論文輪講
Towards Federated Learning at Scale: System Design

杉浦 圭祐

慶應義塾大学理工学部情報工学科 松谷研究室

May 5, 2019

目次

目次

Federated Learning とは

- Federated Learning とは
  - 分散機械学習の新手法
    ⇒ 複数のデバイス間に分散したデータを利用し、共通のモデルを学習
  - 既存の分散機械学習の手法とは異なり、プライバシー等の問題を解決
    ⇐ 学習は各デバイス上で行われ、その結果が共通のモデルに反映される

Federated Learning とは

- 一般的な機械学習との違い
  - 学習に使用する訓練データは、クラウド上には保存されない
    ⇒ 全てのデータは、各デバイスに残されたままである
    ⇒ プライバシーや、データの所有権の問題に対処できる
  - クラウド上では、モデルの学習を行わない
    ⇒ モデルの学習は、各デバイス上で (オンデバイスで) 行われる
    ⇒ 学習時には、自身のデバイス上のデータを用いる
  - 各デバイスは、モデルを使用した推論だけでなく、学習も行う
    ⇒ 学習後、クラウド上にある共通のモデルの、パラメータを更新
    ⇒ 各デバイスからクラウドへは、モデルの更新情報のみ送信
  - 各デバイスで学習されたモデルを、即座に利用できる
    ⇒ クラウド上のモデルをベースとして、各デバイス向けにカスタマイズ可能

Federated Learning とは

- Federated Learning の大まかな流れ
  1. 各デバイスが、クラウド上にある現在のモデルをダウンロードする
  2. デバイス上のデータを使って、モデルを学習する
  3. 学習が終わったら、モデルのパラメータの変更内容 (差分) をまとめる
  4. 差分をクラウドに送信し、クラウド上の共通のモデルに反映させる
  5. (1) から (4) までを、繰り返し行う

