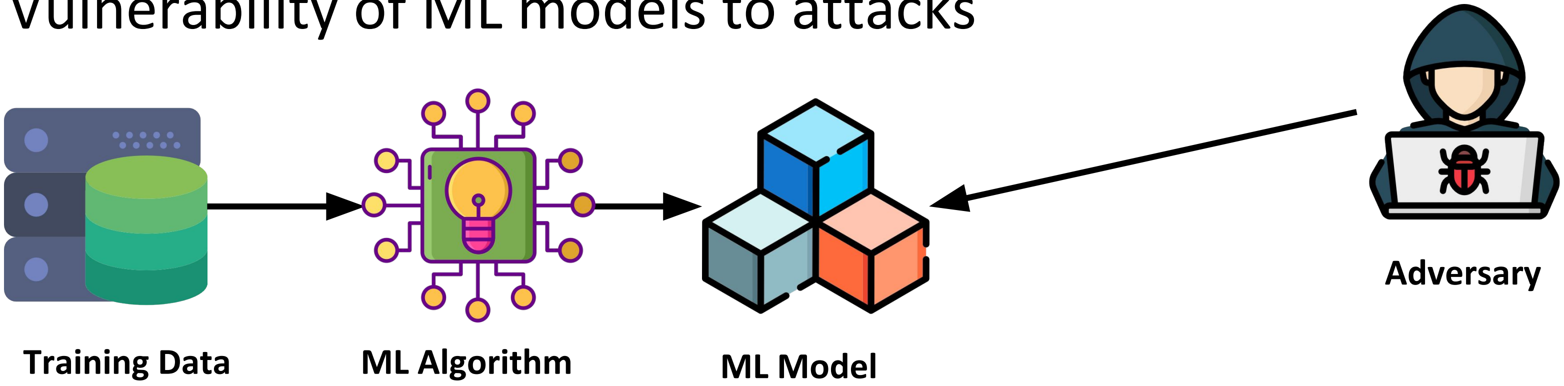
DP-CGAN: Differentially Private Synthetic Data and Label Generation

Reihaneh Torkzadehmahani(rtorkzad@ucsc.edu)
Joint work with Peter Kairouz(Google AI) and
Benedict Paten(UCSC)

Introduction

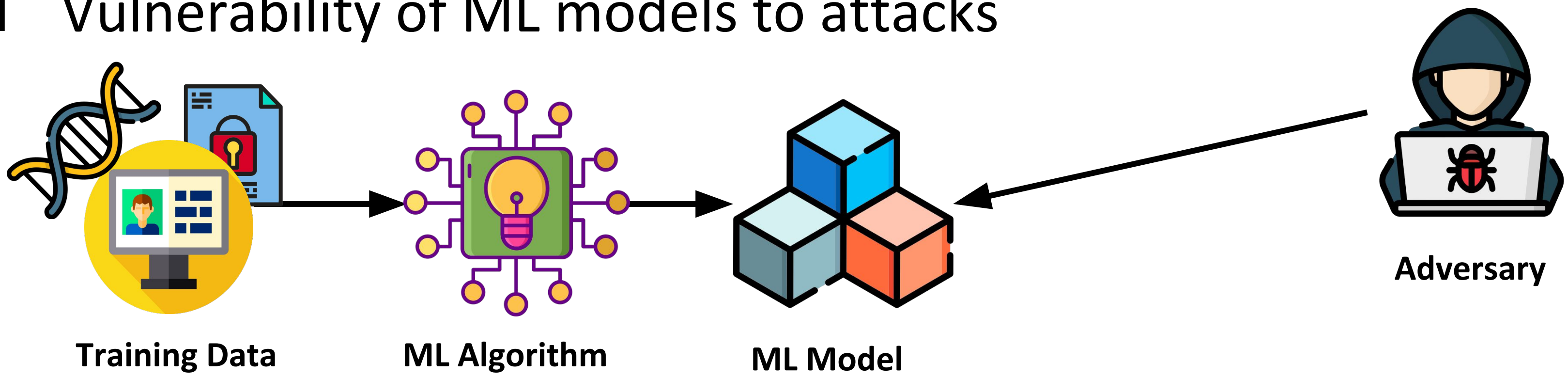
- Vulnerability of ML models to attacks



- Great performance of Generative Adversarial Networks(GANs) in various applications

