \setminus D' = \{x'\} \text{ and } D' \setminus D = \{x\} \text{ and } f(x) = f(x')$$

we have

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

In this definition, f represents a part of the data points that can be disclosed. For instance, imagine a tabular dataset of individuals wherein the education level of each individual is anticipated to be publicly available. In this case, we can assign the feature f as the education level, and the definition of feature-DP necessitates that the adversary should be

[1] Meta AI. Correspondence to: Saeed Mahloujifar <saeedm@meta.com>.

Federated Learning and Analytics in Practice: Algorithms, Systems, Applications, and Opportunities Workshop at ICML 2023, Honolulu, Hawaii, USA.

unable to infer whether a data-point x or x' was used in the training set, given that x and x' have the same level of education, known to the adversary. As we increase the amount of information in f, the feature DP definition becomes a weaker notion of privacy.

One variant of this notion, termed "label differential privacy," where the adversary has knowledge of all the features except the label, has been extensively studied(Ghazi et al., 2021; Malek Esmaeili et al., 2021). However, we contend that this notion holds broader applications and warrants further exploration for other features. For example, consider a federated learning scenario where devices can transmit readings from some sensors openly, but others containing sensitive data necessitate protection. In an example case, a device's location might require privacy, while the speed or orientation of the device is permissible to disclose.

In our study, we attempt to devise a general-purpose optimization algorithm that mirrors the well-known DP-SGD(Song et al., 2013). Our primary focus is to design an algorithm that can exploit amplification by sub-sampling, an achievement that has eluded current algorithms developed for label-differential privacy (Ghazi et al., 2021; Malek Esmaeili et al., 2021; Tang et al., 2022).

2. Optimization with feature differential privacy

Assuming a dataset D, and a loss function ℓ, we target the resolution of the following optimization problem with feature differential privacy pertaining to a leak fed feature f:

$$\min_{\theta} \sum_{x \in D} \ell(x, \theta).$$

DP-SGD, the deferentially private variant of SGD, is an algorithm designed to resolve this optimization with DP assurances. A single iteration of DP-SGD involves computing the gradient of the loss function on a randomly chosen subset of data points, aggregating these to yield the gradient of the total loss function on the batch, and subsequently introducing noise to the aggregated gradient. Under the assumptions concerning the Lipschitz constant of the loss function ℓ, and the magnitude of the added noise, DP for each optimization step can be attained. The random batch selection further amplifies the privacy of each step, utilizing known privacy amplification by sub-sampling results (Balle et al., 2018). Concurrently, considering that the addition of noise and sub-sampling doesn't bias the optimization, one can also obtain proven guarantees for DP-SGD's convergence, under the right set of assumptions on the loss function (Bassily et al., 2019; Kifer et al., 2012).

$$\min_{\theta} \sum_{x \in D} \Big(\ell(x, \theta) - \ell'(f(x), \theta)\Big) + \sum_{x \in D} \ell'(f(x), \theta)$$

This way, we hope to disentangle the signal from the public and private features. Specifically, our goal is to ensure that $\ell(x, \theta) - \ell'(f(x), \theta)$ possesses a smaller Lipschitz constant than that of ℓ to improve the privacy analysis. Although it might appear that we can instantly attain better privacy after this operation, given that we've reduced the Lipschitz constant of the loss function applied to private features, we can't implement DP-SGD and analyze it based on the Lipschitzness of $\ell - \ell'$. The primary reason is that privacy amplification from sub-sampling becomes ineffective since we're leaking the public features of the examples in the batch, nullifying the amplification effect. In simpler terms, the adversary can precisely identify which examples were selected in the batch by observing the public features, hence no additional entropy from sub-sampling is gained. To address this challenge, we propose an alternate algorithm that employs two separate batches for the two losses. This approach is detailed in Algorithm 1.