Federated Learning とは

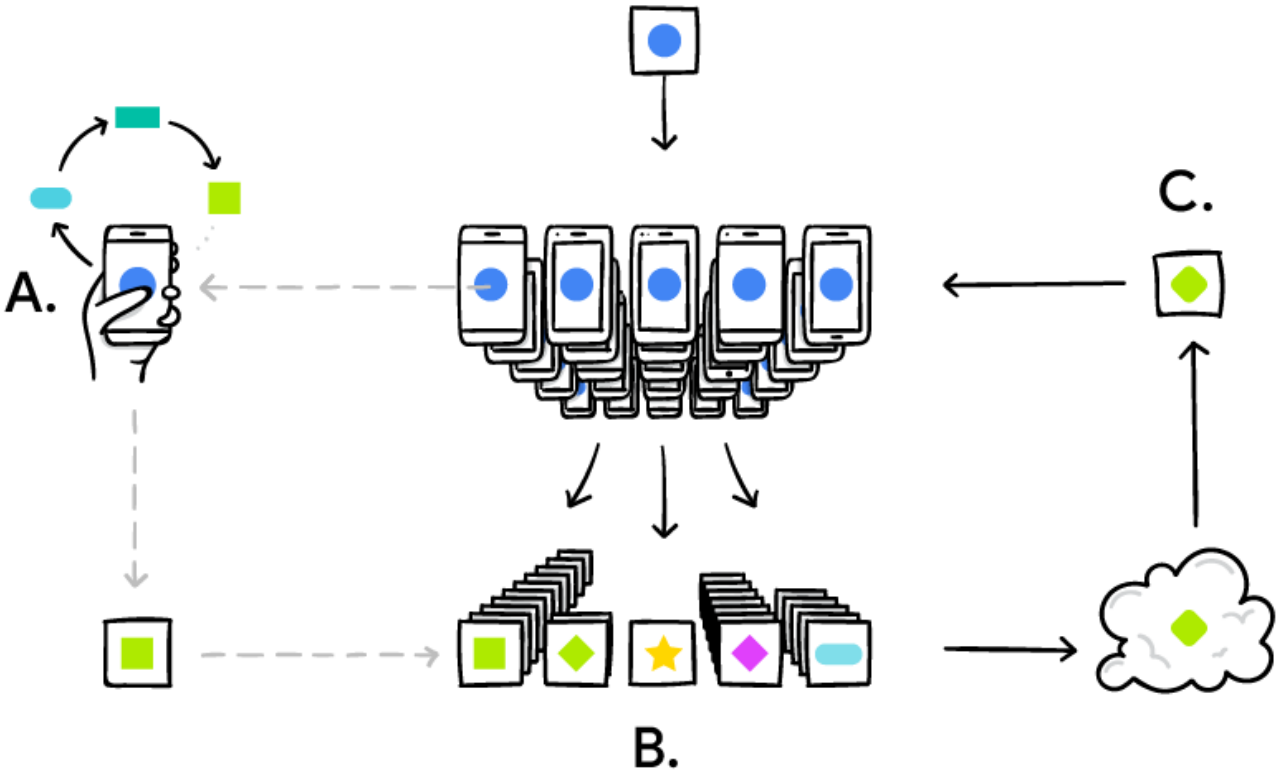

図 1: Federated Learning の流れ [?]

目次

システムの概要

- システムの概要
  - モバイル端末 (Android のスマートフォン) を対象としたシステム
  - スケーラブルかつ、実際の製品にもデプロイ可能なレベル
    ⇐ Gboard(Google Keyboard) というキーボードアプリで実際に使用
  - 実装には、TensorFlow が用いられている
    ⇐ 深層ニューラルネットの学習も可能である
  - 同期型の訓練アルゴリズム (Federated Averaging) を採用
  - セキュリティ向上のための手法 (Secure Aggregation) を利用可能

同期型の訓練アルゴリズム

- 同期型の訓練アルゴリズムが採用された
  - 具体的には、Federated Averaging というアルゴリズムを使用
    ⇐ SGD(Stochastic Gradient Descent) とよく似ている
    ⇐ SGD を、重み付きの更新によって拡張したような手法

- クラウド上のモデルが更新されるまでの流れ
  1. 各デバイスから、差分データ (モデルのパラメータの更新情報) を受信
  2. Federated Averaging を用いて、クラウド上で差分データを一つに集約
  3. 集約された差分を、モデルのパラメータに反映 (モデルの更新)
  4. 更新されたモデルを、各デバイスが取得できるようになる

同期型の訓練アルゴリズム

- 同期型の訓練アルゴリズムが採用された
  1. 近年、同期型の訓練アルゴリズムを採用する動きがみられるため
     ⇐ 同期処理の負担が大きなデータセンタですら、同期型の訓練アルゴリズムを採用する傾向
  2. プライバシーを強化する手法を適用するため
     ⇐ Differential Privacy や Secure Aggregation などの手法がある
     ⇐ これらは原則として、同期型のアルゴリズムでないと適用不可能
  3. サーバ側での処理が単純になるため
     ⇐ 多数のユーザ (デバイス) からの更新データをまとめて、モデルに適用

  - 但し、同期処理のオーバヘッドを軽減するための対策が必要 (後述)

セキュリティ向上のための手法

- セキュリティ向上のための手法を利用可能
  - 具体的には、Secure Aggregation という手法を利用可能
    ⇐ 各デバイスから送信される差分データは、外部から隠される
  - Federated Learning では、訓練データは各デバイス上に留まる
    ⇐ 訓練データには、個人を特定するに足る情報が含まれるかもしれない
    ⇐ クラウド上にデータを保存しないことで、プライバシーが確保される
  - データは送信しない代わりに、データを元に算出した差分データを送信
    ⇐ 差分データには、各デバイスを特定するのに十分な情報が、依然として含まれる可能性
    ⇐ 差分データをも隠蔽することで、セキュリティを更に向上させられる

