PROBLEMS OF PRIVACY AND HETEROGENEITY FOR FEDERATED LEARNING APPLICATIONS IN MEDICAL IMAGE ANALYSIS

Зеленковский Виктор Петрович[a], *Гимбицкий Алексей Владимирович*[a], *Жидович Максим Сергеевич*[a], *Ковалёв Василий Алексеевич*[b]

[a]*Белорусский государственный университет, Факультет прикладной математики и информатики, Беларусь, Минск*
[b]*Объединённый институт проблем информатики Национальной академии наук Беларуси (ОИПИ НАН Беларуси), лаборатория анализа биомедицинских изображений, Беларусь, Минск*
**e-mail: alekseygimbickiy@gmail.com, victor.zelenkovsky@gmail.com, maksim.zhydovich@yandex.by, vassili.kovalev@gmail.com*

Abstract В последнее время использование машинного обучения стало одним из самых перспективных направлений при работе с медицинскими данными. Наиболее эффективными и точными являются глубокие нейросетевые модели, но для их обучения требуются большие вычислительные мощности и объемы информации. Часто необходимые данные невозможно собрать в одном месте по причине их приватности. Одним из способов решения данной проблемы является использование алгоритмов федеративного обучения. В данной статье будут рассмотрены основные проблемы, возникающие при использовании данных алгоритмов в задаче анализа биомедицинских изображений: неоднородность моделей и данных, а также будет предложен метод, позволяющий их избежать.

Keywords: федеративное обучение, неоднородность, приватность, медицинские изображения.

1. ВВЕДЕНИЕ

Идея федеративного обучения возникла и была впервые описана в 2016 году исследователями из компании Google, которые предложили этот подход в статье «Communication-Efficient Learning of Deep Networks from Decentralized Data»[1]. Данный метод относится к распределенному процессу машинного обучения, развернутому на нескольких клиентах, то есть позволяет обучать на сервере единую глобальную модель, используя несколько источников (клиентов), каждый из которых обучается на своём собственном наборе данных. Более формально, пусть в процессе построение глобальной статистической модели участвуют N клиентов $\{C_1, C_2, \ldots, C_N\}$ каждый со своим собственным набором данных $\{D_1, D_2, \ldots, D_N\}$. Координирует работу разных клиентов и их обучение сервер.

Процесс федеративного обучения можно условно разделить на три ключевых этапа:

1. Инициализация. На каждом шаге t клиенты загружают себе последнюю версию модели w^t с сервера.

2. Локальное обучение. Каждый клиент C_k выполняет итеративное обучение на основе своего собственного локального набора данных D_k и гиперпараметра η, обновляет веса локальной модели после нескольких эпох обучения $w_k^t \leftarrow w_k^{t-1}$ (η, D_k) и отправляет их на сервер.

3. Агрегация моделей. Сервер выполняет агрегацию весов, полученных от локальных моделей, и обновляет глобальную модель.

$$w_{global}^t \leftarrow Aggregation(w_k^t;\ k \in [1,\ 2,\ \ldots,\ N])\ . \tag{1.1}$$

Целью всего процесса является минимизация целевой функции, которая может быть записана следующей формулой:

$$min_w \sum_{k=1}^{N} p_k F_k(w)\ . \tag{1.2}$$

, где F_k - локальная целевая функция для k-ого клиента, p_k - величина, отражающая относительное влияние каждого клиента, причем $p_k > 0$ и $\sum_{k=1}^{N} p_k = 1$. То есть на каждом шаге каждый клиент обновляет веса своей модели, а далее сервер агрегирует k наборов весов с помощью определённого метода агрегации (агрегация усреднением, прогрессивная агрегация Фурье, FedGKT и др.)

Более подробно весь процесс федеративного обучения показан на рисунке ниже:

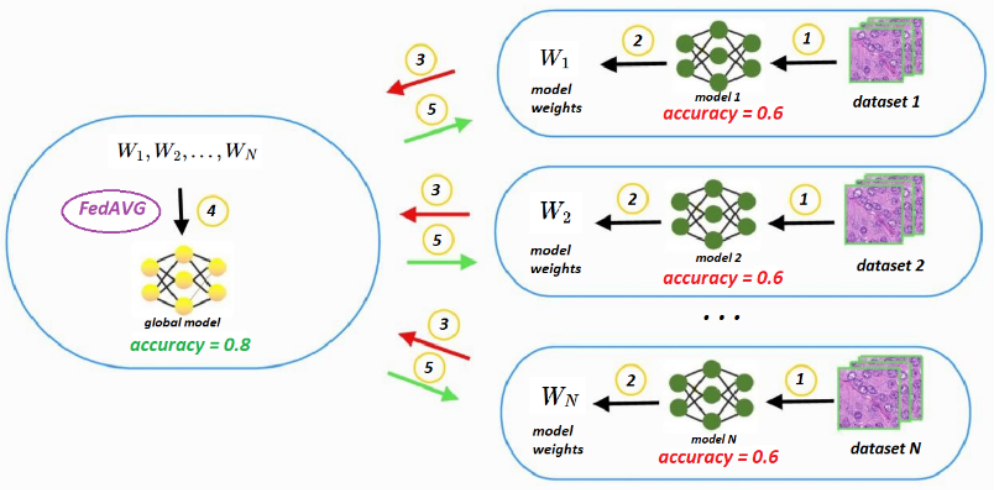

Рисунок 1 - Схема процесса федеративного обучения

Позже такая стратегия была названа централизованным федеративным обучением, а также была предложена децентрализованная стратегия федеративного обучения. При

стратегии децентрализованного федеративного обучения. не требуется использования центрального сервера, с которым бы обменивались данными модели-клиенты. Вместо этого каждый клиент по отдельности обменивается с некоторыми другими клиентами и агрегирует их обновления. Такая стратегия позволяет решить проблему узкого места, то есть вся модель не выйдет из строя из-за точечных ошибок.

К основными преимуществам федеративного обучения можно отнести следующие[2]:

- масштабируемость: распределенная функция федеративного обучения позволяет системе легко адаптироваться к любым изменениям в количестве участвующих устройств;
- упрощение модели: позволяя различным совместным устройствам проводить несколько параллельных циклов обучения с использованием небольших объемов данных, федеративное обучение упрощает традиционный централизованный подход, при котором один объект должен каждый раз обрабатывать внушительный объем данных;
- быстрая сходимость: используя более простые модели, устройства, участвующие в федеративном обучении, быстрее выполняют несколько итераций, поскольку они учатся на опыте других устройств, что приводит к более быстрому построению надежной глобальной модели.

Но, несмотря на перечисленные выше преимущества, методы федеративного обучения ограничены при их использовании в случае неоднородности данных и локальных моделей. Анализу и решению именно этих проблем посвящена данная статья.

2. ПРОБЛЕМА НЕОДНОРОДНОСТИ В ФЕДЕРАТИВНОМ ОБУЧЕНИИ

Существующие методы федеративного обучения имеют следующие **недостатки**, которые ограничивают их использование для медицинских изображений:

- потеря приватности данных: на этапе агрегации сервер получает все веса моделей от клиентов, что приводит к потере конфиденциальности данных. Имея все веса и информацию о гиперпараметрах модели, становится возможным с некоторой точностью восстановить изображения, на которых обучались локальные модели;
- трудность агрегации моделей различной архитектуры: в случае, когда архитектуры локальных моделей сильно отличаются, количество и размерности матриц весов сильно отличаются, что делает невозможным применение базовых методов федеративного обучения.
- трудность агрегации при неоднородных данных: часто данные на клиентах могут быть по-разному предобработанны. Более того, данные могут отличаться своим балансом классов, что также затрудняет использование базовых методов федеративного обучения.

Для решения данных проблем предлагается использовать метод, отличающийся от других подходов, тем, что на этапе агрегации сервер получает не все веса модели, а лишь их определенную часть. Данная особенность позволяет сохранить приватность локальной модели, а, следовательно, и данных, на которых она обучалась.

Агрегация весов происходит все так же на сервере, но теперь сервер - это модель машинного обучения (в частности, нейронная сеть), обучаемая на схожую задачу и на данных, схожих с теми, на которых обучаются локальные модели.

Алгоритм агрегации на основе части весов состоит из следующих этапов:

1. Клиенты отправляют серверу определенную часть (не все) обучаемых весов $\{w_k^t,\ k \in [1, 2, \ldots, N]\}$.
2. Сервер с помощью преобразования F: $R_k \to R_h$ переводит полученные веса w_k^t из пространства R_k (пространство весов k-ого клиента (веса, передаваемые клиентами, могут быть различной размерности)) в пространство R_h, скрытое пространство. Это делается для дальнейшего удобства агрегирования весов, находящихся в едином пространстве.

$$w_k^t \leftarrow F(w_k^t)\ . \tag{2.1}$$

3. Далее с помощью преобразования G: $R_h \to R_h$ сервер агрегирует веса, полученные на предыдущем шаге, следующим образом:

$$w_{global} \leftarrow G(w_k^t)\ . \tag{2.1}$$

4. Затем для всех клиентов сервер, применяя обратное преобразование F^{-1}: $R_h \to R_k$, переводит агрегированные веса w_{global} в исходное пространство весов клиента R_k и отправляет их обратно клиенту.