Regularization with public features We also note that in addition to reducing the lipschitzness of the loss function, the public feature could create an inductive bias using a regularization factor. Specifically, we can instead aim to solve the following optimization problem:

$$\min_{\theta} \sum_{x \in D} \underbrace{\Big(\ell(x, \theta) - \ell'(f(x), \theta)\Big)}_{\text{Private loss}} + \sum_{x \in D} \underbrace{\ell'(f(x), \theta) + \ell_{reg}(f(D), \theta)}_{\text{Public loss}}.$$

Here, by $f(D)$ we mean the set of public features for all examples in D. For instance, in the context of label differential privacy, this regularizer could be a unsupervised loss function on the feature space of a neural network that does not depend on the label. As we will see, such a regularization will not affect the privacy analysis of our proposed algorithm. From now on, we define two general losses ℓ_{priv} and ℓ_{pub} and define our optimization algorithm based on that.

2.1. Convex optimization

n this section, we present a formal analysis of our algorithm's privacy, followed by an evaluation of its utility, specifically for convex loss functions.

Theorem 2.1. *[Privacy analysis] Assume the private loss function in Algorithm 1 is L-Lipschitz. Let*

$$L_f = \sup_{\substack{x, x' \in X \\ \mathbf{w} \in \mathcal{W}}} \| \nabla \ell_{priv}(\mathbf{w}, x) - \nabla \ell_{priv}(\mathbf{w}, x') \|_2$$

Algorithm 1 Noisy SGD with Public Features

```
Require: Public feature $f$, Dataset $D$, Batch sizes $m, m'$, Learning rate $\eta$, standard deviation $\sigma$, Projection space $\mathcal{W} \in R^d$, Loss functions $\ell_{priv}, \ell_{pub}$, Number of iterations $T$
Initialize $\mathbf{w}_1 \in \mathcal{W}$
for $t = 1, \dots, T$ do
    Sample a mini-batch $B_t^{priv}$ with Poisson sampling with probability $m/|D|$.
    Compute private gradient:
        $g_t^{priv} = \frac{1}{m} \sum_{x \in B_t^{priv}} \nabla \ell_{priv}(\mathbf{w}_t; x)$
    Sample a second mini-batch $B_t^{pub}$ of size $m'$ uniformly at random.
    Compute public gradient:
        $g_t^{pub} = \frac{1}{m'} \sum_{x \in B_t^{pub}} \nabla \ell_{pub}(\mathbf{w}_t; f(x)).$
    Let $g_t = g_t^{pub} + g_t^{priv} + \mathcal{N}(0, \sigma^2)$
    Update $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \cdot g_t$
    Project $\mathbf{w}_{t+1}$ into the set $\mathcal{W}$: $\mathbf{w}_{t+1} = \text{Proj}_{\mathcal{W}}(\mathbf{w}_{t+1})$
end for
return $aggregate(\mathbf{w}_1, \dots, \mathbf{w}_{T+1})$
```

then, Algorithm 1 is (ϵ, δ) feature DP for $\sigma = c\frac{mL_f}{\epsilon \cdot n}\sqrt{T\log(\frac{1}{\delta})\log(\frac{T}{\delta})}$ for a constant c. Note that $L_f \leq 2 \cdot L$ by triangle inequality.

Proof. We first analyze one step of the mechanism. Let us start with a lemma about feature-DP.

Proposition 2.2. *Let f be a feature and M_λ be a parameterized mechanism that is (ϵ, δ) feature DP with respect f, for all parameters λ. Also, let M'_λ be an arbitrary parameterized mechanism that operates on $f(D)$. Then, adaptive composition of M and M' in both orders (MoM' and $M'oM$) is also (ϵ, δ) feature DP. Note that adaptive composition means that the parameter of M (or M') could be a function of the output of M' (or M).*

We defer the proof of Proposition 2.2 to Appendix. Note that in the Proposition above, it is crucial that M and M' do not share internal randomness. This is the main reason we cannot use the same batch for the public and private loss functions. We also state a proposition without proof about the connection between feature-DP and DP.