システムを実装する上での課題

- システムを実装する上での課題が非常に多い
  1. デバイスが常に接続されているとは限らない
     ⇐ デバイス (スマートフォン) の接続状態は不安定になりがち
  2. デバイスが常に計算可能とは限らない
     ⇐ 計算が途中で中断させられるかもしれない
     ⇐ デバイスは世界中に散らばって存在するため、地理的な要因 (タイムゾーンなど) を考慮する必要がある
  3. 複数のデバイスの同期処理を取るのが困難
     ⇐ 前述の通り、これらのデバイスは接続状態が不安定で、常に利用できるかどうかも分からない
  4. デバイスの計算能力とストレージの制限が厳しい
     ⇐ 深層ニューラルネットの場合はパラメータ数が多く、メモリと計算資源の消費が特に大きい

システムを実装する上での課題

- これらの課題を 3 つの構成要素で解決
  - 通信プロトコル、デバイス、サーバ
  - この 3 つの動作について、これからみていく

  - 論文の著者によれば、このシステムは数百万、あるいは十億台のデバイス上で利用可能だとしている

目次

通信プロトコルの用語整理

- プロトコルの主人公
  - デバイス (ここでは Android のスマートフォン)
  - FL Server (クラウドベースの分散サービス)

- FL Population
  - 学習アルゴリズムで解こうとしている問題

- FL Task
  - 特定の計算タスク
    ⇐ あるハイパーパラメータが与えられた下での学習
    ⇐ デバイス上のローカルなデータを用いた、モデルの評価

  - FL Population は、複数の FL Task によって構成される
    ⇒ FL Task は、必ず何らかの FL Population に属する

通信プロトコルの用語整理

- FL Plan
  - TensorFlow の計算グラフや、タスクの実行方法を格納するデータ構造
- FL Checkpoint
  - クラウド上の現在のモデルのパラメータ
  - その他の状態 (シリアライズ化された TensorFlow のセッション)
- Round
  - FL Server とデバイスとの一連の通信 (後述)
  - Selection、Configuration、Reporting の 3 段階で構成される
  - 次の図 2 を参照

