Topik : 4.1. Konsep Differential Privacy

Objective : Pahami noise, ε (epsilon), dan trade-off akurasi

Task : Buat ringkasan + visual ilustrasi noise

Source : https://github.com/tensorflow/privacy?tab=readme-ov-file

Tensorflow Privacy Adalah sebuah library python yang menyediakan implementasi optimizer dalam Tensorflow yang mendukung differential privacy. Dengan library ini , pengguna dapat melatih model machine learning sambil menjaga privasi data pelatihan. Selain optimizers, tersedia juga tutorial dan alat analisis untuk menghitung jaminan privasi yang diberikan.

Pembaruan Terbaru

- 14 Febuary 2024 ---- Rilis versi 0.9.0. setelah versi ini, repo dibagikan menjadi dua paket PyPI :
  - tensorflow-privacy : berisi bagian yang terkait dengan pelatihan model DP.
  - Tensorflow-empirical-privacy : berisi bagian yang terkait dengan pengujian privasi empiris
- 21 Febuary 2023 --- Tambahan implementasi per-example gradient clipping yang efisien untuk model DP berbasis keras (Khusus Dense & Embedding layers). Implementasi ini memungkinkan pelatihan DP tanpa overhead memori atau kinerja yang signifikan, sekaligus menghilangkan kebutuhan tuning jumlah *microbatches.*

Langkah Menginstall Tensorflow Privacy

1. Jika anda hanya ingin menggunakan Tensorflow Privacy sebagai sebuah library, kita cukup menjalankan perintah :

   ```
   pip install tensorflow-privacy
   ```

2. Sebaliknua, kita bisa meng-clone repository github ke dalam direktori pilihan :

   ```
   git clone https://github.com/tensorflow/privacy
   ```

   Kemudian , kita bisa menginstall paket local dalam mode *editable* agar dapat ditambahkan ke PYTHONPATH :

   ```
   cd privacy
   pip install -e .
   ```

Setup & Persyaratan

- Membutuhkan Tensorflowb (>= 1.14), meskipun pengguna sangat disarankan menggunakan versi dengan dukungan GPU untuk kinerja lebih baik.

Source : https://www.youtube.com/watch?v=FJMjNOcIqkc&list=PLmd_zeMNzSvRRNpoEWkVo6QY_6rR3SHjp

Kasus – Kasus Kegagalan Privasi

Contoh konkret dari dunia nyata tentang bagaimana data yang "dianonimkan" tetap rentan terhadap serangan re-identifikasi.

- **Data taksi New York City (NYC Taxi)**: Medali taksi yang di-*hash* dengan metode yang lemah yang mudah dibalik mengakibatkan pengungkapan rute dan pendapatan supir. Contoh klasik dari kegagalan teknik de-identifikasi sederhana
- **Netflix Prize Dataset**: Dataset ini, yang memuat rating film dari ribuan pengguna, semula dianggap anonim. Namun, serangan de-anonimisasi berhasil mengidentifikasi pengguna konkret hanya dari sejumlah kecil informasi latar belakang yang diketahui—mengungkap preferensi film dan kemungkinan informasi pribadi lainnya.

Studi Kasus: NYC Taxi Data

- Pemerintah merilis dataset perjalanan taksi New York City dengan informasi:
    - Nomor medali (identifikasi taksi).
    - Waktu & lokasi penjemputan / pengantaran.
    - Biaya perjalanan.
- Kesalahan: nomor medali hanya di-*hash* dengan metode lemah (MD5 tanpa salt).
- Peneliti bisa membalik hash → menemukan identitas supir → menghitung pendapatan pribadi mereka.
- Pelajaran: *hashing ≠ anonimisasi*. Data masih bisa dikaitkan dengan individu.

Studi Kasus: Netflix Prize Dataset

- Netflix merilis dataset rating film dari ratusan ribu pengguna untuk kompetisi peningkatan algoritma rekomendasi.
- Data yang dibagikan:
    - ID pengguna anonim.
    - Film yang dinilai + tanggal.
    - Skor rating.
- Serangan de-anonimisasi:

    - Peneliti membandingkan data Netflix dengan informasi publik di IMDb (misalnya, rating film oleh pengguna tertentu).
    - Hanya perlu beberapa rating untuk mengidentifikasi orang sebenarnya.
    - Hasil: preferensi pribadi (bahkan sensitif) dari pengguna dapat terekspos.
- Dampak: kasus hukum (Gonzalez vs. Netflix), dataset akhirnya ditarik.