Proposition 2.3. *If a mechanism M is (ϵ, δ)-DP (based on the notion of DP with replacement), then it is also (ϵ, δ) feature DP with respect to all public features f.*

Now note that each iteration of the algorithm can be stated as MoM' where $M'(f(D))$ is the process of calculating g_t^{pub}, and $M(D)$ is a sub-sampled Gaussian mechanism that calculates $\frac{1}{m}\sum_{x \in D} \nabla \ell_{priv}(\mathbf{w}_t, x) + \mathcal{N}(g_t^{pub}, \sigma^2)$. Therefore, each step of the algorithm is as private as of a sub-sampled Gaussian mechanism with sub-sampling rate $q = m/n$ and noise multiplier σ/L_f. So, using Propositions 2.2 and 2.3 also applying the advanced composition theorem we get that the entire mechanism is (ϵ, δ)-DP for $\epsilon = c\frac{mL_f}{\sigma \cdot n}\sqrt{\log(\frac{1}{\delta})\log(\frac{T}{\delta})}$ and some constant c. □

Note that standard deviation of the noise is independent of the lipschitz constant of ℓ_{pub} and this enables to obtain a better utility for the same privacy. We now bound the excess risk of the algorithm for convex and lipschitz loss functions.

Theorem 2.4. *[Excess empirical risk] Assume $\ell = \ell_{priv} + \ell_{pub}$ is convex, and L-Lipschitz and ℓ_{priv} is L'-Lipschitz. Let $M = \max_{\mathbf{w} \in \mathcal{W}} \parallel \mathbf{w} \parallel$ Then, setting $m' = |D|$, and using uniform averaging for aggregating the final models, and setting σ based on Theorem 2.1 we have the following*

$$\mathbb{E}_{\mathbf{w} \leftarrow L(D)}[\sum_{x \in D} \ell(\mathbf{w}, x)] - \arg\min_{\mathbf{w} \in \mathcal{W}} \sum_{x \in D} \ell(\mathbf{w}, x)] \leq \frac{M^2}{2\eta T} + \eta L'^2 + c\eta \frac{m^2 L'^2 d}{\epsilon^2 \cdot n^2} T \log(\frac{1}{\delta}) \log(\frac{T}{\delta}).$$

Note that this excess empirical risk is smaller than what one can obtain from the analysis of DP-SGD (see Lemma 3.3 in (Bassily et al., 2019)) because we are working with a smaller Lipschitz constant and our noise is also smaller.

Case Study: Logistic Regression under Label-Differential Privacy In this example, we present an application of our proposed algorithm to logistic regression under the condition of label differential privacy (label-DP), illustrating the specific bounds for this scenario. Consider the logistic regression loss function for a classification problem with K labels $\{0, \dots, k-1\}$, denoted as $\ell(\mathbf{w}, x, y)$. Let us define the public loss function $\ell_{pub}(\mathbf{w}, x)$ so the we have $\nabla \ell_{pub} = x \times p^T$, where p is the probability vector, namely, $p = softmax(x \times \mathbf{w})$. Since this public loss does not depend on the label y, it can indeed be treated as public. Then, we set the private loss function $\ell_{priv}(\mathbf{w}, x, y) = \ell(\mathbf{w}, x, y) - \ell_{pub}(\mathbf{w}, x)$. Consequently, we can say that $\ell = \ell_{priv} + \ell_{pub}$. Using these definitions of private and public losses in our Algorithm 1, we can derive the concrete bounds for this scenario. The first step is to calculate the Lipschitz constants L' and L_f. Assume that the input space $X = \{x \in R^p, \|x\| \leq 1\}$. Since the gradient of the logistic loss equals $x(p-y)^T$, where p represents the prediction probability vector and y stands for the one-hot encoding of the ground truth (the label), we deduce that the gradient of ℓ_{priv} equals $-xy^T$. From this, we infer that the loss function ℓ_{priv} is 1-Lipschitz, and it also enjoys $L_f \leq 2$. In the context of label-DP, the leaked feature f is defined by $f(x, y) = x$. Applying our bounds, assuming the projection is mapping to an ℓ_2 ball of radius 1, and with $m = 1$, we obtain the excess empirical risk as $\frac{1}{1\eta T} + 4\eta\left(1 + \frac{cdT}{\epsilon^2 \cdot n2}\log(\frac{1}{\delta})\log(\frac{T}{\delta})\right)$. Comparing this bound for the bound one would obtain from analyzing DP-SGD, we are saving a factor $\sqrt{2}$ in the Lipschitz constant, therefore, the last term of the excess risk will be $8\eta\frac{cdT}{\epsilon^2 \cdot n2}\log(\frac{1}{\delta})\log(\frac{T}{\delta})$ instead of $4\eta\frac{cdT}{\epsilon^2 \cdot n2}\log(\frac{1}{\delta})\log(\frac{T}{\delta})$.