Federated Learning のプロトコル

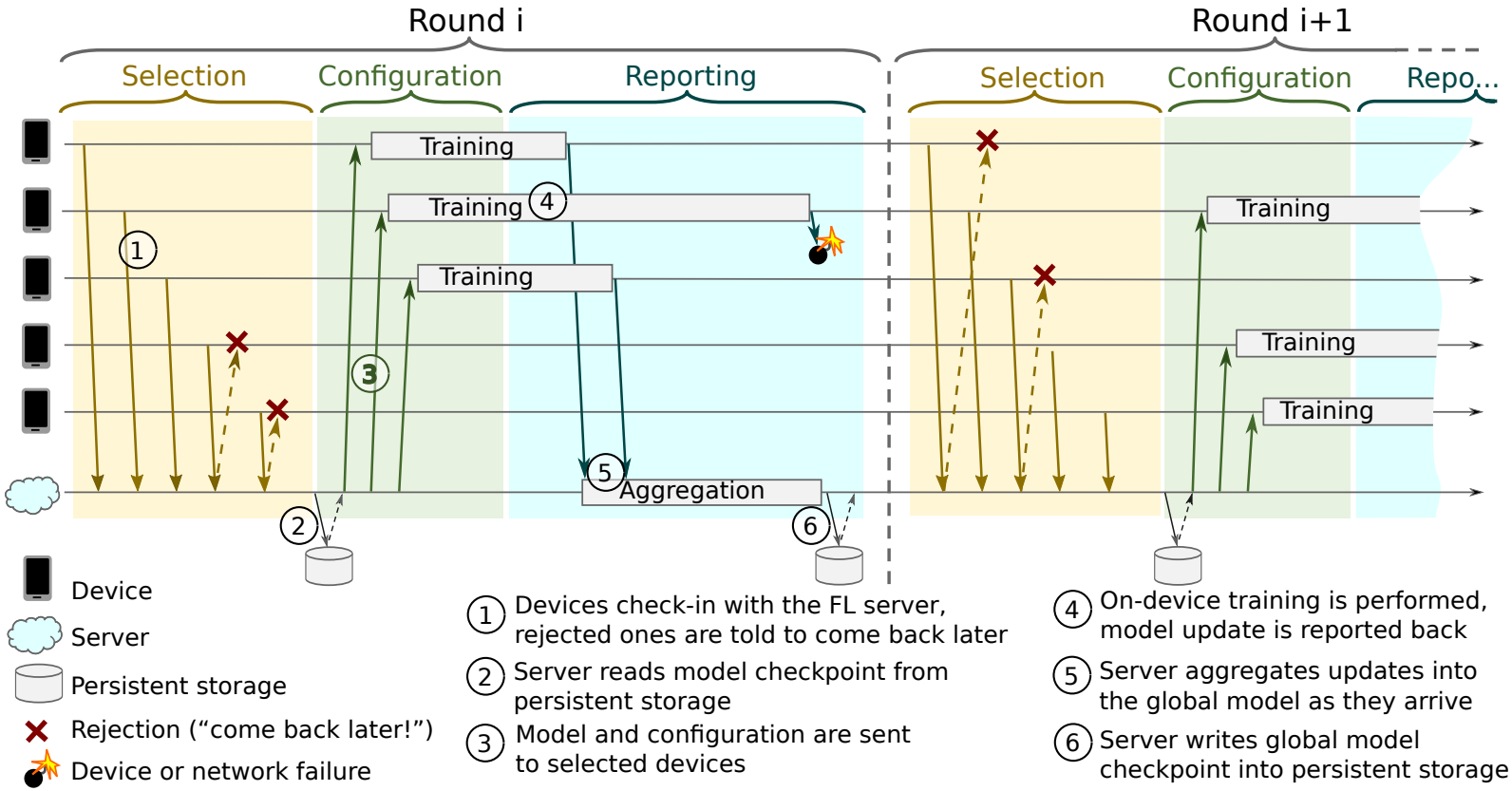

Figure 1: Federated Learning Protocol

図 2: Federated Learning のプロトコル [?]

Federated Learning のプロトコル

- Federated Learning のプロトコルの大まかな流れ
  1. FL Server は、ある決められた時間だけ、デバイスからの報告を待機
     $\Leftarrow$ ある特定の FL Task が実行可能であることの報告を待つ
  2. 多数のデバイスが、指定された FL Task を実行可能であることを、FL Server に伝達
  3. FL Server は、報告してきた数千のデバイスの中から、数百程度のデバイスを選択
  4. 選ばれた数百のデバイスで、FL Task を実行する
  5. (1) から (4) までを繰り返す
     $\Rightarrow$ この繰り返しの単位を Round という
     $\Rightarrow$ (1) から (3) までが Selection フェーズ
     $\Rightarrow$ (4) が Configuration と Reporting フェーズ

Federated Learning のプロトコル

- Federated Learning のプロトコルの Round の流れ
  - Round は Selection、Configuration、Reporting の 3 段階で構成される
  - Round の間は、選択されたデバイスは FL Server との通信を継続する

  - Round の実行途中で、時間内に応答しないデバイスは単に無視される
  - Federated Protocol のプロトコルは、このようなデバイスの脱落を考慮に入れて設計されている

Federated Learning のプロトコル

- Selection フェーズの流れ
  1. FL Task の実行に適したデバイスは、周期的に FL Server にアクセス
     ⇐ 充電中で、かつ Wi-Fi に接続されているデバイス
     ⇐ 従量課金制のネットワークに接続されたデバイスは、アクセスしない (Federated Learning には参加しない)
  2. FL Server にアクセスしたデバイスは、双方向のコネクションを確立
     ⇐ Round の間は、コネクションを維持する必要がある
  3. FL Server は、接続してきた数千のデバイスの中から、数百程度のデバイスを選択
     ⇐ 1 つの Round には数百程度のデバイスが参加
     ⇐ 選択に使用するアルゴリズムは何でもよい (溜池サンプリング)
  4. 選ばれなかったデバイスに対して、FL Server は次にアクセスすべき時刻を送信 (適当な時間の経過後に、再接続させる)

Federated Learning のプロトコル

- Selection フェーズで指定可能なパラメータの例
  - FL Task の実行に協力して欲しいデバイスの数 (希望)
  - FL Task の実行に最低限必要なデバイスの数 (閾値)
  - FL Server がデバイスからの接続を待つべき時間 (タイムアウト)

