Implementasi Model Pembelajaran Mesin untuk Data Terenkripsi

Juniardi Akbar 13517075
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
juniardiakbar@gmail.com, 13517075@std.stei.itb.ac.id

***Abstract*—Pembelajaran mesin merupakan salah satu cabang dari bidang kecerdasan buatan yang sedang berkembang pesat saat ini. Inti utama dari teknologi ini adalah penggunaan data untuk "mengajari" komputer guna menyelesaikan suatu tugas dengan suatu metrik yang telah ditetapkan. Pada penggunaan di beberapa industri, seringkali dibutuhkan pelatihan model pembelajaran mesin dengan data yang bersifat pribadi bahkan rahasia. Dalam konteks ini, salah satu solusi yang mungkin diterapkan adalah dengan mengenkripsi model dan data. Proses komputasi dengan data yang terenkripsi dan yang tidak terenkripsi mungkin akan memiliki perbedaan pada beberapa aspek seperti akurasi sampai waktu pelatihan. Untuk melakukan pelatihan pada data terenkripsi, terdapat beberapa skema enkripsi yang dapat digunakan untuk komputasi salah satu yang akan digunakan adalah *Secure Multi-Party Computation* (SMPC).**

***Keywords*—kriptografi, enkripsi, machine learning, neural network, Secure Multi-Party Computation**

I. Pendahuluan

Pembelajaran mesin merupakan salah satu cabang dari bidang kecerdasan buatan yang sedang berkembang pesat saat ini. Dengan meningkatnya animo pada teknologi ini, wajarlah bila pembelajaran mesin kemudian menjadi salah satu topik yang sering diangkat dalam riset. Selain itu, dengan pesatnya penggunaan pembelajaran mesin menyebabkan semakin luasnya penggunaan teknologi ini baik untuk penggunaan pribadi sampai ke korporasi besar.

Seperti yang kita ketahui, inti utama dari kesuksesan teknologi ini adalah data. Bahkan beberapa orang menganggap data sebagai sumber daya yang sangat penting saat ini dan mengungkapkan istilah "*data is the new oil"*. Pada dasarnya, algoritma ini memang akan menggunakan data untuk belajar guna menyelesaikan suatu tugas dengan suatu metrik yang telah ditetapkan.

Pada penggunaan di beberapa industri, seringkali dibutuhkan pelatihan model pembelajaran mesin dengan data yang bersifat pribadi bahkan rahasia. Contohnya adalah data-data pribadi pengguna atau data perusahaan yang tentu saja bersifat rahasia (*confidential*). Pada kasus ini, enkripsi data perlu dilakukan untuk memenuhi aspek *confidentiality* (kerahasian) sehingga informasi tidak dapat diketahui oleh pihak yang tidak bertanggung jawab.

Dalam konteks ini, salah satu solusi yang mungkin dapat diterapkan adalah dengan mengenkripsi model dan data, lalu melatih model pembelajaran mesin melalui data yang terenkripsi. Dengan cara ini, perusahaan tidak akan dapat mengakses data rahasia. Pengguna juga tidak dapat menggunakan model dari data rahasia mereka tanpa izin dari perusahaan.

Beberapa skema enkripsi yang memungkinkan untuk dilakukan komputasi atas data terenkripsi, termasuk pelatihan model pembelajaran mesain, di antaranya adalah *Secure Multi-Party Computation* (SMPC), *Homomorphic Encryption* (FHE) dan *Functional Encryption* (FE).

II. Dasar Teori

A. Kriptografi

Kriptografi adalah studi tentang penerapan teknik matematika dalam yang berhubungan terhadap aspek keamanan informasi seperti *confidentiality* (kerahasiaan), *data integrity* (kebenaran), *authentication* (identifikasi), dan *non-repudiation* (anti penyangkalan). Tujuan mendasar dari kriptografi adalah menerapkan keempat bidang ini secara memadai baik dalam teori maupun praktek. Seiring perkembangan, fungsi dari algoritma kriptografi kian meluas sehingga tidak hanya berfungsi sebagai metode untuk menyembunyikan pesan, tetapi juga upaya pencegahan dan deteksi kecurangan atau aktivitas jahat lainnya.

Sebelum kita masuk ke pembahasan yang lebih dalam, terdapat beberapa istilah atau terminologi yang penting tentang kriptografi untuk diketahui terlebih dahulu.