Introduction

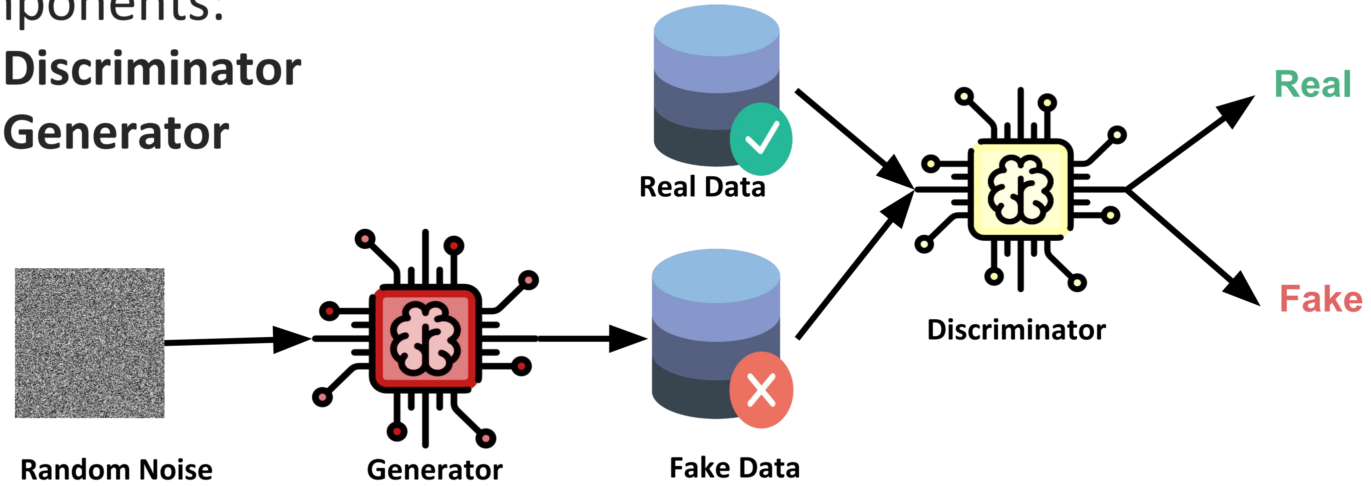
- Vulnerability of ML models to attacks



- Great performance of Generative Adversarial Networks(GANs) in various applications
- Using GANs to generate synthetic **sensitive** data

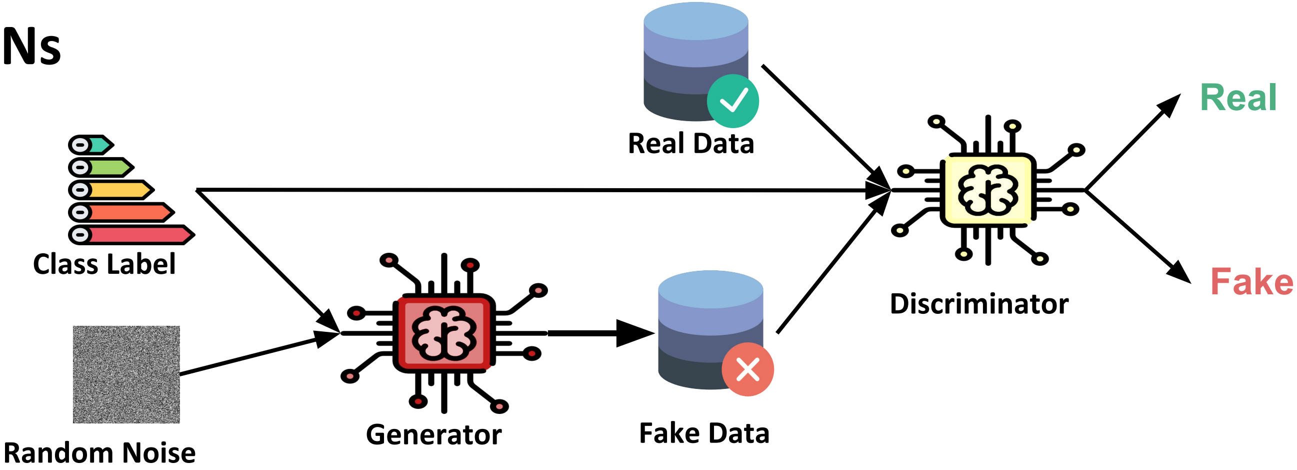
Background

- **Generative Adversarial Networks(GANs)**
  - Learn the training data distribution and generate synthetic data
  - Components:
    - **Discriminator**
    - **Generator**
    -



