1. Analysis of Private Gradient Descent

We consider optimizing a function using Differentially Private Gradient Descent (DP-GD). The presence of noise in GD introduces a deviation between the iterates of GD with noise, denoted as w_T, and without noise, denoted as w_{T_b}, at iteration T. We first upper bound this deviation in expectation, which we refer to as the radius r. We then use this to upper bound the excess empirical risk of noisy GD. We finally use this bound to motivate the design of our private adaptive HPO method.

1.1. Assumptions

We present four assumptions that simplify the convergence analysis. We acknowledge that these assumptions do not hold true in all settings, but nevertheless provide an important foundation for illustrating the intuition of our method. We empirically validate the success of our algorithm in complex neural network settings, such as training a 13B-parameter OPT Transformer model on the benchmark SQuAD task, in Section 3.

- A function is α-strongly convex if for any two points x, y and any subgradient g at x, it holds that $f(y) \geq f(x) + g^\top(y - x) + \frac{\alpha}{2}\|y - x\|^2$.
- A function is β-smooth if its gradient is β-Lipschitz continuous, meaning for any two points x, y, $\|\nabla f(x) - \nabla f(y)\| \leq \beta\|x - y\|$.
- A function is L-Lipschitz if there exists a positive L such that $|f(x) - f(y)| \leq L\|x - y\|$
- A function satisfies the *bounded gradient assumption* if there exists a constant C such that $E[\|\nabla f(w)\| \leq C \; \forall w \in R^d$

The bounded gradient assumption is implied by convexity and Lipschitzness. This allows us to ignore the impact of clipping in DP-SGD, which reduces the analysis to that of noisy GD.

The noise added at each iteration for privacy has an expected norm $\rho = \sqrt{d} \cdot \sigma$, where d is the dimension of the model, and σ is the scale of the noise. The learning rate η satisfies $0 < \eta < \frac{2}{\beta}$, ensuring convergence.

Let $c = \max(|1 - \eta\alpha|, |1 - \eta\beta|)$, which characterizes the contraction factor in the optimization process. Given that η is chosen appropriately, we have $0 < c < 1$.

1.2. Definitions

The empirical loss $\mathcal{L}(w_T)$ for a model parameterized by w_T (ex. iterate T of GD) over a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ is defined as the average loss over all training samples:

$$\mathcal{L}(w_T) = \frac{1}{N}\sum_{i=1}^{N} \ell(f(x_i; w_T), y_i)$$

The goal of our private Hyperparameter Optimization (HPO) is to find the hyperparameter set Λ^* that minimizes the loss:

$$\Lambda^* = \underset{\Lambda}{\operatorname{argmin}}\, \mathcal{L}_{\text{val}}(\Lambda)$$

where $\mathcal{L}_{\text{val}}(\Lambda)$ denotes the loss on a validation dataset for a given hyperparameter configuration Λ. Because Noisy GD typically does not overfit due to the heavy regularization effect of the noise, and to make the convergence analysis straightforward, we use the empirical loss as a proxy for the validation loss throughout. We will analyze the excess empirical risk to motivate the design of our private HPO method.

Let w_T be the T^{th} iterate of noisy GD that optimizes a function satisfying the assumptions, and w_{T_b} be the T^{th} iterate of non-noisy GD that optimizes that same function. We define the excess empirical risk of noisy GD as:

$$\begin{aligned} R_{\text{noisy}} &= E[\mathcal{L}(w_T)] - \mathcal{L}(w^*), \\ &\leq E[\mathcal{L}(w_T) - \mathcal{L}(w_{T_b})] + \mathcal{L}(w_{T_b}) - \mathcal{L}(w^*) \\ &\leq E[L \cdot \|w_T - w_{T_b}\|] + R_{\text{non-noisy}} \end{aligned}$$

Where $\mathcal{L}(w^*)$ denotes the empirical loss at the optimal parameter set (without noise). $R_{\text{non-noisy}} = \mathcal{L}(w_{T_b}) - \mathcal{L}(w^*$ is the excess empirical risk of non-noisy GD. In the last line, we upper bounded the excess risk induced by noise $\mathcal{L}(w_{T_b}) - \mathcal{L}(w^*)$ by applying Lipschitzness of the loss.

We now bound $\|w_T - w_{T_b}\|$.