1. Pesan, Plainteks, dan Cipherteks

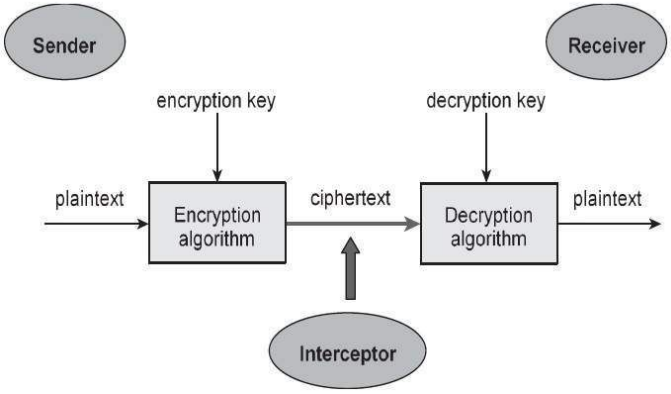
Pesan adalah informasi yang dapat dibaca dan dipahami artinya oleh penerimanya. Pesan dapat dikirim dan disimpan. Pesan yang terkirim adalah informasi yang disampaikan kepada penerima. Sedangkan pesan yang tersimpan adalah pesan yang diamankan dalam suatu media yang dapat berupa teks, citra, suara, dan video.

Plainteks adalah pesan dalam bentuk teks yang memiliki informasi yang masih dapat dibaca dan dipahami. Agar pesan tidak dapat dipahami oleh pihak lain, maka pesan harus disandikan sehingga pesan akan sulit untuk dimengerti. Pesan

yang telah disandikan sehingga memiliki informasi yang tidak dapat dipahami ini disebut dengan cipherteks.

2. Enkripsi dan Dekripsi

Enkripsi adalah proses menyandikan data dari plainteks ke chiperteks. Adapun sebaliknya, dekripsi adalah proses menerjemahkan data dari cipherteks ke plainteks. Dalam konteks ini, enkripsi bertujuan untuk menyembunyikan informasi dari pihak yang tidak berwenang. Sehingga hasil enkripsi hanya bisa diterjemahkan oleh penerima pesan saja.



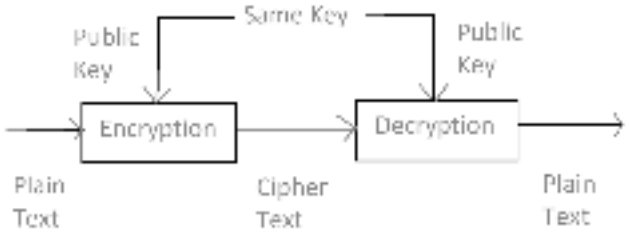
3. Cipher dan Kunci

Cipher adalah algoritma kriptografi yang digunakan untuk mengenkripsi dan mendekripsi suatu data. Pada beberapa algoritma kriptografi modern, dibutuhkan kunci untuk melakukan proses enkripsi dan dekripsi. Kunci adalah parameter yang digunakan pada fungsi enkripsi dan dekripsi.

Terdapat beberapa metode kriptografi yang berhubungan dengan implementasi proses enkripsi pesan yaitu enkripsi kunci simetri dan enkripsi kunci publik.

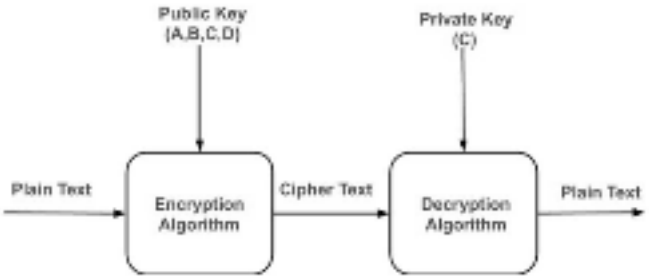
Algoritma kriptografi memerlukan kunci untuk melakukan proses enkripsi dan dekripsi. Pada algoritma enkripsi kunci-simetri, kunci yang digunakan untuk proses enkripsi sama dengan kunci yang akan digunakan untuk proses dekripsi. Sistem kriptografi kunci-simetri mengasumsikan pengirim dan penerima pesan telah memiliki kunci yang sama.