Background(cnt'd)

❑ Conditional GANs



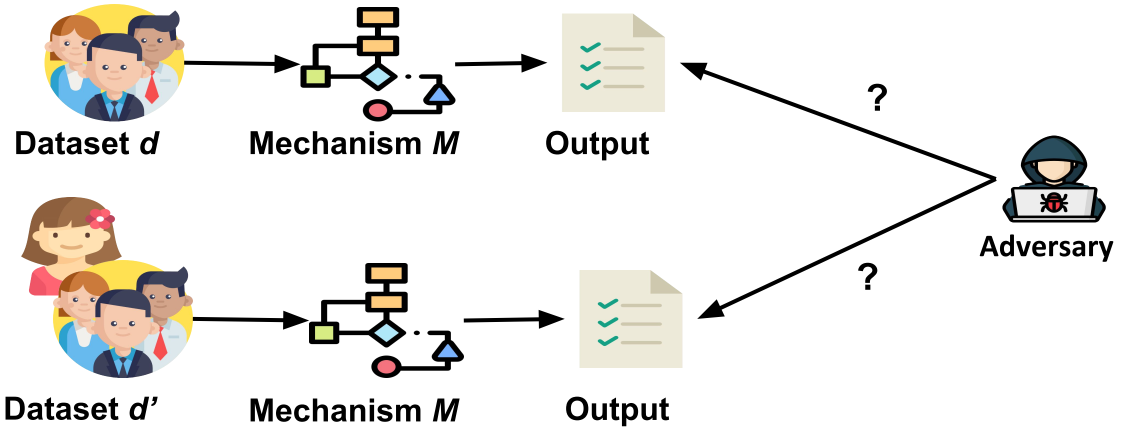
❑ Objective Function:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x}|\boldsymbol{y})] + \mathbb{E}_{\boldsymbol{z}\sim p_{z}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}|\boldsymbol{y})))].$$

M. Mirza and S. Osindero. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784, 2014

Background(cnt'd)

- **Differential Privacy**



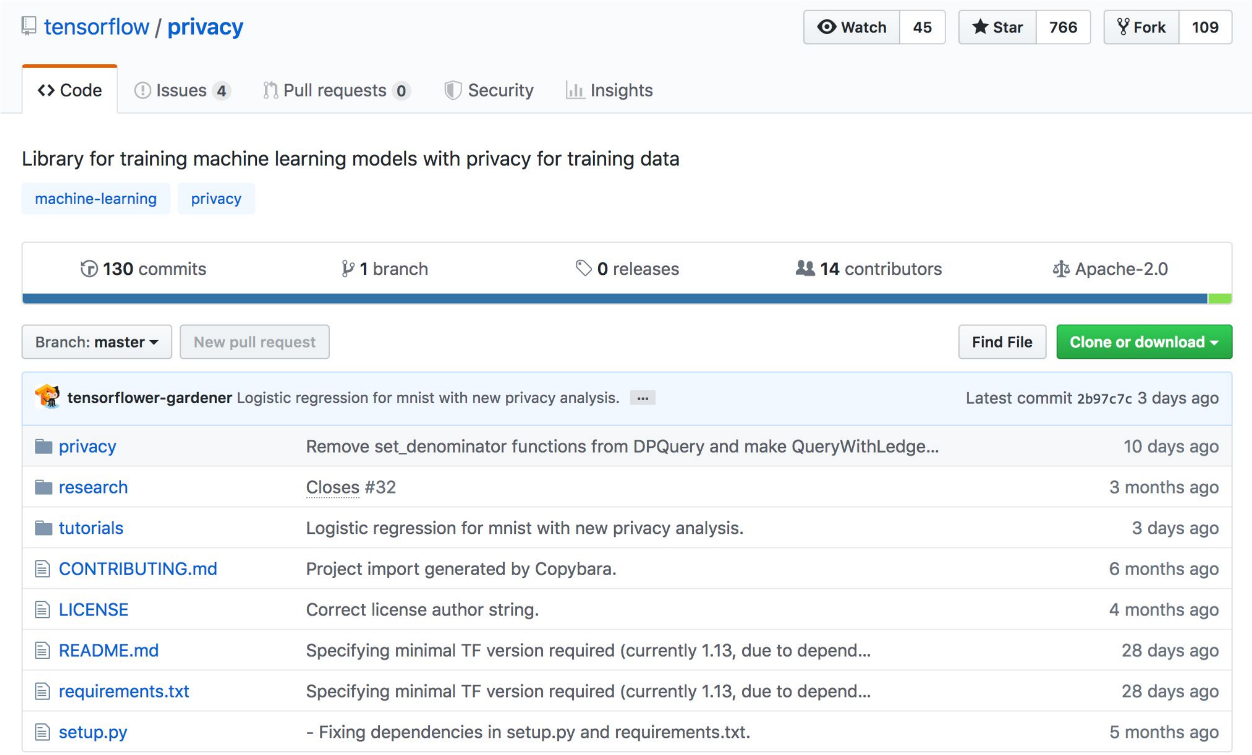
- *A randomized mechanism* ***M*** *with domain* ***D*** *and range* ***R*** *satisfies (ε,**δ**)-differential privacy if for all pairs of adjacent datasets (d , d') and for any subset* ***S*** *of output:*