Pentingnya Privasi Formal yang Kuat

Kedua kasus tersebut menunjukkan bahwa penghapusan identitas eksplisit saja tidak cukup dalam melindungi data individu. Hal ini menjadi motivasi utama untuk menggunakan pendekatan formal seperti **differential privacy (DP)**, yang memberikan jaminan matematis bahwa informasi individu tidak dapat diidentifikasi melalui keluaran sistem.

Apa itu Differential Privacy (DP)

Differential Privacy (DP) adalah sebuah kerangka matematis untuk melindungi privasi individu di dalam dataset.

Inti :

- Hasil analisis (output) hampir sama baik ketika data seorang individu ada maupun tidak ada di dalam dataset.
- Artinya, partisipasi satu orang tidak akan mengubah hasil secara signifikan, sehingga sulit (atau hampir mustahil) bagi penyerang untuk mengetahui apakah data orang itu ada di dataset atau tidak.

Source :

https://openlibrary.telkomuniversity.ac.id/pustaka/files/172304/jurnal_eproc/analisis-performansi-exponential-mechanism-pada-dataset-student-s-alcohol-consumptions-dalam-memenuhi-differential-privacy.pdf

Differential Privacy adalah pendekatan baru yang memungkinkan seseorang untuk mengumpulkan, mengolah, ataupun menganalisis data sekaligus menjamin perlindungan privasi terhadap informasi individu yang ada pada dataset . Sehingga ada atau tidaknya suatu informasi individu pada dataset, tidak akan mempengaruhi hasil analisis data. Differential Privacy belakangan ini masih menjadi topik penelitian yang popular dalam bidang privasi data serta penggunaannya telah berkembang pesat. Apple dan Google adalah dua perusahaan besar yang telah mengaplikasikan metode differential privacy pada sistem keamanannya. Differential

Privacy dianggap dapat menjamin perlindungan informasi sensitif dari data suatu individu, terlepas dari latar belakang informasi yang diketahui oleh pihak ketiga.

Differential Privacy adalah metode pertama yang dapat membuktikan privacy guarantee secara matematis dan dapat mengukur potensi dari privacy loss (s). Privacy loss merupakan parameter yang membatasi berapa banyak hilangnya privasi pada suatu data. Parameter s merepresentasikan tradeoff antara privasi dan akurasi data, dimana parameter s menentukan berapa banyak noise yang akan ditambahkan ke dataset. Jika value s lebih kecil, maka jaminan privasi lebih besar namun akurasi dari hasil ouput makin kecil. Jika value s lebih besar, maka akurasi dari hasil output makin besar namun jaminan privasi semakin kecil. Sehingga menentukan s yang tepat merupakan pilihan yang sangat penting ketika mengimplementasi differential privacy.

Penerapan Differential Privacy menggunakan Laplace Mechanism pada sampel dataset terpilih. Berdasarkan penelitian tersebut disebutkan bahwa penerapan Laplace Mechanism hanya terbatas pada tipe data numerik saja, serta diperlukannya suatu pendekatan yang dapat mempermudah dalam menentukan nilai s(epsilon) yang optimal. Disebutkan juga bahwa penggunaan nilai s(epsilon) yang sama dapat menghasilkan privacy guarantee yang berbeda tergantung dari jenis kueri data yang dilakukan dan atribut data yang digunakan.

Cara Kerja Differential Privacy

Differential privacy bekerja dengan menambahkan random noise kedalam dataset pada saat melakukan analisis data, yang mana hal ini menutup kemungkinan untuk mempelajari informasi individu yang ada pada dataset berdasarkan hasil analisis. Namun perlu digaris bawahi bahwa hasil analisis data setelah penambahan noise bukanlah 100% akurat, melainkan perkiraan yang mendekati akurasi asli dan nilai yang sebenarnya. Hal ini memungkinkan hasil analisis data akan terus berubah selama beberapa kali karena noise yang digunakan adalah random . Dengan demikian, differential privacy memberikan jaminan privasi (privacy guarantee) resiko yang sama pada tiap individu terlepas apakah informasi mereka ada pada analisis ataupun tidak