Pada kriptografi kunci-simetri, keamanannya hanya terletak pada kerahasiaan kuncinya, sedangkan algoritmanya sendiri tidak perlu rahasia. Kelemahan dari kriptografi kunci-simetri adalah penerima pesan harus memiliki kunci yang sama dengan pengirim pesan sehingga pengirim pesan harus mencari suatu cara untuk memberitahu kunci tersebut kepada penerima pesan.



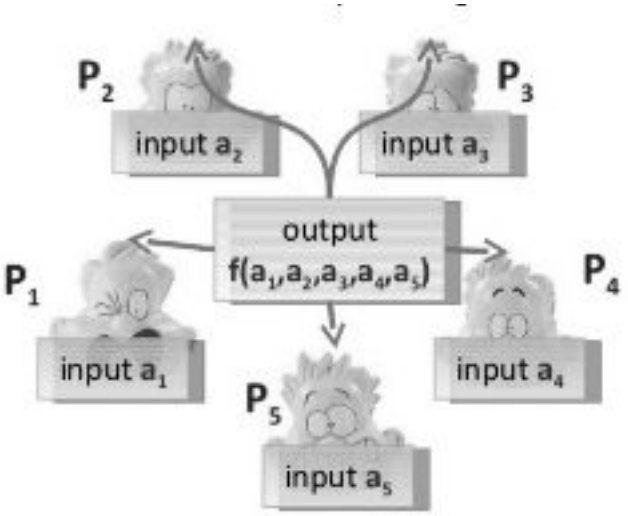
Pada kriptografi kunci publik, kunci yang digunakan pada proses enkripsi tidak rahasia sedangkan kunci yang digunakan pada proses dekripsi harus dirahasiakan. Oleh karena itu, pada algoritma ini pengirim dan penerima pesan akan memiliki sepasang kunci yaitu kunci privat dan kunci publik.

Pada kriptografi kunci publik, tidak terdapat kebutuhan untuk mendistribusikan kunci rahasia (kunci privat). Kemudian dengan adanya dua buah kunci, maka jumlah kunci dapat ditekan. Untuk berkomunikasi secara rahasia dengan banyak orang tidak perlu kunci privat sebanyak jumlah orang, cukup mendeklarasikan dua buah kunci yaitu kunci publik untuk mengenkripsi pesan dan kunci privat untuk mendekripsi pesan yang diterima.



B. Secure Multi-Party Computation

Secure Multi-Party Computation (SMPC) adalah salah satu bagian dari ilmu kriptografi yang memiliki tujuan untuk menggabungkan masukan dari setiap anggota dalam suatu grup dengan mengedepankan prinsip *confidelity* (kerahasiaan). Artinya, setiap masukan yang diberikan bersifat rahasia dan hanya diketahui oleh pemilik masukan tersebut. Ilustrasi dari SMPC akan diperlihatkan pada gambar berikut.



Pada kasus diatas, terdapat lima partisipan P1,…,P5 yang masing-masing memiliki masukan $a_1,..,a_5$. Kelima anggota menyepakati fungsi f yang akan menerima seluruh masukan untuk mendapatkan sebuah informasi yang disebut $y = f(a_1, a_2, a_3, a_4, a_5)$. Pada SMPC, terdapat dua buah kondisi yang perlu untuk dipenuhi yaitu *correctness* dan *privacy*. Kondisi *correctness* akan memastikan bahwa informasi nilai y merupakan hasil dari komputasi yang benar, sedangkan kondisi *privacy* akan menjamin bahwa hanya informasi baru

yang dipublikasikan. Setiap masukan yang diberikan akan tetap bersifat *private*, sehingga hanya pemilik masukan tersebut yang mengetahuinya

C. Pembelajaran Mesin

Pembelajaran mesin atau yang lebih dikenal dengan istilah *machine learning* adalah salah satu bidang keilmuan dalam *computer science* yang sedang naik daun. Pembelajaran mesin sendiri merupakan cabang ilmu dari teknologi kecerdasan buatan atau *artificial intelligence*. Secara umum, pembelajaran mesin merupakan suatu bidang keilmuan yang memberikan kemampuan kepada komputer untuk belajar tanpa harus diprogram secara eksplisit.

Secara lebih spesifik, suatu algoritma pembelajaran mesin akan memberikan komputer suatu pengalaman *E* terhadap suatu tugas tertentu *T,* dengan penilaian kinerja *P*. Suatu komputer dikatakan belajar apabila kinerjanya dalam mengerjakan *T* yang diukur dengan menggunakan *P* meningkat dengan pengalaman *E*.