Theorem 1.1. *Let w_T be the T^{th} iterate of noisy GD that optimizes an α-strongly convex and β-smooth function, and let w_{T_b} be the T^{th} iterate of non-noisy GD that optimizes that same function. The ”noisy radius” distance, the ℓ_2-norm between w_T and w_{T_b} at iteration T, can be bounded in expectation as follows:*

$$E[\|w_T - w_{T_b}\|] \leq \rho\eta \times \left(\sum_{i=0}^{T-1} c^i\right) = r$$

Proof sketch. The full proof is in Appendix B.5. At each iteration the distance between the noisy iterate and the non-noisy iterate contracts by a factor of $c = \max(|1 - \eta\alpha|, |1 - \eta\beta|)$ and then increases additively by $\rho\eta$. The overall distance then can be represented by scaling the additive noise term $\rho\eta$ by a geometric series that converges. Future work might incorporate additional factors such as momentum acceleration, bias introduced by clipping, or extend our analysis to the setting of more general neural networks. However, our objective here is to provide some theoretical intuition for our algorithm.

Substituting Theorem 1.1 into the excess empirical risk:

$$R_{\text{noisy}} \leq Lr + R_{\text{non-noisy}}$$

Where L is the Lipschitz constant.

We can see that our private HPO needs to find HPs that are good for non-noisy optimization but do not create a large divergence between the noisy and non-noisy iterates.

2. Connections to Private HPO

We have established a relationship between the excess empirical risk and the noisy radius. We can now connect this back to our goal of doing private HPO, which is to find the HPs that minimize the excess empirical risk. We want to find $r^* = r(\varepsilon)$, the optimal value of r for a given value of ε. We will first *reduce the dimensionality* of the search problem and then *introduce a principled approximation*.

2.1. Reducing the Dimensionality of HPO

We want to reduce the dimensionality of HPO so that we can reduce the cost of HPO.

For fixed ε, if we increase or decrease T then we will correspondingly increase or decrease σ by the Composition Theorem of DP. The actual statements of DP composition are somewhat complicated, but we can simplify them as saying σ grows slower than αT for some constant α. Because $E[\rho] = \sqrt{d}\sigma$, we have that ρ grows slower than T.

The geometric series converges to $\dfrac{1}{1-c}$ as T increases, giving us:

$$E[||w_T - w_{T_b}||] \leq (T\eta) \cdot (\sqrt{d}\frac{1}{1-c}) \qquad (1)$$

Because we are interested in writing the radius in terms of hyperparameters that we can optimize, we drop the second term for simplicity. Now we can write our hyperparameter of interest as $r = \eta \times T$, reducing the 2D HPO to 1D. If we wanted to search for additional terms such as the batch size or clipping threshold, we could incorporate them into our theory, but we empirically find that it's best to fix all other HPs to the values we provide and just search for η, T.

2.2. Our Private HPO

In order to find the optimal $r^* = r(\varepsilon)$ without exhaustively searching, we need to approximate $r(\varepsilon)$. A natural choice is Taylor approximation. We can sample points from $r(\varepsilon)$ at different values of ε via random search, use this to approximate a Taylor polynomial, and then use that Taylor polynomial to estimate r for any desired target ε. After we have our estimated r, we can decompose it into η, T by randomly sampling η, T until their product is close to r. This is the procedure we use in Figure 2, paying for the privacy cost of building the approximation and then using it to estimate the optimal HPs for many values of $\varepsilon \in [0.5, 8]$.

We now elaborate on the implementation of the method.

The first-order Taylor approximation of a function $r(\varepsilon)$ around a point ε_0 is given by $r(\varepsilon) \approx r(\varepsilon_0) + \frac{dr}{d\varepsilon}\big|_{\varepsilon=\varepsilon_0} \cdot (\varepsilon - \varepsilon_0)$, which linearly approximates r near ε_0. Because we cannot analytically determine $\frac{dr}{d\varepsilon}$, we will have to approximate this.

To approximate the first-order Taylor polynomial we fit a line. We first use random search to find two empirical points $(\varepsilon_1, r(\varepsilon_1))$ and $(\varepsilon_2, r(\varepsilon_2))$. We then fit a line to these points to obtain the parameters of the line m, b (slope and intercept). We finally estimate the optimal $r(\varepsilon_f) = m\varepsilon_f + b$ such that the composition of privacy guarantees for the entire private HPO satisfies a target privacy budget according to Theorem 2.3. In practice we choose smaller values of ε for these points such as $\varepsilon_1 = 0.1, \varepsilon_2 = 0.2$, that we find provide a good privacy-utility tradeoff.

More generally, we can approximate the Taylor polynomial by fitting a degree N polynomial with $N + 1$ points $(\varepsilon_1, r(\varepsilon_1)) \cdots (\varepsilon_{N+1}, r(\varepsilon_{N+1}))$. We provide results comparing the linear approximation to quadratic approximation in the 2nd common response PDF, but use the linear approximation throughout our work because we find that it provides a good privacy-utility tradeoff.

2.3. Limitations

Although this theory does not hold in general for training neural networks, we quantitatively evaluate the heuristic we develop in Section 3.4 and find that our method holds even for the complex setting of training Transformers on NLP benchmarks. Our HPO also requires more runtime than random search because it is adaptive.