Deep Learning with differential privacy

Abhishek Karkee

Fernando Merida

Background

For this project we used this research paper:

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016, October). Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security* (pp. 308-318).

Research problem studied

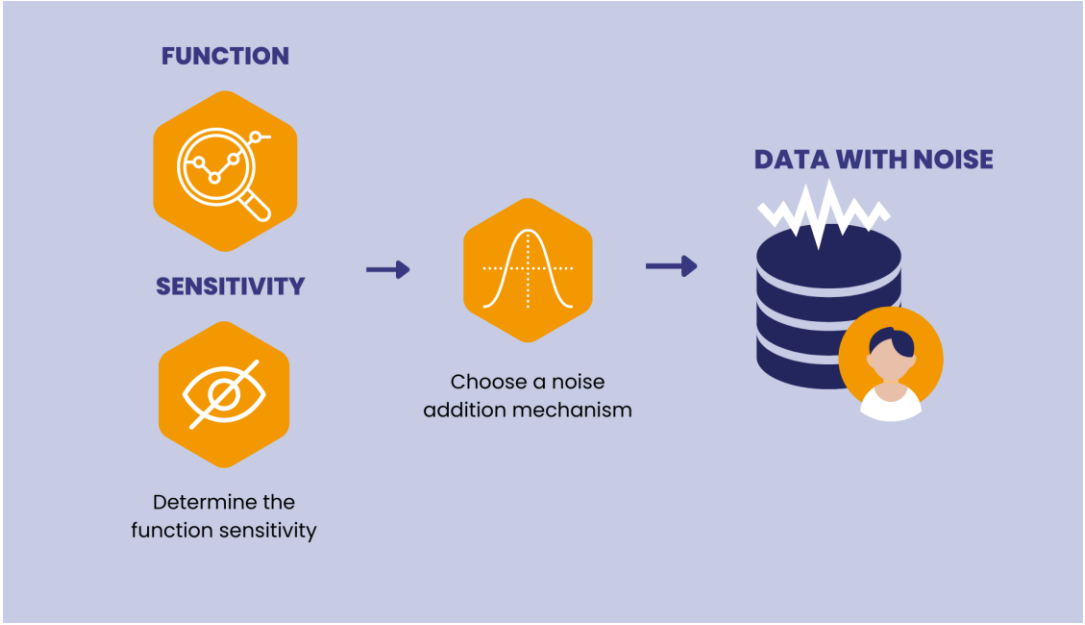
Training a machine learning model often requires large, representative datasets, which may include sensitive information.

This necessity is being address by this paper by using differential privacy.

Differential privacy

Is an approach for providing privacy while sharing information about a group of individuals, by describing the patterns within the group while withholding information about specific individuals

It works by adding statistical noise to the data(either to their inputs or outputs)



DP-SDG

The paper proposes Differentially-private descent stochastic gradient as it's approach to apply this technique

DP-SDG modifies the minibatch stochastic optimization process in order to make it differentially private.

Algorithm 1 Differentially private SGD (Outline)

Input: Examples $\{x_1, \ldots, x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i)$. Parameters: learning rate η_t, noise scale σ, group size L, gradient norm bound C.

Initialize θ_0 randomly

for $t \in [T]$ **do**

Take a random sample L_t with sampling probability L/N

Compute gradient

For each $i \in L_t$, compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t}\mathcal{L}(\theta_t, x_i)$

Clip gradient

$\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i)/\max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)$

Add noise

$\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}\left(\sum_i \bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$

Descent

$\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$

Output θ_T and compute the overall privacy cost (ε, δ) using a privacy accounting method.