Secara umum, terdapat 3 buah teknik dalam pembelajaran mesin yang sudah digunakan secara luas yaitu *supervised learning, unsupervised learning,* dan *reinforcement learning*. *Supervised learning* menggunakan data yang memiliki label yang merupakan jawaban atau tujuan dari setiap input masukan. Sebaliknya, *unsupervised learning* akan menggunakan data yang tidak menggunakan label.

Berbeda dengan dua teknik sebelumnya, *reinforcement learning* akan mengamati lingkungan sembari mencoba untuk mengambil tindakan atau keputusan berdasarkan pengamatannya. Untuk setiap keputusan, akan diberikan hadiah atau hukuman. Dari hadiah atau hukuman ini, sistem akan mengenali dan menyusun strategi yang paling optimal untuk mendapatkan sebanyak mungkin hadiah dan menghindari sebanyak mungkin hukuman.

Teknik *supervised learning* umumnya digunakan untuk persoalan klasifikasi dan prediksi. Klasifikasi merupakan sebuah persoalan untuk menentukan kelas (bersifat diskrit) dari sampel data yang diberikan. Contoh persoalan klasifikasi adalah menentukan digit dari suatu citra tulisan tangan. Sedangkan prediksi merupakan tugas mencari suatu nilai (bersifat kontinu) dari sampel data yang diberikan. Contohnya adalah persoalan menentukan harga suatu rumah dari informasi lokasi, luas, dan atribut lainnya.

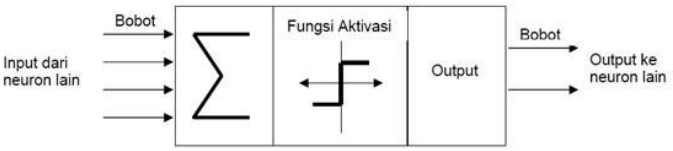
Adapun teknik *unsupervised learning* umumnya digunakan untuk persoalan *clustering* yaitu persoalan menentukan kelompok-kelompok dari sampel data yang diberikan. Contoh persoalan *clustering* adalah mengelompokkan berita dari suatu kumpulan berita. Selain persoalan *clustering,* unsupervised learning juga dapat digunakan untuk tahapan penyederhanaan data dengan cara mengurangi terlalu informasi atau yang biasa disebut dengan istilah *dimensionality reduction*.

Teknik terakhir, *reinforcement learning* biasanya digunakan untuk persoalan seperti *real-time decision*, navigasi pada robot, kecerdasan buatan pada *game*, dan persoalan *learning*. Sifatnya yang dapat mengamati lingkungan, sangat cocok digunakan kepada agen yang bergerak sehingga menciptakan gerak (keputusan) yang optimal.

D. Artificial Neural Network

Neural network adalah jaringan dari sekelompok unit pemroses kecil yang dimodelkan berdasarkan sistem saraf manusia. Pada jaringan saraf manusia, neuron menerima impuls dari neuron lain melalui dendrit dan mengirimkan sinyal yang dihasilkan oleh badan sel melalui akson. Akson dari sel saraf ini bercabang-cabang dan berhubungan dengan dendrit dari sel saraf lain dengan cara mengirimkan impuls melalui sinapsis.

Dari struktur neuron pada otak manusia, dan proses kerja yang dijelaskan di atas, terciptalah suatu konsep pembangunan jaringan saraf tiruan atau yang lebih dikenal dengan istilah artificial neural network (ANN). Ide dasar dari ANN adalah mengadopsi mekanisme berpikir sebuah sistem atau aplikasi yang menyerupai otak manusia, baik untuk pemrosesan elemen yang diterima, toleransi terhadap kesalahan/*error*, dan juga *parallel processing*.



Gambar di atas menjelaskan struktur ANN secara mendasar, yang secara mudahnya dapat dijelaskan sebagai berikut.