Gambar 1. Alur Differential Privacy

Differential privacy bergantung pada probabilitas. Maka, noise yang ditambahkan kedalam dataset haruslah acak. Random noise yang dibutuhkan ini harus diperoleh dari distribusi probabilitas . Probabilitas yang digunakan juga harus bergantung terhadap jenis mekanisme dan tipe data yang akan digunakan. Sebagai contoh jika tipe data yang akan digunakan adalah data dengan tipe data numerik, maka salah satu mekanisme yang cocok untuk

digunakan adalah Laplace Mechanism. Dimana Laplace Mechanism menambahkan sejumlah noise (ke perhitungan data asli) yang diperoleh dari fungsi probabilitas distribusi laplace. Jumlah noise yang ditambahkan akan berujung kepada kondisi dimana user harus bisa menyeimbangkan privacy dan utilitas data (kondisi ini disebut sebagai privacy tradeoff). Menambahkan terlalu banyak noise akan membuat data anonim, namun utilitas data akan berkurang. Dalam differential privacy, privacy tradeoff ini dikontrol oleh parameter privacy loss yang dilambangkan dengan ꜱ (epsilon). Apabila privacy tradeoff ini dapat dikontrol dengan baik, maka data akan memenuhi ꜱ-differential privacy

Task :

Differential Privacy (DP)	kerangka matematis untuk melindungi privasi individu dalam dataset.
Noise	Kebisingan acak yang ditambahkan ke data atau hasil query agar kontribusi individu tersamarkan (menyamarkan kontribusi individu)
ε (epsilon)	Parameter privasi. • ε kecil → lebih banyak noise → privasi lebih kuat, akurasi menurun. • ε besar → lebih sedikit noise → akurasi tinggi, privasi lebih lemah.
Trade-off	Semakin tinggi privasi (banyak noise), semakin rendah akurasi hasil analisis. Sebaliknya, jika ingin akurat, maka jaminan privasi berkurang. Artinya keseimbangan antara *privasi* dan *utilitas data*.

[16] ✓ 0d
```
import numpy as np
import matplotlib.pyplot as plt
from IPython.display import Image, display
```

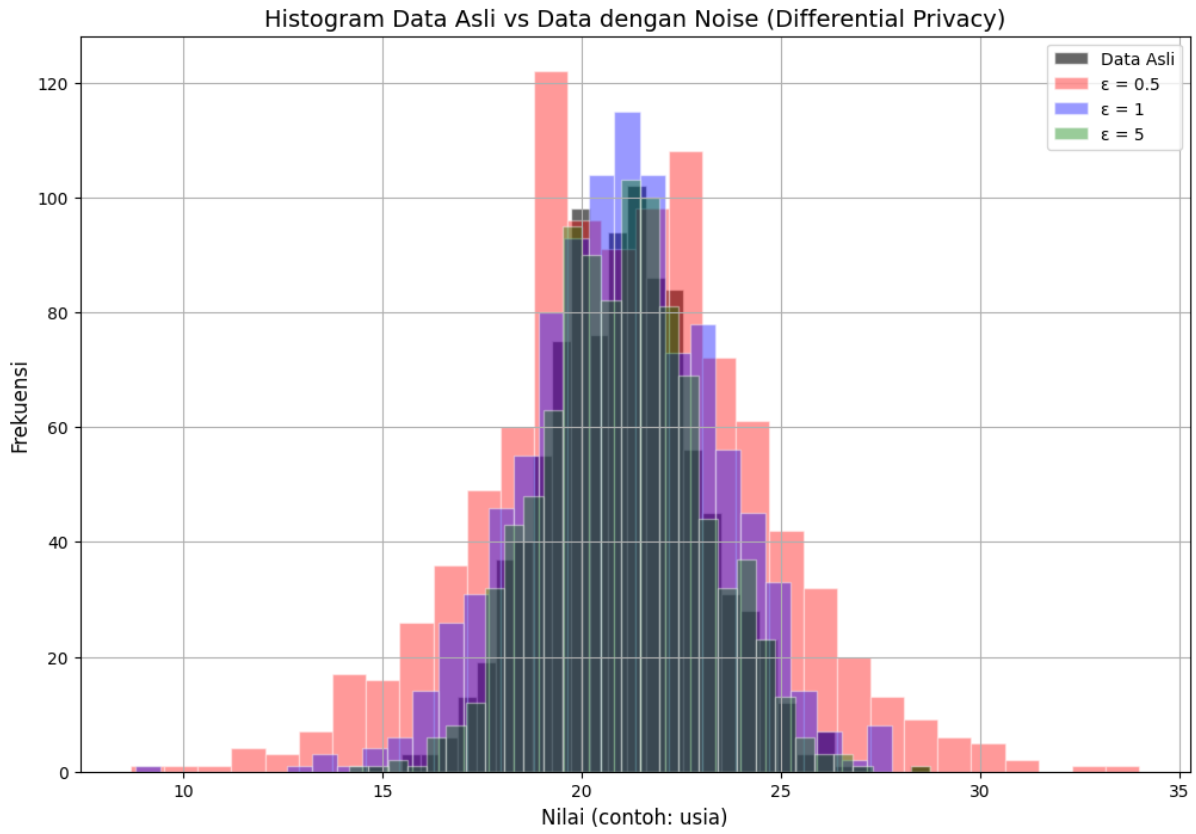
[17] ✓ 0d
```
np.random.seed(42)
data = np.random.normal(21, 2, 1000)  # data asli

epsilons = [0.5, 1, 5]
noisy_data = {eps: data + np.random.laplace(0, 1/eps, len(data)) for eps in epsilons}
```

