Topik : 4.3. Evaluasi Trade-off

Objective : Uji beberapa nilai noise\_multiplier → analisis akurasi vs ε

Task : Plot hasil trade-off

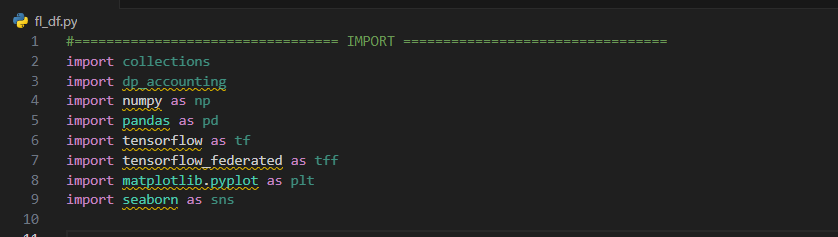
Source : <https://www.tensorflow.org/federated/tutorials/federated_learning_with_differential_privacy>

**Differential Privacy in TFF**

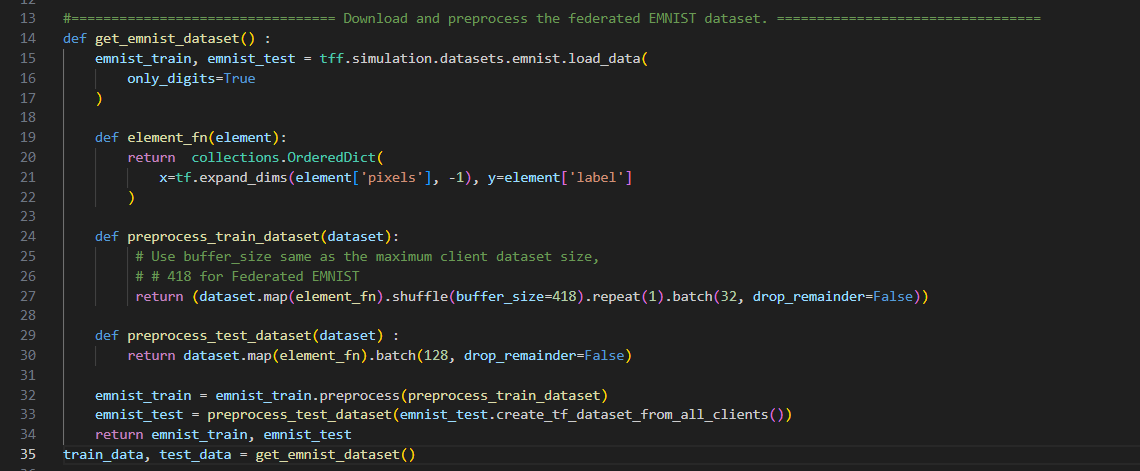
Differential Privacy (DP) Adalah metode yang banyak digunakan untuk membatasi dan mengukur kebocoran privasi data sensitive Ketika melakukan tugas pembelajaran. Melatih model dengan DP Tingkat pengguna menjamin bahwa model tidak akan belajar sesuatu yang signifikan tentang data individu tertentu, tetapi tetap dapat mempelajari pola yang ada pada data dari banyak client.

Kita akan melatih model pada dataset **federated EMNIST**. Terdapat trade-off yang melekat antara utilitas dan privasi, sehingga mungkin sulit untuk melatih model dengan tingkat privasi tinggi yang memiliki performa setara dengan model *non-private* terbaik. Demi kecepatan dalam tutorial ini, kita hanya akan melatih selama **100 putaran**, sehingga mengorbankan sebagian kualitas demi mendemonstrasikan cara melatih dengan privasi tinggi. Jika kita menggunakan lebih banyak putaran pelatihan, kita tentu bisa mendapatkan model privat dengan akurasi yang agak lebih tinggi, tetapi tidak akan setinggi model yang dilatih tanpa DP.