- *Input*, berfungsi seperti dendrite
- *Output*, berfungsi seperti akson
- Fungsi aktivasi, berfungsi seperti sinapsis

Proses pada ANN akan dimulai dengan diterimanya masukan oleh neuron beserta nilai bobot dari tiap-tiap input yang telah diinisialisasi. Setelah masuk ke dalam neuron, nilai input yang ada akan dijumlahkan oleh suatu fungsi yang bisa dilihat seperti pada di gambar di atas dengan lambang sigma (∑). Hasil dari fungsi ini akan diproses oleh fungsi aktivasi pada setiap neuron. Pada fungsi aktivasi, hasil penjumlahan akan dibandingkan dengan suatu nilai ambang batas (*threshold*). Jika nilai melebihi *threshold*, maka aktivasi neuron akan dibatalkan, sebaliknya, jika masih dibawah nilai *threshold*, neuron akan diaktifkan. Setelah aktif, neuron akan mengirimkan nilai output melalui bobot-bobot outputnya ke semua neuron yang berhubungan dengannya. Proses ini akan terus berulang pada masukan selanjutnya.

ANN terdiri dari banyak neuron di dalamnya. Neuron-neuron ini akan dikelompokkan ke dalam beberapa lapisan (*layer*). Neuron yang terdapat pada tiap *layer* dihubungkan dengan neuron pada *layer* berikutnya. Informasi yang diterima di *layer* input dilanjutkan ke *layer-layer* dalam ANN secara satu persatu hingga mencapai *layer* terakhir yaitu output. *Layer* yang terletak di antara *input* dan *output* disebut sebagai *hidden layer*.

III. Analisis Model

Pada bagian ini, penulis akan menjabarkan bagaimana implementasi model yang akan dilakukan termasuk menyoroti elemen kunci dalam proses pelatihan model pembelajaran dengan data terenkripsi yaitu *Secure Multi-Party Computation*

(SMPC). Untuk implementasi model pembelajaran mesin, penulis menggunakan PyTorch yang merupakan pustaka pembelajaran mesin *open source* yang dikembangkan oleh lab AI Research Facebook. Pustaka PyTorch umumnya banyak digunakan untuk aplikasi seperti *computer vision* dan pemrosesan bahasa alami Sedangkan untuk implementasi SMPC akan digunakan pustaka PySyft yang dikembangkan oleh *openmined*. Pustaka PySyft akan menghubungkan dua *virtual worker* (alice dan bob) dan meminta pekerja lain bernama *crypto_provider* untuk mengerjakan semua tugas primitif kriptografi yang diperlukan.

Tujuan yang ingin diperoleh dari pemanfaatan SMPC dalam pelatihan pembelajaran mesin adalah mengamankan model dan data. Dengan SMPC, setiap data yang berikan oleh setiap anggota dalam suatu grup akan dienkripsi dan diproses oleh fungsi yang dalam hal ini adalah pelatihan model *neural network*. Lebih konkretnya, dengan melakukan pelatihan pada data terenkripsi, kita dapat mencegah siapa pun melihat tidak hanya data individu, tetapi juga parameter model yang dilatih.

Untuk pengujian, penulis akan menggunakan persoalan MNIST yaitu persoalan klasifikasi tulisan tangan angka yang umum digunakan untuk melatih berbagai pengolahan citra sistem. Data akan dilatih dengan menggunakan model *neural network* dari PyTorch dan data latih akan dienkripsi terlebih dahulu dengan menggunakan PySyft.

Dalam menggunakan PySyft, kita harus menginisialisasi terlebih dahulu *virtual worker*. Untuk menginisialisasinya, kita perlu mensimulasikan mesin (perangkat) jarak jauh dan kita dapat melakukannya dengan membuat *virtual worker*. Pada model yang penulis coba, digunakan dua buah *worker* alice dan bob selaku *worker0* dan *worker1*.

```
def connect_to_workers(n_workers):
  return [
    sy.VirtualWorker(hook, id=f"worker{i+1}")
    for i in range(n_workers)
  ]

workers = connect_to_workers(n_workers=2)
```

```
private_train_loader, private_test_loader =
get_private_data_loaders(
   precision_fractional=args.precision_fractional,
   workers=workers,
   crypto_provider=crypto_provider,
   dataset_sizes=(n_train_items, n_test_items)
)
```