Comparison with the Excess Bound in Ghazi et al., 2021 for Label-DP The work of Ghazi et al.(Ghazi et al., 2021) presents an excess risk bound for label-DP under convex functions, thereby raising the question of whether we can leverage sub-sampling for label-DP. Our analysis above answers this query affirmatively. Their approach involves a variant of DP-SGD in which a single example is sampled at each round, and all possible labels for that example are considered to calculate the corresponding gradients. A Gaussian noise is then sampled and projected into the span of all gradients created by different labels. The resulting noise is added to the true gradient. This approach enables them to benefit from the privacy provided by the Gaussian mechanism in the label-DP setting, while reducing the variance created by noise to $K\sigma^2$, where K is the number of classes. This is in stark contrast with the conventional DP-SGD, where the noise creates a variance of $d\sigma^2$. For the logistic regression setting, their empirical excess risk is bounded by $\frac{1}{2\eta T} + 2\eta\left(1 + \frac{2cKT}{\epsilon^2}\log(\frac{1}{\delta})\log(\frac{T}{\delta})\right)$. When compared, their excess empirical loss increases by the number of classes K, while ours increases by $\frac{d}{n^2}$. The division by n^2 in our case is due to our ability to leverage amplification by sub-sampling, a capability they lack. Particularly, as long as the number of examples n is greater than $\sqrt{d/K}$, our algorithm yields a lower excess risk than theirs.

3. Experimental results

To demonstrate the performance of our algorithm, we design an experiment for feature differential privacy and demonstrate the results here.

Dataset and Architecture: We use the Purchase100 () dataset to perform our experiments. This dataset contains 600 real valued features and each example is labeled with class from a set of 100 classes. The high number of classes makes this dataset specially challenging for training with differential privacy. Our models are two-layer perceptrons with ReLU activations and 300 neurons in the middle layer.

Selecting the public feature f: As stated in the definition of feature DP (Definition 1.1), we need to specify the public features that we want to leak through a function f. For the case of purchase dataset, we select 100 out of 600 features at random and call them x_{pub}. We call the remaining features x_{priv}. We also use y to refer to the label. Then we define $f(x, y) = (x_{pub}, y)$. This means that we leak 100 public features as well as the label. Then, we want to preserve the privacy of the remaining 500 features.

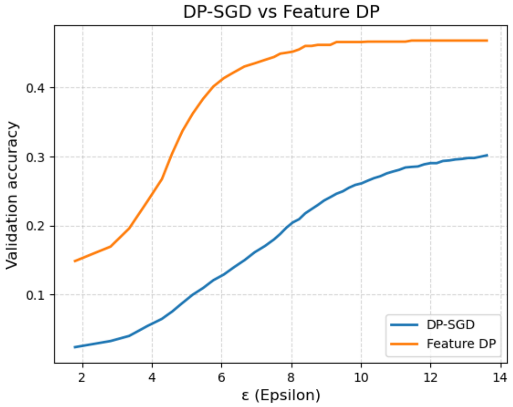

Figure 1: Comparing DP-SGD with our feature DP framework while leaking 100 random features.Our algorithm is able to leverage the public features and achieve 10-20% more accuracy in all values of ϵ.