Описанный метод обладает теми же плюсами, что и базовые подходы федеративного обучени, однако имеется еще одно ключевое преимущество, которого не было у других методов: **сохранение приватности изображений**.

Утечка данных не происходит в силу того, что клиенты обмениваются с сервером лишь частью весов, на основе которых невозможно восстановить исходные данные, на которых обучалась модель.

Именно данное преимущество позволяет использовать данный алгоритм для обучения моделей, используемых для медицинских изображений.

Среди **недостатков** данного подхода можно выделить следующие:

- необходимость дополнительно обучать модель сервера, что требует, во-первых, дополнительного времени, а во-вторых, дополнительных данных для обучения;
- отсутствие улучшения качества моделей в случае, когда количество весов, передаваемых клиентами для агрегации, слишком мало;
- потенциальное ухудшение качества в случае, когда архитектуры моделей слишком сильно отличаются.

3.ЭКСПЕРИМЕНТЫ И РЕЗУЛЬТАТЫ

Для анализа эффективности разработанного метода использовался набор 12000 (8400 - тренировочный, 3600 - тестовый) гистологических изображений, разбитых на 2 класса: со злокачественными опухолями и без них. Тренировочный набор был случайным образом разбит на 5 частей: 4 клиента, 1 сервер.

Круг обучения(отправка весов от клиентов к серверу, их агрегация и отправка обратно) производилась каждые 3 эпохи обучения. Количество таких кругов равно 10. Веса для агрегации отправлялись от каждого клиента. Таким образом, итоговое количество эпох обучения для каждого клиента равно 30.

Веса моделей для обмена (простравнство R_k) - матрица весов линейного слоя классификационной части сети размерности 1024x1024 (для всех клиентов).

Оценка метода производилась путем сравнения следующих значений с применением данного метода во время обучения и без него(традиционное обучение без агрегации весов 30 эпох):

- значение функции потерь(кросс-энтропия) во время обучения;
- значение функции потерь на тестовой выборке;
- точность предсказания на тестовой выборке.

Модели одинаковой архитектуры.

Для анализа эффективности разработанного метода в случае однородности локальных моделей использовалась следующая архитектура простой нейронной сети:

```
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1         [-1, 16, 222, 222]             448
         MaxPool2d-2           [-1, 16, 74, 74]               0
            Conv2d-3           [-1, 16, 72, 72]           2,320
            Conv2d-4           [-1, 32, 70, 70]           4,640
         MaxPool2d-5           [-1, 32, 23, 23]               0
           Flatten-6                [-1, 16928]               0
            Linear-7                 [-1, 1024]      17,335,296
            Linear-8                 [-1, 1024]       1,049,600
            Linear-9                    [-1, 2]           2,050
================================================================
Total params: 18,394,354
Trainable params: 18,394,354
Non-trainable params: 0
```

Рисунок 2 - Архитектура простой нейронной сети

К качестве глобальной модели(сервер) использовалась предобученная сеть resnet18, дообученая на 1170 гистологических изображениях 30 эпох. Для агрегации весов с клиентов последний слой, предназначенный для классификации, был изменен на следующий:

```
(fc): Sequential(
  (fc): Linear(in_features=512, out_features=1048576, bias=True)
  (reshape): Reshape()
  (aggregate): Sequential(
    (0): Linear(in_features=1024, out_features=1024, bias=True)
    (1): ReLU()
  )
  (max_pool): MaxPool2d(kernel_size=(1024, 1), stride=(1024, 1), padding=0, dilation=1, ceil_mode=False)
  (squeeze): Squeeze()
  (classifier): Linear(in_features=1024, out_features=2, bias=True)
)
```

Рисунок 3 - Последний слой сети resnet18 для агрегации

Для агрегации весов используется линейный слой aggregate, на вход которому подаются веса от клиентов для агрегации, а выходные значения которого - модифицированные веса каждого клиента.

Ниже приведен график оценки различных метрик качества для клиента 0 (графики для остальных клиентов аналогичны).