Pada kode di atas, *virtual worker* alice dan bob akan dibentuk. Kemudian, kode selanjutnya akan mengirim nilai data training ke bob lalu referensi (*pointer*) dari data training dikirim ke alice.

```
workers[0]._objects
```

```
{51573677535: (Wrapper)>[AdditiveSharingTensor]
        -> [PointerTensor | me:6841995512 ->
worker1:25075963984]
        -> [PointerTensor | me:46654356980 ->
worker2:98661899731]
        *crypto provider: crypto_provider*,
 70959159883: (Wrapper)>[AdditiveSharingTensor]
        -> [PointerTensor | me:49588438391 ->
worker1:68692812797]
        -> [PointerTensor | me:40513725667 ->
worker2:72537611820]
        *crypto provider: crypto_provider*,
...
}
```

Dapat dilihat pada data di atas, data yang terdapat pada *worker* alice hanya berupa pointer untuk data pelatihan yang telah dienkripsi.

```
workers[1]._objects
```

```
{2028246868: tensor([-3878389477496664558,
-387622327283577266,  5531063561060065138,
          8344534316387867658, -1926610631375143425,
-1313606739866655464,
          3128511453031411831,  4824455919677893170,
-7024949952183029989,
         -5378928876356312115]),
 3183306624: tensor([[[[-2200733108487317433,
-4168398854395378014,  8448968655615509883,
               ..., -8652365685364841079,
-3843903902128735008,
             -5859542374425493198],
            [-3187616045760830294,
-6035478300740364591, -8538507452478295933,
               ...,  1701575479649842174,
-4120126030905203595,
             -8811629996638583622],
            [ 3669465359476683044,
7730570084091472606,  6044521298639878363,
               ...,  3980902388756671194,
-8428526494753962432,
              4593162795809799164],
            ...,
            [  978868789422909912,
-5516850901813555995, -8559604770826411 66,
               ...,  3489227996432367251,
-899114660569253536,
              9082274515306680378],
            [ 1452952510572157452,
8935355808459654713,  4774766064324174162,
               ...,  4527842714552895636,
-6582233588372681126,
             -1900708499050302662],
            [-2014569333436101871,
-4688060462064694249, -1567694277037251903,
               ..., -4791596992722148962,
-1353966409309565456,
              5152738911977200121]]],
...
}
```

Dapat dilihat pada data di atas, data yang terdapat pada bob adalah data yang sudah dienkripsi. Dalam hal ini karena bob hanyalah *worker* yang akan melatih data, maka bob tidak memiliki akses untuk mengetahui data tersebut.

Secara teknis, model pembelajaran mesin yang digunakan

adalah ANN sederhana yang memiliki 2 buah *hidden layer* dengan ukuran masing-masing 128 dan 64. Kemudian *input layer*-nya akan berupa node dengan ukuran 28*28 yang merupakan pixel gambar yang telah dikecilkan. Untuk *output layer* sendiri tentu saja terdiri dari 10 layer karena persoalan ini adalah klasifikasi digit tulisan tangan dari angka 0 sampai 9

```
class Net(nn.Module):
  def __init__(self):
    super(Net, self).__init__()
    self.fc1 = nn.Linear(28 * 28, 128)
    self.fc2 = nn.Linear(128, 64)
    self.fc3 = nn.Linear(64, 10)

  def forward(self, x):
    x = x.view(-1, 28 * 28)
    x = F.relu(self.fc1(x))
    x = F.relu(self.fc2(x))
    x = self.fc3(x)
    return x
```

IV. Hasil Implementasi

Di bagian ini, kita akan melihat hasil dari pelatihan model dengan menggunakan data enkripsi. model ini juga akan dibandingkan dengan hasil pelatihan model dengan menggunakan data biasa (tidak dienkripsi). Masing-masing pelatihan akan dilakukan selama 10 epoch.

```
def train(args, model, private_train_loader,
optimizer, epoch):
  model.train()
  for batch_idx, (data, target) in
enumerate(private_train_loader):
    start_time = time.time()
    optimizer.zero_grad()

    output = model(data)
    batch_size = output.shape[0]
    loss = ((output -
target)**2).sum().refresh()/batch_size

    loss.backward()
    optimizer.step()

    if batch_idx % args.log_interval == 0:
      loss = loss.get().float_precision()
      print('Train Epoch: {} [{}/{} ({:.0f}%)]\tLoss:
{:.6f}\tTime: {:.3f}s'.format(
        epoch, batch_idx * args.batch_size,
len(private_train_loader) * args.batch_size,
        100. * batch_idx / len(private_train_loader),
loss.item(), time.time() - start_time))
```