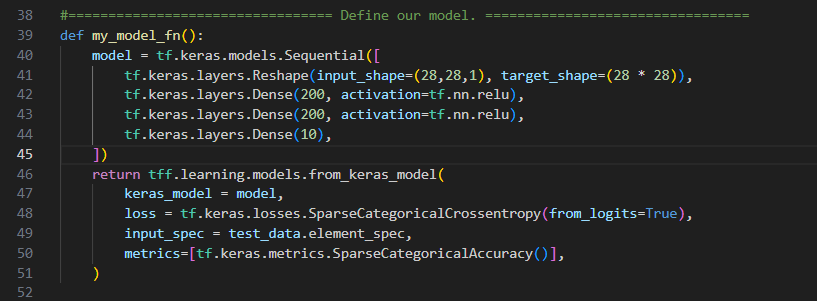
1. Import



1. Download and pre-process the federated EMNIST dataset.



1. Define Our Model



1. Determine the noise sensitivity of the model

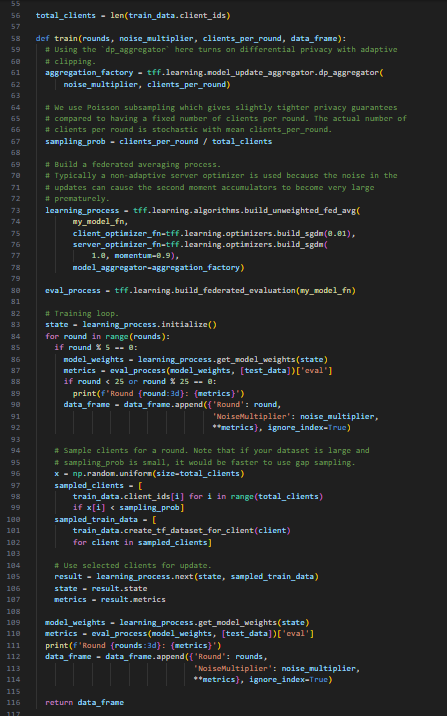
Untuk mendapatkan jaminan Differential Privacy (DP) di level pengguna, kita perlu mengubah algoritma dasar Federated Averaging dengan dua cara :

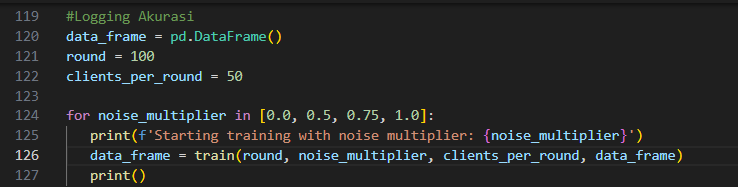
1. Update model dari klien harus di-clip sebelum dikirim ke server, sehingga membatasi pengaruh maksimum dari setiap klien.
2. Server harus menambahkan noise yang cukup pada jumlah total update pengguna sebelum melakukan averaging, agar pengaruh terburuk dari satu klien dapat tersamarkan.

Untuk proses *clipping*, kita menggunakan metode *adaptive clipping* dari Andrew dkk. (2021), *Differentially Private Learning with Adaptive Clipping*, sehingga tidak perlu menetapkan norma clipping secara eksplisit.

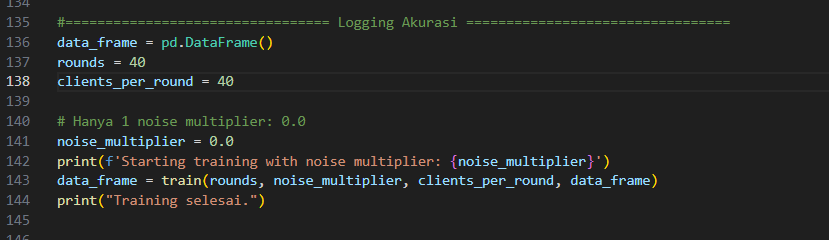
Penambahan noise umumnya akan menurunkan utilitas model, tetapi kita bisa mengontrol jumlah noise pada setiap rata – rata update dengan dua parameter : standar deviasi dari noise Gaussian yang ditambahkan pada jumlah total , serta jumlah klien yang dilibatkan dalam rata – rata. Strategi kita Adalah pertama – tama menentukan seberapa besar noise yang masih bisa ditoleransi model dengan jumlah klien per ronde yang relative kecil tanpa kehilangan kualitas model yang signifikan. Kemudian , untuk melatih model final, kita bisa meningkatkan jumlah noise pada aggregasi, sambil secara proposional menambah jumlah klien per ronde (dengan asumsi dataset cukup besar untuk mendukung jumlah klien tersebut). Hal ini tidak mungkin berdampak signifikan pada kualitas model , karena efeknya hanya mengurangi variansi akibat *sampling* klien (dan memang akan kita lakukan verifikasi bahwa ini tidak memengaruhi kualitas pada kasus kita).