  - 希望通りの数のデバイスが接続してきた時点で、Round の実行が開始
  - タイムアウトになるまでは、接続デバイス数が希望通りになるまで待機

  - タイムアウト時に、接続デバイス数が閾値を超えていなければ、Round は実行されない

Federated Learning のプロトコル

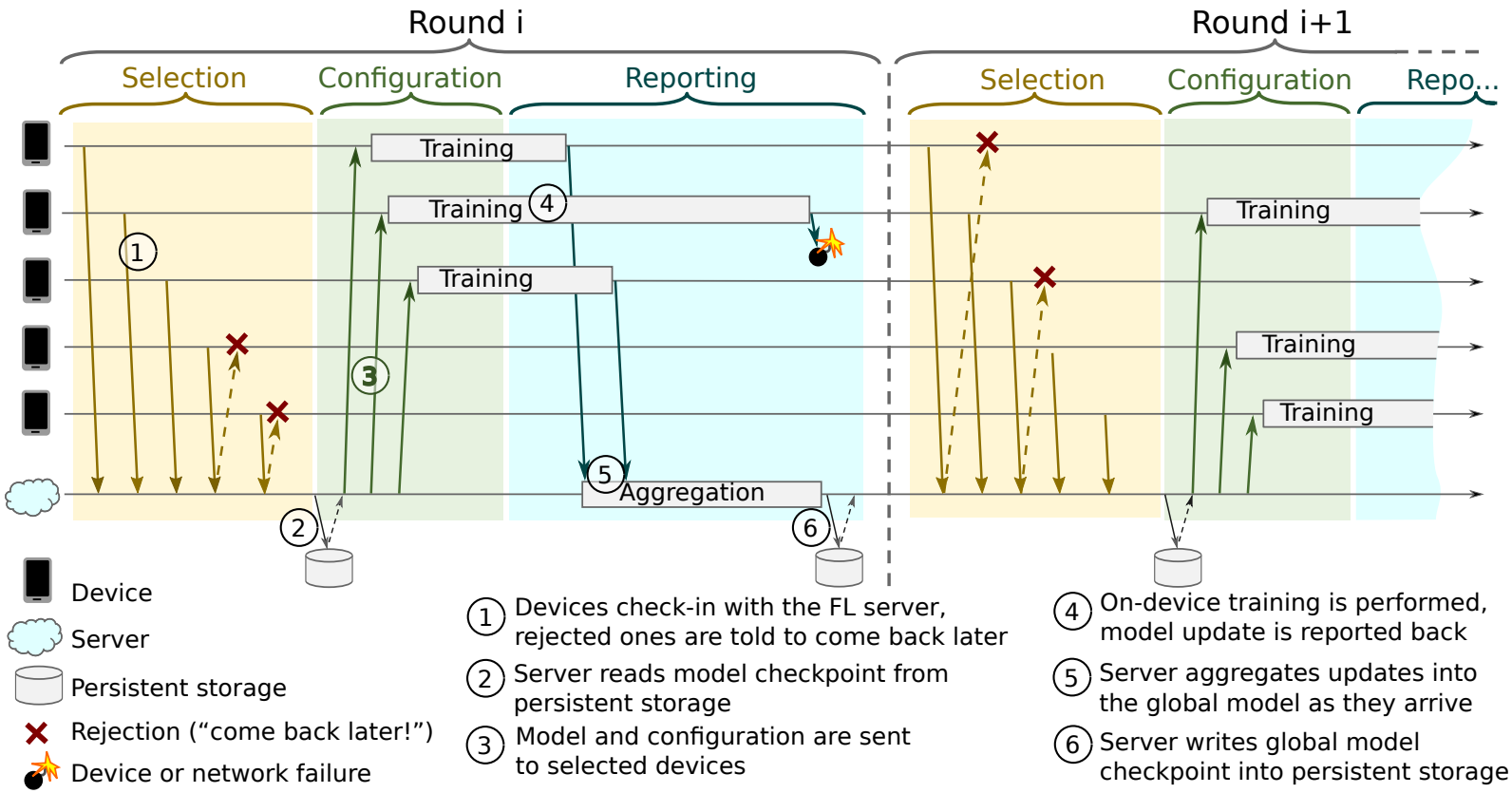

Figure 1: Federated Learning Protocol

図 3: Federated Learning のプロトコル (再掲) [?]