Pada hasil pelatihan dengan data yang terenkripsi dapat dilihat hasilnya di bawah ini.

```
Train Epoch: 1 [0/640 (0%)]     Loss: 1.128000  Time:
6.416s
Train Epoch: 1 [64/640 (10%)]   Loss: 1.004000  Time:
5.873s
Train Epoch: 1 [128/640 (20%)]  Loss: 0.986000  Time:
6.434s
Train Epoch: 1 [192/640 (30%)]  Loss: 0.900000  Time:
6.397s
Train Epoch: 1 [256/640 (40%)]  Loss: 0.884000  Time:
6.150s
Train Epoch: 1 [320/640 (50%)]  Loss: 0.878000  Time:
6.573s
Train Epoch: 1 [384/640 (60%)]  Loss: 0.854000  Time:
6.647s
Train Epoch: 1 [448/640 (70%)]  Loss: 0.848000  Time:
6.390s
Train Epoch: 1 [512/640 (80%)]  Loss: 0.827000  Time:
6.202s
Train Epoch: 1 [576/640 (90%)]  Loss: 0.839000  Time:
6.149s

Test set: Accuracy: 223.0/640 (35%)

...

Train Epoch: 10 [0/640 (0%)]    Loss: 0.364000  Time:
6.065s
Train Epoch: 10 [64/640 (10%)]  Loss: 0.278000  Time:
6.081s
Train Epoch: 10 [128/640 (20%)] Loss: 0.401000  Time:
6.098s
Train Epoch: 10 [192/640 (30%)] Loss: 0.306000  Time:
5.950s
Train Epoch: 10 [256/640 (40%)] Loss: 0.306000  Time:
6.148s
Train Epoch: 10 [320/640 (50%)] Loss: 0.289000  Time:
6.260s
Train Epoch: 10 [384/640 (60%)] Loss: 0.326000  Time:
6.323s
Train Epoch: 10 [448/640 (70%)] Loss: 0.369000  Time:
6.250s
Train Epoch: 10 [512/640 (80%)] Loss: 0.350000  Time:
6.246s
Train Epoch: 10 [576/640 (90%)] Loss: 0.437000  Time:
6.184s

Test set: Accuracy: 484.0/640 (76%)
```

Sedangkan hasil pelatihan dengan data yang tidak dienkripsi dapat dilihat hasilnya di bawah ini.

```
Train Epoch: 1 [0/640 (0%)]     Loss: 0.146695  Time:
0.020s
Train Epoch: 1 [64/640 (10%)]   Loss: 0.195114  Time:
1.096s
Train Epoch: 1 [128/640 (20%)]  Loss: 0.176686  Time:
1.111s
Train Epoch: 1 [192/640 (30%)]  Loss: 0.106177  Time:
1.105s
Train Epoch: 1 [256/640 (40%)]  Loss: 0.078360  Time:
1.095s
Train Epoch: 1 [320/640 (50%)]  Loss: 0.078104  Time:
1.108s
Train Epoch: 1 [384/640 (60%)]  Loss: 0.127179  Time:
1.121s
Train Epoch: 1 [448/640 (70%)]  Loss: 0.279967  Time:
1.098s
```

```
Train Epoch: 1 [512/640 (80%)]  Loss: 0.122385  Time:
1.108s
Train Epoch: 1 [576/640 (90%)]  Loss: 0.067995  Time:
1.110s

Epoch 1 - Training loss: 0.150366864433246

...

Train Epoch: 10 [0/640 (0%)]    Loss: 0.069421  Time:
0.016s
Train Epoch: 10 [64/640 (10%)]  Loss: 0.047578  Time:
1.069s
Train Epoch: 10 [128/640 (20%)] Loss: 0.022107  Time:
1.080s
Train Epoch: 10 [192/640 (30%)] Loss: 0.020558  Time:
1.079s
Train Epoch: 10 [256/640 (40%)] Loss: 0.073139  Time:
1.085s
Train Epoch: 10 [320/640 (50%)] Loss: 0.054382  Time:
1.082s
Train Epoch: 10 [384/640 (60%)] Loss: 0.055884  Time:
1.110s
Train Epoch: 10 [448/640 (70%)] Loss: 0.070654  Time:
1.088s
Train Epoch: 10 [512/640 (80%)] Loss: 0.013502  Time:
1.050s
Train Epoch: 10 [576/640 (90%)] Loss: 0.236934  Time:
1.048s

Epoch 10 - Training loss: 0.05820022581039922
```

Dari kedua hasil di atas, kita dapat melihat bahwa waktu komputasi pada pelatihan dengan data yang terenkripsi jauh lebih lambat dibandingkan pelatihan dengan data biasa. Secara khusus, iterasi lebih dari 1 kumpulan 64 item membutuhkan 6 detik pada data terenkripsi sementara jika menggunakan data biasa hanya membutuhkan waktu 1 detik.

