Topik : 1.2. Arsitektur FL

Objective : Memahami arsitektur server-client dan mekanisme update

Task : Gambar diagram alur FL dengan server dan beberapa client

Source : https://www.tensorflow.org/federated

TensorFlow Federated: Machine Learning on Decentralized Data

TensorFlow Federated (TFF) Adalah kerangka kerja sumber terbuka untuk melakukan pembelajaran machine dan komputasi lainnya pada data yang di desentralisasi. TFF telah dikembangkan untuk memfasilitasi penelitian dan eksperimen terbuka dengan Federated Learning (FL), yang merupakan sebuah pendekatan pembelajaran mesin Dimana model global Bersama dilatih oleh banyak klien yang berpartisipasi yang menyimpan data pelatihan mereka secara local. Misalnya, FL telah digunakan untuk melatih model prediksi untuk keyboard seluler tanpa mengunggah data pengetikan sensitive ke server.

TFF memungkinkan pengembang untuk mensimulasikan algoritma pembelajaran gabungan yang disertakan pada model dan data mereka, serta bereksperimen dengan algoritma baru. Blok penyusun yang disediakan oleh TFF juga dapat digunakan untuk mengimplementasikan komputasi non-pembelajaran. Antarmuka TFF diatur dalam dua lapisan utama :

1. Federated Learning (FL) API

   Lapisan ini menawarkan serangkaian antarmuka Tingkat tinggi yang memungkinkan pengembang menerapkan implementasi pelatihan dan evaluasi gabungan yang disertakan ke model TensorFlow yang sudah ada. Lapisan ini menawarkan serangkaian antarmuka Tingkat tinggi yang memungkinkan pengembang menerapkan implementasi pelatihan dan evaluasi gabungan yang disertakan ke model TensorFlow yang ada.
2. Federated Core (FC) API

   Inti dari system ini Adalah seperangkat antarmuka Tingkat rendah untuk mengekspresikan algoritma gabungan baru secara ringkas dengan menggabungkan Tensorflow dengan operator komunikasi terdistribusi dalam lingkungan pemrograman fungsional yang sangat spesifik. Lapisan ini juga berfungsi sebagai fondasi yang kami gunakan untuk membangun federasi.

TFF memungkinkan pengembang untuk mengekspresikan komputasi gabungan secara deklaratif, sehingga dapat diterapkan ke lingkungan runtime yang beragam. Termasuk dengan TFF adalah runtime simulasi multi-mesin berkinerja untuk eksperimen.

Source : https://www.tensorflow.org/federated/federated_learning

1. Tujuan TensorFlow Federated (TFF)

   TensorFlow Federated (TFF) adalah framework untuk melakukan **Federated Learning (FL)**, yaitu melatih atau mengevaluasi model machine learning secara terdistribusi di berbagai perangkat (misalnya ponsel) tanpa harus mengumpulkan data di server pusat. Tujuan TFF adalah:

   - Memudahkan eksperimen federated learning tanpa harus memahami detail teknisnya.
   - Mendukung berbagai jenis model TensorFlow yang sudah ada.
   - Mendukung penelitian dan pengembangan melalui dataset simulasi.

2. Komponen Utama

   - Models
     - Model Tensorflow/Keras bisa dibungkus ke dalam TFF.
     - Model harus bisa di serialisasi ke bentuk graph TensorFlow, agar bisa dijalankan di perangkat yang terbatas (misalnya smarthphone)
     - TFF tidak dapat menggunakan model yang sudah dibangun sebelumnya; sebaliknya, logika definisi model dikemas dalam fungsi tanpa argumen yang mengembalikan `tff.learning.models.VariableModel`.
     - Mendukung konversi langsung dari Keras lewat `tff.learning.models.from_keras_model`.

   - Federated Computation Builders
     - Funggsi pembangun untuk membuat perhitungan federated, seperti (federated averaging) dan evaluasi
     - Proses Terdiri dari
       - Compile → kode model + distribusi komunikasi diserialisasi

        - Execute → dieksekusi, biasanya dalam simulasi local
    - Proses pelatihan bersifat stateful (model parameter diperbarui tiap ronde) Ditangani denngan `tff.templates.IterativeProcess`