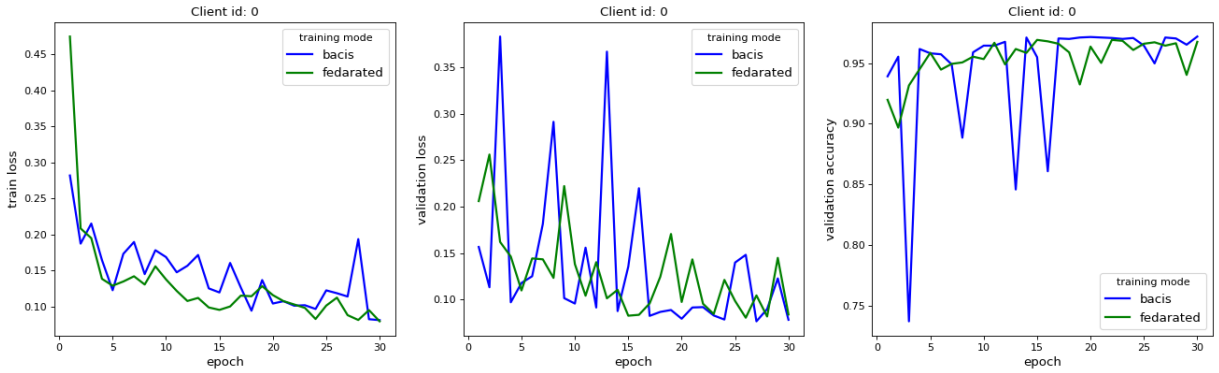

Рисунок 4 - Графики анализируемых метрик(SimpleModel)

Проанализировав данные графики, можно сделать следующие выводы о работе данного метода на простых моделях одинаковой архитектуры:

- для всех клиентов наблюдается снижение и стабильность значений функции потерь во время обучения в случае применения метода;
- для 3/4 клиентов наблюдается более высокая стабильность точности предсказаний на тестовом наборе данных в случае применения метода;
- точность предсказания с применением метода в среднем не изменилась.

Модели различной архитектуры.

Для анализа эффективности в случае неоднородности локальных моделей использовались следующие предобученные нейронные сети:

- клиент 1: SimpleModel(см. выше);
- клиент 2: mobilenetv3 large;
- клиент 3: mobilenetv3 small;
- клиент 4: densenet121.

Для каждой локальной модели последний слой нейросети был заменен на следующий:

```
(classifier): Sequential(
  (fc): Linear(in_features=1024, out_features=1024, bias=True)
  (relu1): ReLU()
  (shared): Linear(in_features=1024, out_features=1024, bias=True)
  (relu2): ReLU()
  (classifier): Linear(in_features=1024, out_features=2, bias=True)
```

Рисунок 4 - Последний слой локальных моделей

Передаваемые веса - веса линейного слоя shared. Модель для сервера аналогична модели из предыдущего раздела.

Ниже приведены графики оценки различных метрик качества для клиентов 0 и 2.

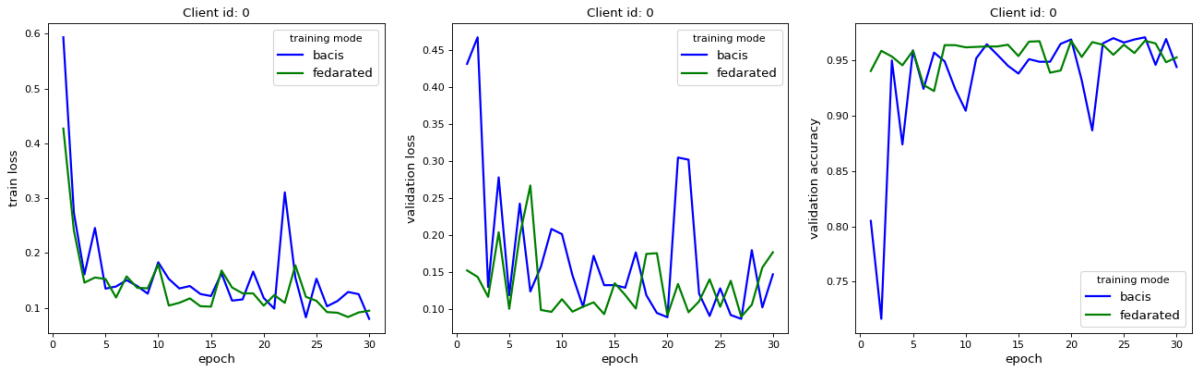

Рисунок 5 - Графики анализируемых метрик(SimpleModel)