Kemudian dari segi akurasi, model dengan data terenkripsi juga memiliki hasil yang kurang bagus jika dibandingkan dengan model dengan data biasa. Hal ini mungkin karena data yang akan dilatih merupakan data yang teracak sehingga akan semakin sulit bagi model untuk mempelajarinya. Akan tetapi, sebenarnya model sudah cukup baik dengan memberikan nilai akurasi sebesar 75% dengan menggunakan model ANN. Penulis berasumsi hasil akurasi mungkin akan lebih baik lagi jika menggunakan model CNN.

Terakhir, dari segi keamanan, model yang ditawarkan telah mampu mencapai kerahasiaan bagi pemilik data maupun pemilik model. Namun, meskipun SMPC akan memastikan bahwa data pelatihan tidak diakses, namun tetap saja masih terdapat ancaman seperti didapatnya prediksi yang mungkin mengungkapkan informasi tentang data pelatihan. hal ini tentu saja sudah diluar cakupan dari model.

V. Kesimpulan dan Saran

Model pembelajaran mesin dapat dibuat dan sudah memiliki akurasi yang cukup baik untuk menangani data yang terenkripsi dengan menggunakan pustaka PyTorch. Dengan menggunakan SMPC, data uji yang akan dilatih akan terenkripsi sehingga tidak akan ada orang yang dapat memahami data kecuali orang memiliki data tersebut. Dari hasil implementasi model pembelajaran mesin yang telah dibuat, akurasi yang diperoleh dengan menggunakan data terenkripsi memang lebih buruk dibanding dengan data yang tidak dienkripsi. Namun tentu saja dengan metode ini memiliki keistimewaan yaitu kerahasiaan data.

Untuk pengembangan selanjutnya, penulis menyarankan untuk mengubah model *neural network* menjadi *Convolutional Neural Network* yang lebih cocok digunakan untuk menangani jenis data seperti persoalan MNISt ini. Selain itu, penulis juga menyarankan untuk menggunakan teknik *federated learning* untuk semakin menjamin kerahsiaan data.

VI. Ucapan Terima Kasih

Pertama-tama, penulis mengucapkan rasa syukur dan terima kasih untuk Allah SWT, Tuhan Yang Maha Esa atas limpahan syukur dan rahmatnya yang memberi kemudahan sehingga penulis masih dapat menempuh studi di Teknik Informatika Institut Teknologi Bandung dengan selamat.

Selanjutnya penulis mengucapkan terima kasih untuk kedua orang tua dan keluarga penulis yang selalu mendoakan dan mendukung kesuksesan penulis. Penulis juga ingin mengucapkan banyak terima kasih atas limpahan ilmu yang senantiasa diberikan oleh Bapak Rinaldi Munir selaku dosen mata kuliah Kriptografi yang sangat membantu penulis dalam penyusunan makalah ini.

Terakhir, penulis ingin mengucapkan terima kasih kepada teman-teman satu angkatan UNIX 2017 terkhusus untuk teman-teman kelas K3 IF 2017 terutama Asif, Irfan, Faiz, Abda, Hanif, Winston, and Jofiandy yang telah peduli dan membantu perkembangan studi penulis termasuk dalam penyusunan makalah ini.

Referensi

[1] Menezes, A. J., Van Oorschot, P. C., & Vanstone, S. A. (2018). Handbook of applied cryptography.

[2] Munir, Rinaldi. "Pengantar Kriptografi." Informatika, Bandung (2006).

[3] Goldreich, Oded. "Secure multi-party computation." Manuscript. Preliminary version 78 (1998).

[4] Michie, Donald, David J. Spiegelhalter, and C. C. Taylor. "Machine learning." Neural and Statistical Classification 13.1994 (1994): 1-298.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 18 Desember 2020

Juniardi Akbar 13517075