Training: We use our Algorithm 1 to train the model. We first create a public loss function as follows. Let $g : \mathbb{R}^{100} \to \mathbb{R}^{600}$ be a Gaussian padding that extends the 100 public features into a full vector of size 600 and fills the private features with Gaussian noise $\mathcal{N}(0, 1)$. We then define the public loss function $\ell_{pub}(\mathbf{w}, f(x), y) = \ell(\mathbf{w}, g(f(x)), y)$ where ℓ is the cross entropy loss function. Then we set $\ell_{priv}(\mathbf{w}, x, y) = \ell(\mathbf{w}, x, y) - \ell_{pub}(\mathbf{w}, x, y)$. Our privacy analysis in Section 2 heavily relies on the private loss function being lipschitz. Unfortunately, we cannot guarantee this for a neural network. To achieve the same privacy, we use clipping of the gradient from the private loss function. In particular, we calculate $g_{priv} = \nabla\ell_{priv} \cdot \frac{\min(|\nabla\ell_{priv}|, C)}{|\nabla\ell_{priv}|}$ for $C = 0.01$. Then we add Gaussian noise to get $\tilde{g}_{pub} = g_{pub} + \mathcal{N}(0, C^2\sigma^2)$. We also calculate $g_{pub} = \nabla\ell_{pub}$. Now, since clipping has biased the ratio between the norm of g_{pub} and $\tilde{g}_{priv}$, we aggregate them using a ratio α, $g = g_{pub} + \alpha\tilde{g}priv$. We fix α in all iterations and tune it as a hyperparameter. We use a learning rate of 0.1 and use momentum of 0.9 to update our model.

Results: Figure 1 shows the result of our experiments as described above. As it is clear from the figure, in all values of ϵ we are able to improve the accuracy by $10 - 20\%$. This comes at the cost of leaking 100 features from the set of 600 features available. In both experiments the sampling rate is $1/16$ and the noise multiplier is set to 1.0. The optimal learning rate schedule is different for two cases and is hyperparameter tuned for best results.

References

Balle, B., Barthe, G., and Gaboardi, M. Privacy amplification by subsampling: Tight analyses via couplings and divergences. *Advances in Neural Information Processing Systems*, 31, 2018.

Bassily, R., Feldman, V., Talwar, K., and Guha Thakurta, A. Private stochastic convex optimization with optimal rates. *Advances in neural information processing systems*, 32, 2019.

Dwork, C. Differential privacy. *ICALP'06 Proceedings of the 33rd international conference on Automata, Languages and Programming - Volume Part II*, 2006.

Ghazi, B., Golowich, N., Kumar, R., Manurangsi, P., and Zhang, C. Deep learning with label differential privacy. *Advances in neural information processing systems*, 34: 27131–27145, 2021.

Kifer, D., Smith, A., and Thakurta, A. Private convex empirical risk minimization and high-dimensional regression. In *Conference on Learning Theory*, pp. 25–1. JMLR Workshop and Conference Proceedings, 2012.

Malek Esmaeili, M., Mironov, I., Prasad, K., Shilov, I., and Tramer, F. Antipodes of label differential privacy: Pate and alibi. *Advances in Neural Information Processing Systems*, 34:6934–6945, 2021.

Song, S., Chaudhuri, K., and Sarwate, A. D. Stochastic gradient descent with differentially private updates. In *2013 IEEE global conference on signal and information processing*, pp. 245–248. IEEE, 2013.