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

Related Work

- Privacy-preserving Deep Learning*[Shokri et al., 2015]*
  - High privacy loss
- PATE*[Papernot et al., 2016]*
  - Assumes the model has access to public data which may not be the case in practice
- DP-GAN*[Xie et al., 2018]*
  - Results do not look promising even on MNIST
  - No methodology to create labels with synthetic images
- PATE-GAN*[Yoon et al., 2018]*
  - Assigns just binary labels to synthetic images

TensorFlow Privacy



tensorflow / privacy

Watch 45 | Star 766 | Fork 109

Code | Issues 4 | Pull requests 0 | Security | Insights

Library for training machine learning models with privacy for training data

machine-learning | privacy

130 commits | 1 branch | 0 releases | 14 contributors | Apache-2.0

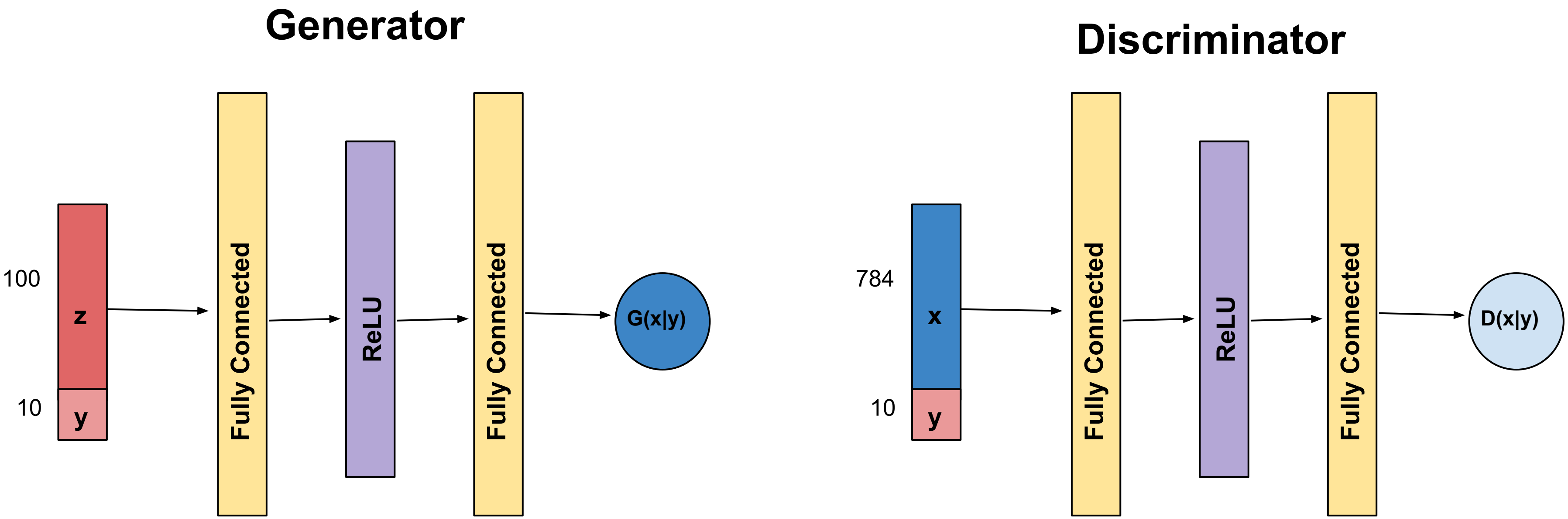
Branch: master | New pull request | Find File | Clone or download