- Available Builders
    - Saat ini, TFF menyediakan berbagai fungsi pembangun yang menghasilkan perhitungan federasi untuk pelatihan dan evaluasi federasi. Dua contoh yang menonjol meliputi:
    - `tff.learning.algorithms.build_weighted_fed_avg` yang menerima fungsi model dan optimizer klien sebagai input, dan mengembalikan objek `tff.learning.templates.LearningProcess` yang merupakan subkelas dari `tff.templates.IterativeProcess`
    - `tff.learning.build_federated_evaluation` menerima fungsi model dan mengembalikan perhitungan federasi tunggal untuk evaluasi federasi model, karena evaluasi tidak bersifat berstatus.
- Datasets
    - Dataset simulasi tersedia di `tff.simulation.datasets`
    - Data federated di representasikan sebagai list Python → tiap elemen = dataset 1 clien
    - Interface standar : `tff.simulation.datasets.ClientData`
    - Bisa memilih subset klien untuk melakukan simulasi

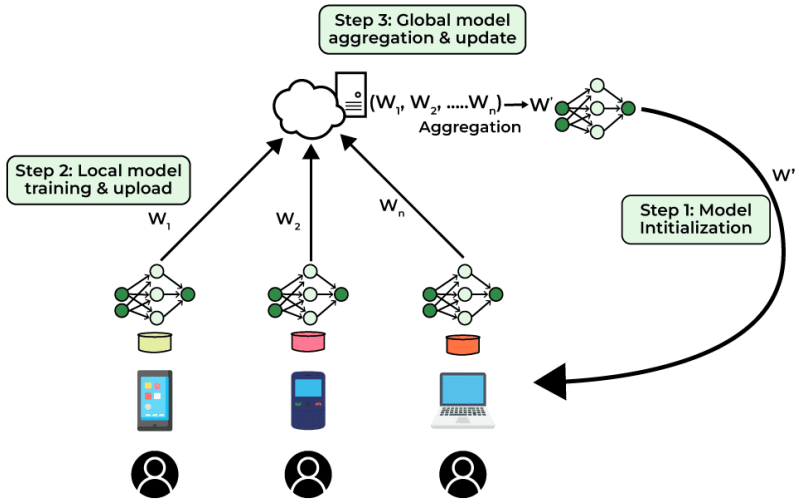
3. Agregasi

   Dalam Federated Learning ada 2 tahap agregasi :

   - Lokal (di perangkat klien) → model dilatih per batch, lalu menghitung metrik seperti loss/accuracy
   - Federated (lintas klien) → server menggabungkan parameter model & metrix dari banyak klien

   Agregasi federated otomatis ditangani TFF, tapi pengguna bisa mengontrol cara agregasinya.

Source : https://www-geeksforgeeks-org.translate.goog/machine-learning/collaborative-learning-federated-learning/?_x_tr_sl=en&_x_tr_tl=id&_x_tr_hl=id&_x_tr_pto=imgs



- Langkah 1 : perangkat – perangakat tertentu akan mengunduh model terkini
- Langkah 2 : model akan membuat perbaikan dari data baru yang di peroleh dari perangkat
- Langkah 3 : perubahan model di rangkum sebagai pembaruan dan di komunikasikan ke cloud (komunikasi ini di enkripsi)
- Langkah 4 : di cloud, terdapat banyak pembaruan yang datang dari berbagai pengguna. Semua pembaruan ini digabungkan dan model akhir di bangun

Jadi, tidak ada data dalam jumlah besar yang diunggah ke cloud, dan model juga dilatih dengan berbagai data tersebut. Dalam proses ini, data yang telah dilatih tersimpan di ponsel pintar/perangkat seluler Anda.

Langkah 1: Unduh model awal
Setiap perangkat (misalnya HP atau laptop) akan ambil model awal dari server. Jadi semua mulai dari titik yang sama biar adil.

Langkah 2: Latih model di perangkat
Model ini kemudian belajar dari data yang ada di perangkat masing-masing. Contohnya HP belajar dari data chat, aplikasi, atau foto yang ada di dalam HP itu sendiri.

Langkah 3: Kirim hasil belajar ke server
Setelah selesai belajar, perangkat nggak ngirim data mentah (kayak foto atau chat), tapi cuma ngirim hasil perbaikan modelnya. Hasil ini dikirim ke server dengan aman karena sudah dienkripsi.

Langkah 4: Gabungkan semua hasil belajar
Di server, semua hasil belajar dari banyak perangkat digabungin jadi satu model baru yang lebih pintar. Model terbaru ini kemudian bisa dibagikan lagi ke semua perangkat biar siklusnya berulang.