[18] ✓ 0d
```
plt.figure(figsize=(12,8))
plt.hist(data, bins=30, alpha=0.6, label="Data Asli", color="black", edgecolor="white")

colors = ["red", "blue", "green"]
for eps, color in zip(epsilons, colors):
    plt.hist(noisy_data[eps], bins=30, alpha=0.4, label=f"ε = {eps}", color=color, edgecolor="white")

plt.title("Histogram Data Asli vs Data dengan Noise (Differential Privacy)", fontsize=14)
plt.xlabel("Nilai (contoh: usia)", fontsize=12)
plt.ylabel("Frekuensi", fontsize=12)
plt.legend()
plt.grid(True)
```



Interpretasi Histogram Differential Privacy

1. **Hitam = Data Asli (tanpa noise)**
    - Ini distribusi asli dari data (misalnya usia mahasiswa).
    - Berbentuk lonceng normal dengan rata-rata sekitar 21 tahun.
    - Inilah bentuk data yang *paling akurat*, tapi paling berisiko karena identitas individu bisa lebih mudah ditebak jika data dipublikasikan apa adanya.

2. **Merah = ε (epsilon) = 0.5 → Noise Besar**
    - Karena ε kecil, mekanisme Differential Privacy menambahkan noise yang banyak.
    - Akibatnya distribusi jadi melebar dan lebih “acak”.
    - Privasi kuat → sangat sulit menebak data asli individu.
    - Tapi akurasi rendah → hasil analisis jadi kurang mewakili data sebenarnya (informasi global agak kabur).

3. **Biru = ε = 1 → Noise Sedang**
    - Noise yang ditambahkan tidak sebesar ε=0.5, tapi masih ada penyebaran.
    - Distribusinya masih cukup mirip dengan data asli, hanya sedikit lebih melebar.
    - Privasi dan akurasi seimbang → ini biasanya kondisi yang paling masuk akal dipilih dalam praktik.

4. **Hijau = ε = 5 → Noise Kecil**
    - Karena ε besar, noise yang ditambahkan sangat kecil.
    - Distribusi hampir menyatu dengan data asli (hitam).
    - Akurasi tinggi → hasil analisis sangat dekat dengan data sebenarnya.
    - Tapi privasi lemah → individu dalam dataset lebih berisiko terekspos.

Kesimpulan :

- Untuk ε = 0.5 (merah): noise yang ditambahkan paling besar, makanya distribusinya jadi lebih menyebar/lebar.
- Untuk ε = 1 (biru): noise sedang, distribusi masih mirip dengan data asli tapi ada sedikit penyebaran tambahan.
- Untuk ε = 5 (hijau): noise sangat kecil, distribusi hampir sama dengan data asli, jadi hampir tidak terlihat perbedaan.

Dengan kata lain, noise = selisih yang ditambahkan ke data asli.

```
Data asli  : [21.99, 20.72, 22.29, 24.04, 20.53]
Data noisy : [21.16, 18.11, 20.76, 23.65, 21.46]
```

Kalau dihitung:

```
Noise      : [-0.83, -2.61, -1.53, -0.39, +0.93]
```

angka-angka acak yang membuat nilai data berubah → itulah **noise**.