tensorflower-gardener Logistic regression for mnist with new privacy analysis. ... Latest commit 2b97c7c 3 days ago

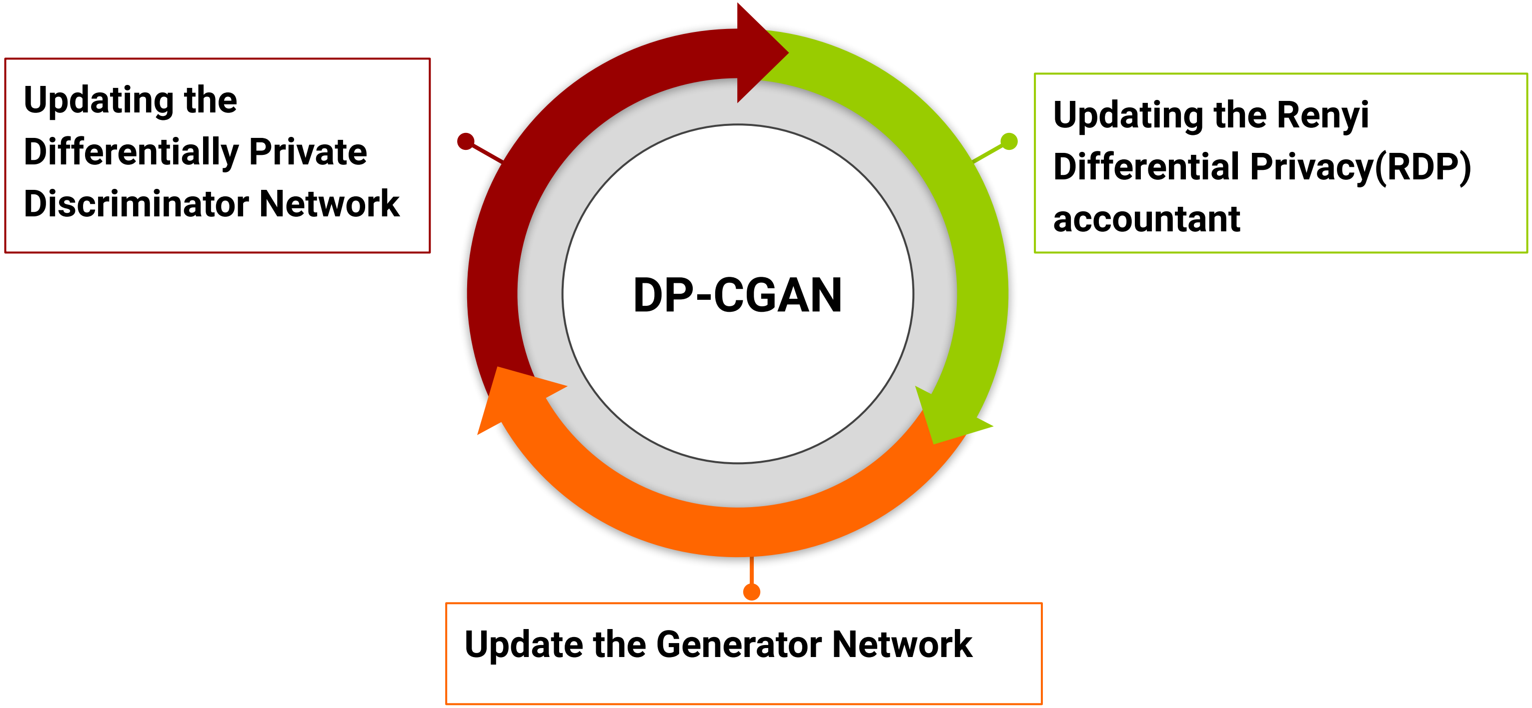
privacy	Remove set_denominator functions from DPQuery and make QueryWithLedge...	10 days ago
research	Closes #32	3 months ago
tutorials	Logistic regression for mnist with new privacy analysis.	3 days ago
CONTRIBUTING.md	Project import generated by Copybara.	6 months ago
LICENSE	Correct license author string.	4 months ago
README.md	Specifying minimal TF version required (currently 1.13, due to depend...	28 days ago
requirements.txt	Specifying minimal TF version required (currently 1.13, due to depend...	28 days ago
setup.py	- Fixing dependencies in setup.py and requirements.txt.	5 months ago