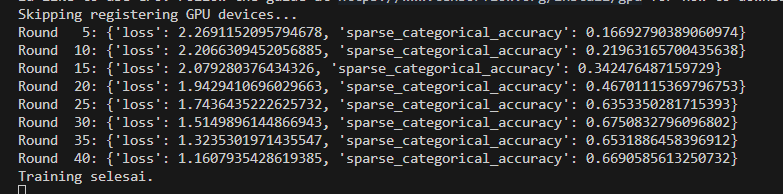
Untuk itu, kita akan melatih serangkaian model dengan 50 klien per ronde, dengan jumlah noise yang meningkat. Secara khusus, kita meningkatkan nilai *noise\_multiplier,* yaitu rasio standar deviasi noise terhadap clipping norm. kerena kita menggunakan *adaptive clipping,* maka besar actual dari noise akan berubah dari ronde ke ronde.

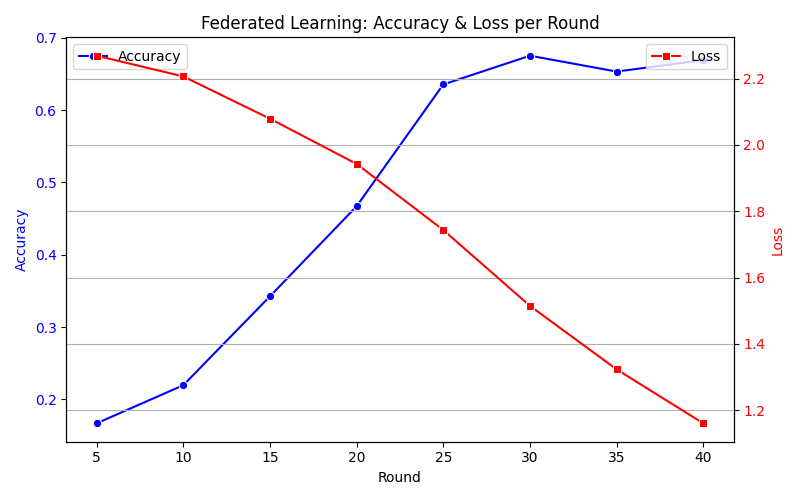




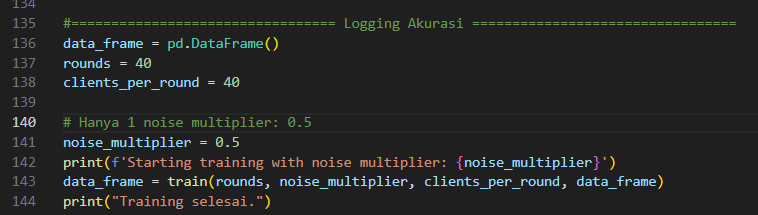
Starting training with noise multiplier: 0.0