Federated Learning のプロトコル

- Configuration フェーズの流れ
  1. 選択されたデバイスに対して、FL Plan を送信
     ⇐ FL Plan は、TensorFlow の計算グラフや、FL Task の実行方法を格納
  2. 続いて、選択されたデバイスに対して、FL Checkpoint を送信
     ⇐ FL Checkpoint は、モデルのパラメータや、FL Task の実行に必要な様々な情報を格納
     ⇐ シリアライズ化された TensorFlow のセッションオブジェクトなど
  3. デバイスは、FL Server から渡された情報を元に、FL Task を実行
     ⇐ デバイス上に保存されたデータを用いた、モデルの訓練や評価

Federated Learning のプロトコル

- Reporting フェーズの流れ
  1. FL Server は、デバイスから結果が送信されるのを待機
     ⇐ FL Task がモデルの訓練であれば、差分データ (モデルのパラメータの更新情報) の送信を待機
  2. FL Server は、Federated Averaging を使って、差分データを一つに集約
     ⇐ 集約された差分を、モデルのパラメータに反映 (モデルの更新)
  3. FL Server は、タスクを実行し終えたデバイスに対して、次にアクセスすべき時刻を送信
     ⇐ 適当な時間の経過後に、デバイスが FL Server に再接続するように指示

  - 十分な数のデバイスが結果を報告すれば、モデルの更新が実行される
  - それ以外の場合は、Round の実行は失敗 (無かったことにされる)

Federated Learning のプロトコル

- デバイスの FL Server への接続頻度の調節
  - FL Population(解こうとしている問題) の大きさに応じて、デバイスの接続頻度 (一度に FL Server に接続してくるデバイス数) を調節
  - FL Server は、次に再接続すべき時間を、各デバイスに対して指示する
    ⇒ この時間をうまく調節することで、デバイスの接続頻度を調節可能
  - FL Task に協力可能 (アクティブ) なデバイスの数は、周期的に変動
    ⇒ 充電中で、かつ Wi-Fi に接続されていればアクティブとみなす
    ⇒ 昼の時間帯は人間が使用するので、アクティブなデバイスが減少
    ⇒ それ以外の時間帯 (深夜) では、逆にアクティブなデバイスが増加
  - これらの周期的な変動も考慮して、再接続までの時間を指定
    ⇒ 例えば、昼間の時間帯は、FL Server にアクセスする頻度を落とす

Federated Learning のプロトコル

- 小さな FL Population の場合 (解こうとしている問題が小さい)
  - FL Task に参加するデバイスの数も少なくて済む
  - 十分な数のデバイスが、FL Server にほとんど同時に接続できるように、接続頻度を上手く調節する
    ⇒ Selection フェーズに掛かる時間が短縮
    ⇒ 単位時間に実行可能な Round の数が増加
    ⇒ 学習が速やかに進行する

Federated Learning のプロトコル

- 大きな FL Population の場合 (解こうとしている問題が大きい)
  - 一般的に、FL Task に協力してくれるデバイスが多数存在する
  - 但し、一度の Round に参加するデバイスは、せいぜい数百程度である

  - 多数のデバイスが、一度に FL Server に接続しないようにする
    ⇒ 一度に多数のデバイスが FL Server に接続しても、そのうちのごく一部のデバイスが選択され、他の多数のデバイスの接続が無駄になるかもしれない (Thundering Herd と呼ばれる問題)

  - 各デバイスが、FL Server にアクセスする時間を、ランダムに決める
    ⇒ デバイスの接続を時間的に分散させる
    ⇒ 必要なときに、必要な数のデバイスだけが接続する