https://github.com/tensorflow/privacy/

CGAN Architecture

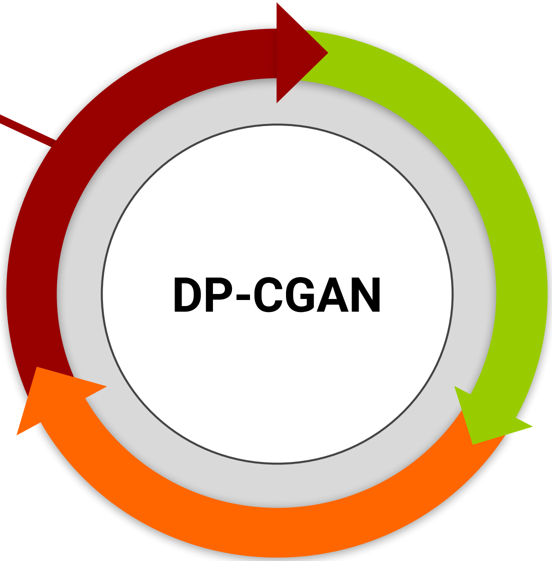


DP-CGAN


Updating the Differentially Private Discriminator Network
Updating the Renyi Differential Privacy(RDP) accountant
DP-CGAN
Update the Generator Network

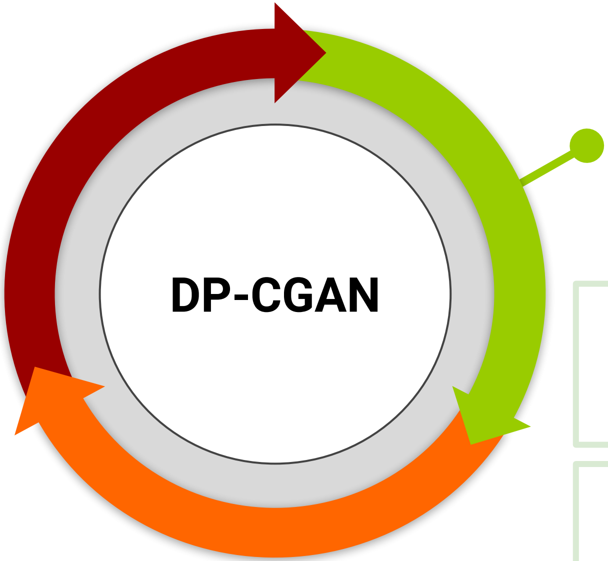
DP-CGAN

Updating the Differentially Private Discriminator Network



- Compute **per-example gradients** of discriminator loss on **real data** and clip them
- Compute **per-example gradients** of discriminator loss on **fake data** and clip them
- Compute the **overall gradients** of discriminator and add **Gaussian Noise** to them
- Take the **gradient Descent** step for discriminator