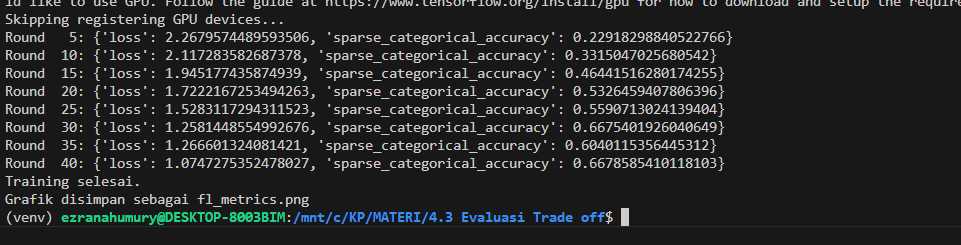


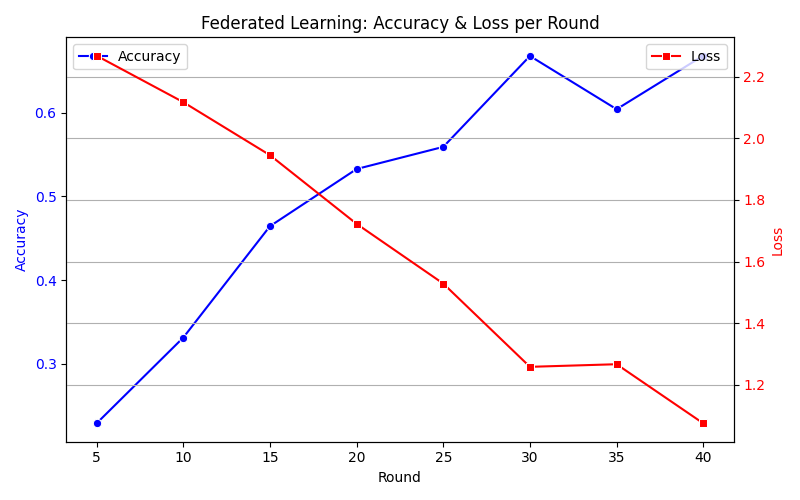




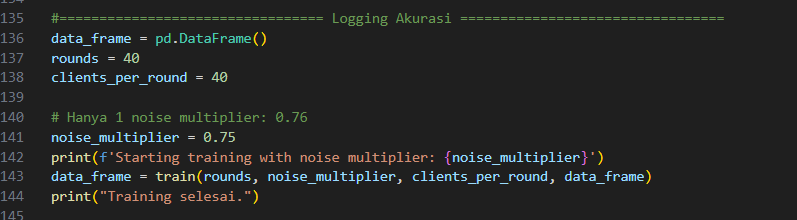
Starting training with noise multiplier: 0.5

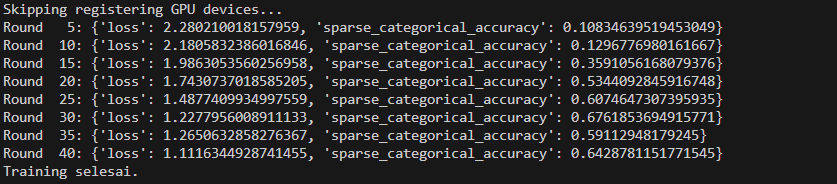


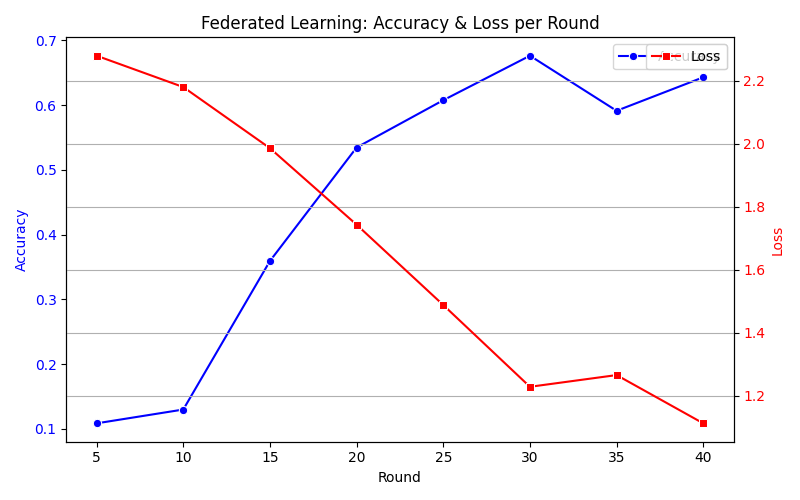




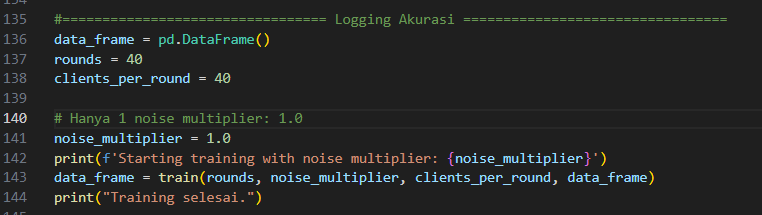
Starting training with noise multiplier: 0.75