Approach applied

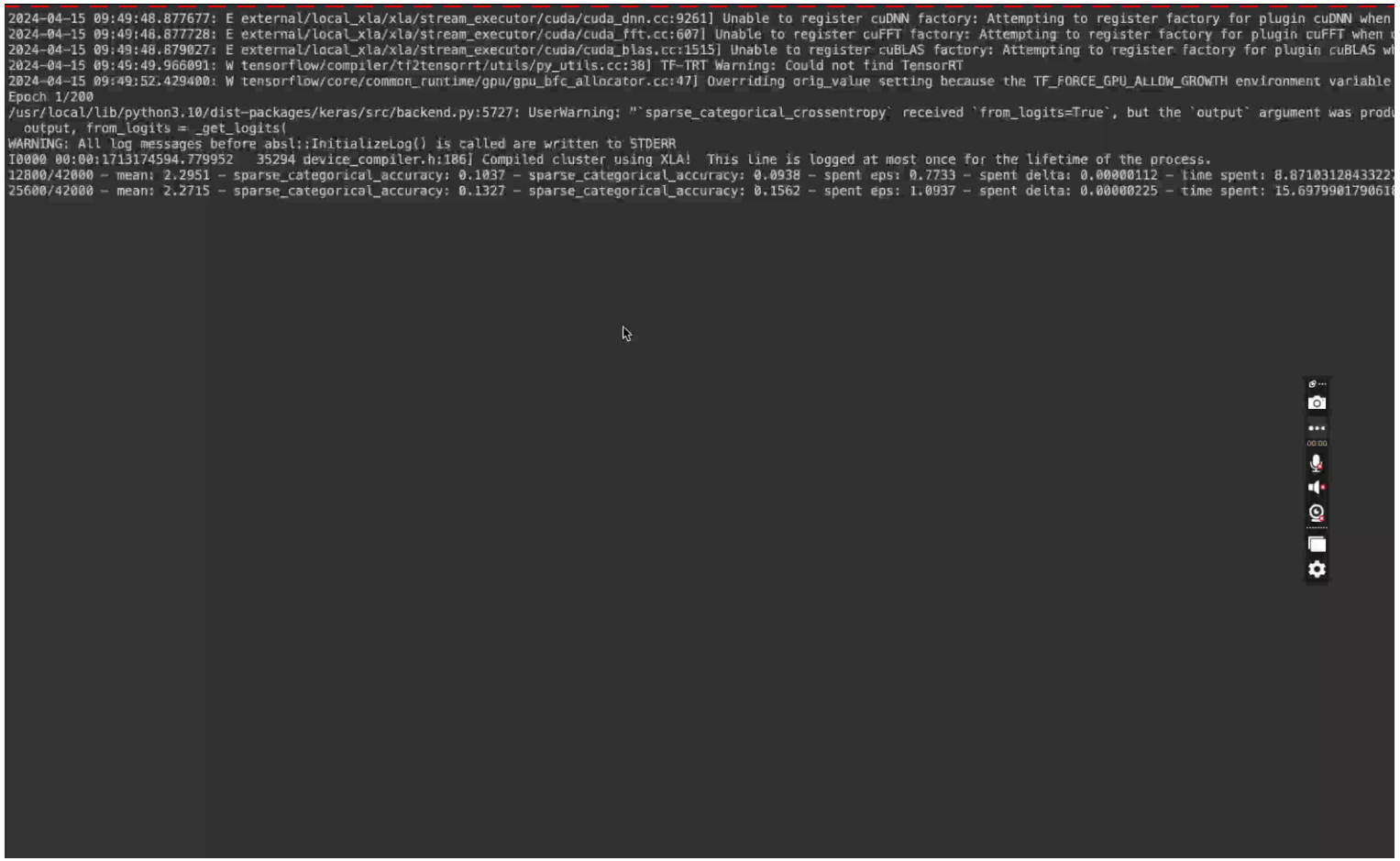
```
class DPSGD_Optimizer():
  def __init__(self, accountant, sanitizer):
    self._accountant = accountant
    self._sanitizer = sanitizer

  def Minimize(self, loss, params,
               batch_size, noise_options):
    # Accumulate privacy spending before computing
    # and using the gradients.
    priv_accum_op =
        self._accountant.AccumulatePrivacySpending(
            batch_size, noise_options)
    with tf.control_dependencies(priv_accum_op):
      # Compute per example gradients
      px_grads = per_example_gradients(loss, params)
      # Sanitize gradients
      sanitized_grads = self._sanitizer.Sanitize(
          px_grads, noise_options)
      # Take a gradient descent step
      return apply_gradients(params, sanitized_grads)

def DPTrain(loss, params, batch_size, noise_options):
  accountant = PrivacyAccountant()
  sanitizer = Sanitizer()
  dp_opt = DPSGD_Optimizer(accountant, sanitizer)
  sgd_op = dp_opt.Minimize(
      loss, params, batch_size, noise_options)
  eps, delta = (0, 0)
  # Carry out the training as long as the privacy
  # is within the pre-set limit.
  while within_limit(eps, delta):
    sgd_op.run()
    eps, delta = accountant.GetSpentPrivacy()
```

Figure 1: Code snippet of `DPSGD_Optimizer` and `DP-Train`.

Learning from the paper

- We learned different techniques to add security to a dataset by adding noise to the data. In this case using SDG
- We learned that using differential privacy does affect the performance of the training process. The one used in the paper reached 65% training accuracy instead of 97%. So, there is a tradeoff between model accuracy and privacy.
- We learned to combine both, deep learning concepts with security concepts.

Re implementation of the work



Enhancement beyond the work

Future work on this research paper could be

- Test different frameworks: we used TensorFlow privacy to implement this. Other libraries can be explored for a different approach
- Fine tune model parameter selection. Differentially Private algorithms often need to select the best amongst many candidate options.

Thank you