DP-CGAN

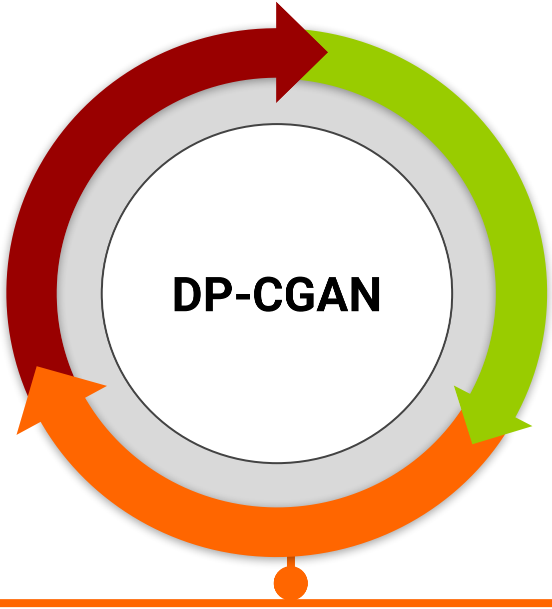


Updating the Renyi Differential Privacy(RDP) accountant

- Accumulate the spent privacy budget using **RDP accountant**

- Set the **termination flag ON** if the spent privacy budget exceeds the target privacy budget

DP-CGAN

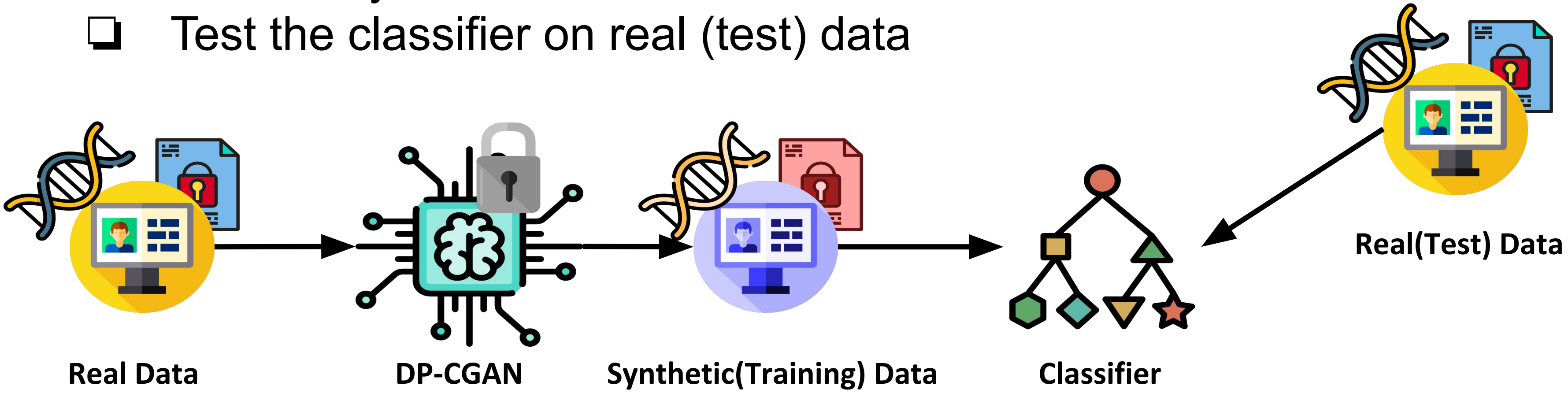


- Compute the **gradients** of Generator loss
- Take the **gradient Descent** step for Generator

Update the Generator Network

Evaluation Methodology

- Evaluation process:
  - Generate synthetic data
  - Use the synthetic data to train the classifiers
  - Test the classifier on real (test) data



Evaluation Methodology

- Three methods for generating synthetic training data:
  - CGAN with no privacy
  - CGAN with Basic DP
    - Applies clipping and adds noise to gradients of discriminator loss
  - DP-CGAN
    - Splits the gradients of discriminator to gradients on real data and gradients on fake data, then applies clipping and adds noise to them separately
- In all the experiments:
  - **Delta** is set to **10^{-5}**
  - MNIST dataset is used (60,000 training samples and 10,000 test samples)

Experimental Results

- Evaluation Metric: AuROC
- ε=9.6 for DP approaches

	Real	CGAN	DP-CGAN	CGAN with Basic DP
Logistic Regression	92.17%	91.10%	**87.57%**	83.42%
Multi-Layer Perceptron	97.60%	91.06%	**88.16%**	83.29%

Future Directions

- ❑ Improving DP-CGANs accuracy using techniques such as warm-start, random sampling, using a public dataset, etc.
- ❑ Evaluating the DP-CGANs on larger scale datasets
- ❑ Considering advanced CGAN architectures

Thank you!

Questions?