Tang, X., Nasr, M., Mahloujifar, S., Shejwalkar, V., Song, L., Houmansadr, A., and Mittal, P. Machine learning with differentially private labels: Mechanisms and frameworks. *Proceedings on Privacy Enhancing Technologies*, 4:332–350, 2022.

A. Proof of Proposition 2.2

We first restate the Proposition.

Proposition A.1. *Let f be a feature and M_λ be a parameterized mechanism that is (ϵ, δ) feature DP with respect f, for all parameters λ. Also, let M'_λ be an arbitrary parameterized mechanism that operates on $f(D)$. Then, adaptive composition of M and M' in both orders (MoM' and $M'oM$) is also (ϵ, δ) feature DP. Note that adaptive composition means that the parameter of M (or M') could be a function of the output of M' (or M).*

Proof. We have

$$\begin{aligned}
\Pr[MoM'(D) \in S] &= \mathbb{E}_{\lambda\sim\Lambda(M'(f(D)))} \Pr[M_\lambda(D) \in S] \\
&= \mathbb{E}_{\lambda\sim\Lambda(M'(f(D')))} \Pr[M_\lambda(D) \in S] \\
&\leq \mathbb{E}_{\lambda\sim\Lambda(M(f(D')))} e^\epsilon \Pr[M_\lambda(D') \in S] + \delta \\
&= e^\epsilon \Pr[MoM'(D') \in S] + \delta.
\end{aligned}$$

Similarly, for the other direction we have,

$$\begin{aligned}
\Pr[M'oM(D) \in S] &= \mathbb{E}_{\lambda\sim\Lambda(M(f(D)))} \Pr[M'_\lambda(f(D)) \in S] \\
&= \mathbb{E}_{\lambda\sim\Lambda(M(f(D)))} \Pr[M'_\lambda(f(D')) \in S] \\
&\leq \mathbb{E}_{\lambda\sim\Lambda(M(D'))} e^\epsilon \Pr[M_\lambda(f(D')) \in S] + \delta \\
&= e^\epsilon \Pr[M'oM(D') \in S] + \delta.
\end{aligned}$$

□

B. Proof of Theorem 2.4

Theorem 2.4. *[Excess empirical risk] Assume $\ell = \ell_{priv} + \ell_{pub}$ is convex, and L-Lipschitz and ℓ_{priv} is L'-Lipschitz. Let $M = \max_{\mathbf{w}\in\mathcal{W}} \parallel \mathbf{w} \parallel$ Then, setting $m' = |D|$, and using uniform averaging for aggregating the final models, and setting σ based on Theorem 2.1 we have the following*

$$\mathbb{E}_{\mathbf{w}\leftarrow L(D)}[\sum_{x\in D} \ell(\mathbf{w}, x)] - \operatorname*{arg\,min}_{\mathbf{w}\in\mathcal{W}} \sum_{x\in D} \ell(\mathbf{w}, x)] \leq \frac{M^2}{2\eta T} + \eta L'^2 + c\eta \frac{m^2 L'^2 d}{\epsilon^2 \cdot n^2} T \log(\frac{1}{\delta}) \log(\frac{T}{\delta}).$$

Proof. First note that the gradient g_t is and unbiased estimate of the empirical gradient by linearity of expectation. We can also bound the variance of the estimate by the variance from sub-sampling, which is L'^2/m and the variance of the added noise, which is $d\sigma^2$. Therefore, applying the standard stochastic gradient oracle techniques for analyzing the convergence of SGD, we can bound the gap between between the empirical risk of the optimal and the obtained model to be at most

$$\frac{M^2}{2\eta T} + \eta L'^2 + \eta\sigma^2 d.$$

Setting σ according to Theorem 2.1, we get the empirical excess risk

$$\frac{M^2}{2\eta T} + \eta L'^2 + c\eta \frac{m^2 L'^2 d}{\epsilon^2 \cdot n^2} T \log(\frac{1}{\delta}) \log(\frac{T}{\delta}).$$

□