目次

デバイス上のソフトウェア

- デバイス上のソフトウェア
  - 今回のシステムは、Android のスマートフォンが対象
  - 但し、それ以外のプラットフォームでも実装可能

  - アプリケーションプロセス、FL Runtime、Example Store の 3 つが連携して動作
  - 次の図**??**を参照

デバイス上のソフトウェアのアーキテクチャ

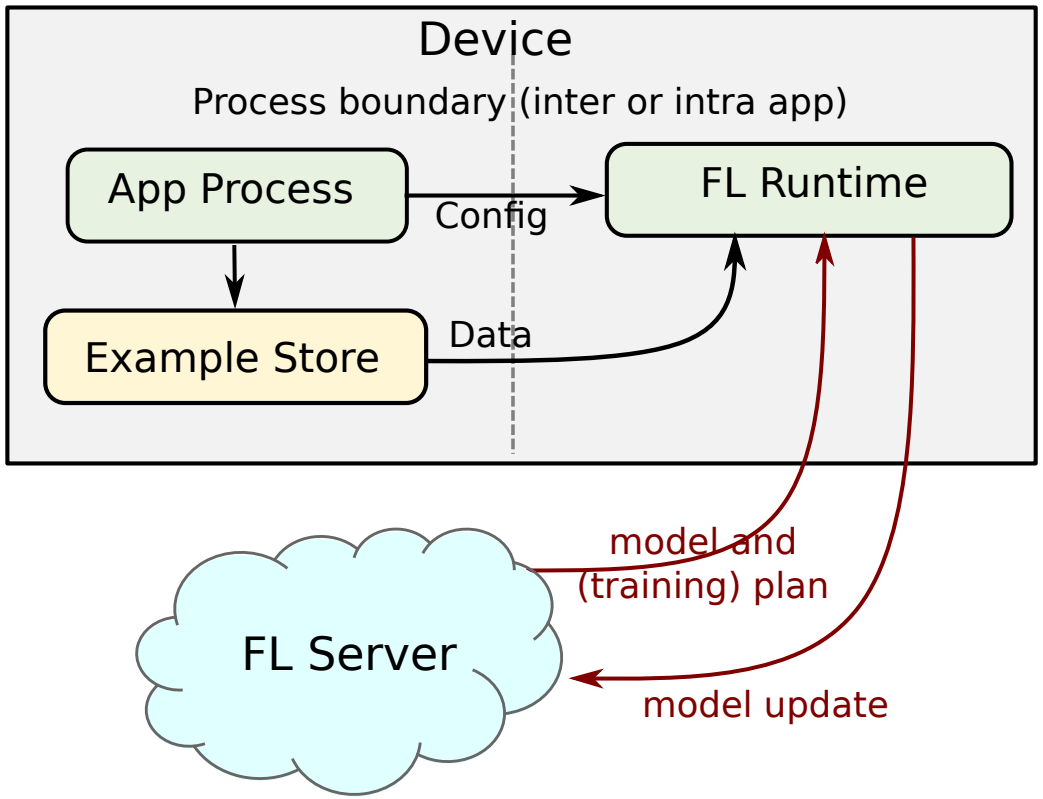

Figure 2: Device Architecture

図 4: デバイス上のソフトウェアのアーキテクチャ [?]

デバイス上のソフトウェア

- Example Store
  - Federated Learning で使用するデータを保存しておくデータベース
  - クラウド上にあるモデルの学習と、改良に用いられる訓練データ

  - 以下のような推奨事項がある

1. ストレージを圧迫しないように、データベースの最大容量を、予め決めておく
2. 各データの保存期間を決めておき、期間を過ぎたデータが自動的に削除されるようにする
3. マルウェアによる不正アクセスを防止するため、各データを適切に暗号化しておく

デバイス上のソフトウェア

- FL Runtime
  - FL Server とのやり取りを行うソフトウェアのコンポーネント
  - FL Task を実行する際、Example Store からデータを取得
  - 取得したデータを用いて、FL Task で指定された処理 (モデルの訓練や評価) を実行する

参考文献

[1] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloé Kiddon, Jakub Konecný, Stefano Mazzocchi, H. Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander.
Towards federated learning at scale: System design.
CoRR, abs/1902.01046, 2019.

[2] Brendan McMahan and Daniel Ramage.
Federated learning: Collaborative machine learning without centralized training data, Apr 2017.