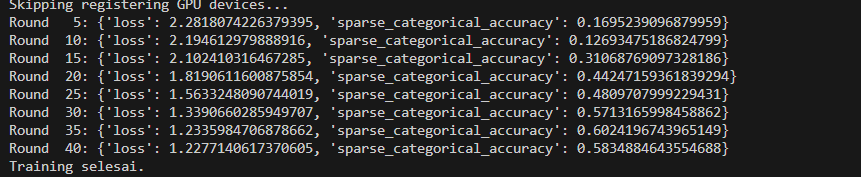


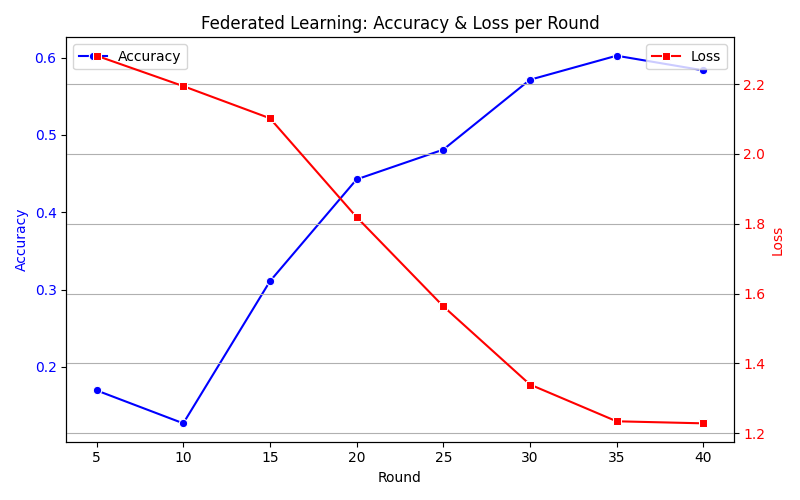




Starting training with noise multiplier: 1.0



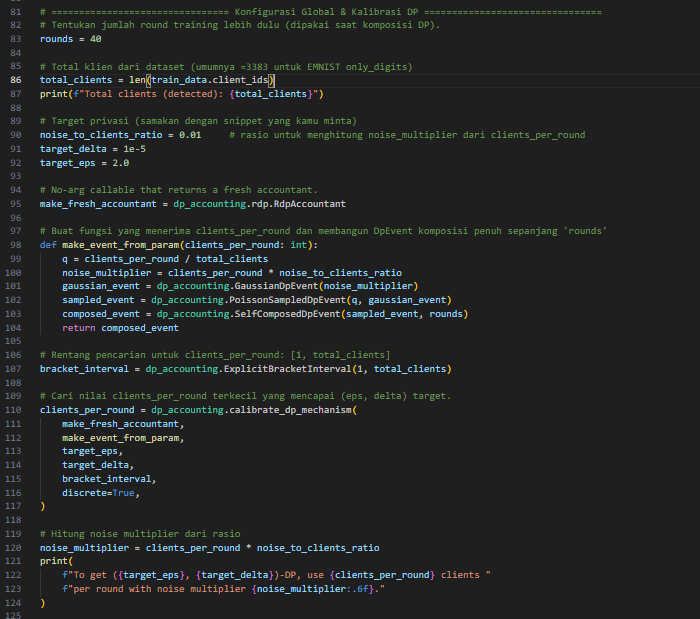


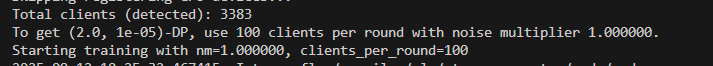


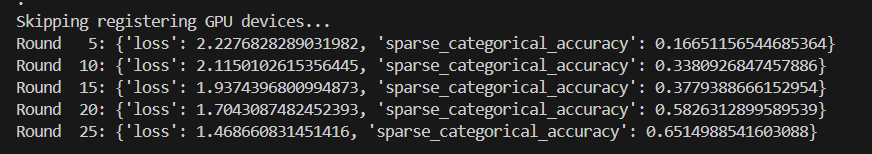
Biasanya ada pertukaran antara kualitas model dan privasi. Semakin tinggi noise yang kita gunakan, semakin banyak privasi yang bisa kita dapatkan untuk jumlah waktu pelatihan dan jumlah klien yang sama. Sebaliknya, dengan noise yang lebih sedikit, kita mungkin mendapatkan model yang lebih akurat, tetapi kita harus melatih dengan lebih banyak klien per putaran untuk mencapai tingkat privasi yang kita targetkan.

