NO.

DATE.

1: To calculate b from the Lap(x|b) that satisfies ε-differential privacy with an l1-sensitivity of 1 we can use this formula: $b = \frac{\Delta f}{\varepsilon}$

where Δf is the l1 sensitivity of the function f that we want to compute on the data, ε is the privacy budget

In this case, we have $\Delta f = 1$. so $b = \frac{1}{\varepsilon}$

2: DPSGD is a variant of the standard SGD algorithm that adds noise to gradients to achieve differential privacy.

Here are the steps

① Initial the model parameters

② For each epoch, shuffle the training data

And for each batch:

select random data point

<1> Compute the gradient of the loss function with respect to the model parameters on the current batch

<2>: Add Laplace noise to gradient to achieve differential privacy. The noise depends on ε and the sensitivity of the gradient

<3>: randomly select unbiased gradient estimate

<4>: Update the model parameter using the noisy gradient and a learning rate

③ return the final model parameters

NO.

DATE.

3: Here is an algorithm based on the requirements:

① Compute gradient matrix $G \in R^{n \times p}$ using the training data

② Perform SVD on G to obtain $G = U \Sigma V^T$

③ Choose the top k rows and let them be $B = V_k$

④: $\hat{G} = GB$, using this formula to compress G

⑤: Doing per-example clipping and use $\hat{G}$ in DPSGD to add Gaussian noise

⑥: Inject $\hat{G}$ back to R^P using B^T

⑦: update the model parameters