Sekarang kita dapat menggunakan fungsi dp\_accounting untuk menentukan berapa banyak klien yang diharapkan per putaran yang kita butuhkan untuk mendapatkan privasi yang dapat diterima. Praktik standar adalah memilih delta yang agak lebih kecil dari satu dibagi jumlah rekaman dalam kumpulan data. Kumpulan data ini memiliki total 3383 pengguna pelatihan, jadi mari kita targetkan (2, 1e-5)-DP.

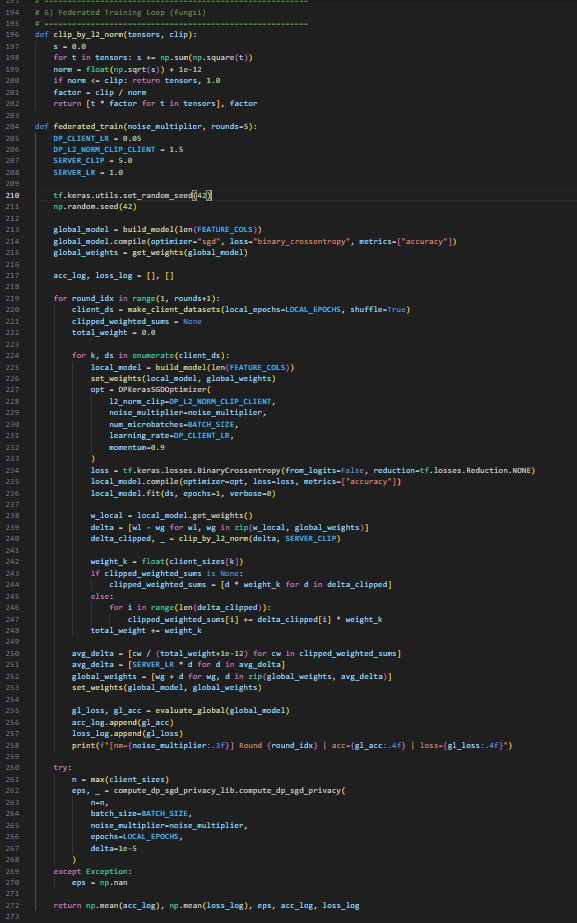
Kami menggunakan dp\_accounting.calibrate\_dp\_mechanism untuk mencari jumlah klien per putaran. Akuntan privasi (RdpAccountant) yang kami gunakan untuk memperkirakan privasi yang diberikan dp\_accounting.DpEvent didasarkan pada Wang et al. (2018) dan Mironov et al. (2019).

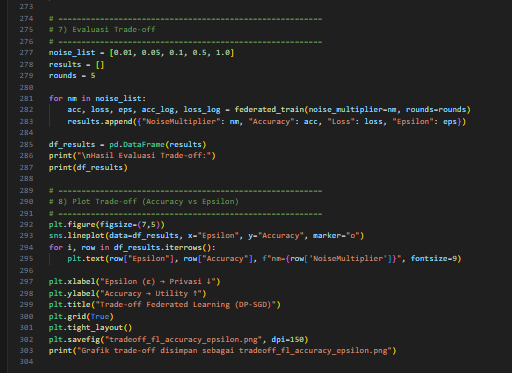






**Task : Plot Hasil Trade-off**

****

****