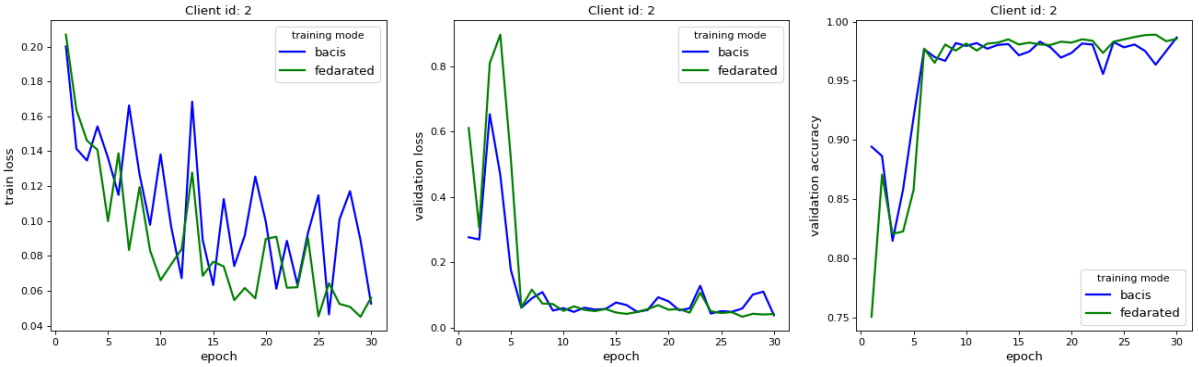

Рисунок 6 - Графики анализируемых метрик(mobilenetv3 large)

Проанализировав данные графики, можно сделать следующие выводы о работе данного метода на предобученных моделях различной архитектуры:

- для всех клиентов наблюдается снижение и стабильность значений функции потерь во время обучения в случае применения метода;
- наблюдается более высокая стабильность точности предсказаний на тестовом наборе данных для более простых моделей и некоторых предобученных сетей в случае применения метода;
- точность предсказания с применением метода в среднем не изменилась.

Неоднородные данные.

Для анализа эффективности в случае неоднородности данных рассмотрен случай следующего деления исходного набора данных на два клиента. В данных первого клиента содержится 95% объектов класса 0 и 5% класса 1. Для второго клиента пропорция обратная: 5% объектов класса 0 и 95% класса 1.

В подобной ситуации невозможно представить стабильное обучение и хорошие результаты при использовании последовательного обучения сначала на одном клиенте, а затем на другом. Однако в случае использования методов федеративного обучения нам удалось добиться таких же значений метрики, как и в случае сбалансированных данных. Связано это, в первую очередь с тем фактом, что при агрегации глобальная модель стремится получить знания со всех клиентов и усреднить их.

Таким образом, с помощью методов федеративного обучения удается избежать влияния неоднородности данных при обучении моделей.

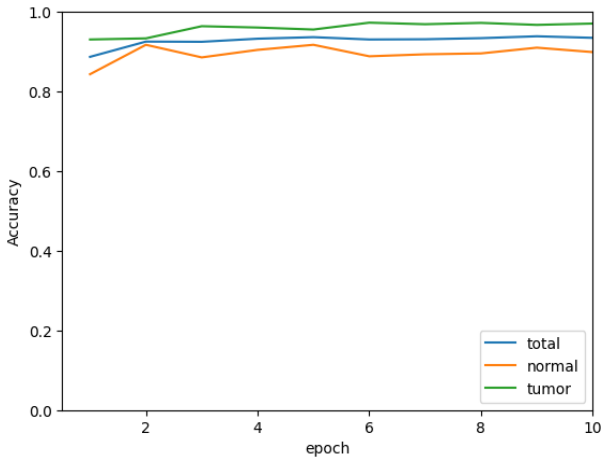

Рисунок 7 - График accuracy для неоднородных данных

5. ЗАКЛЮЧЕНИЕ

В данной статье были описаны основные проблемы, возникающие в задаче анализа биомедицинских изображений, а также предложен метод, позволяющий их избежать. Далее были описаны эксперименты, доказывающие возможность его применения на практике. Полученные выводы и рекомендации могут быть полезными для исследователей в области биомедицинской информатики и медицины, стремящихся использовать передовые методы машинного обучения для анализа изображений в условиях приватности данных.

ФИНАНСИРОВАНИЕ

Это исследование было поддержано Объединённым институтом проблем информатики Национальной академии наук Беларуси (ОИПИ НАН Беларуси).

КОНФЛИКТ ИНТЕРЕСОВ

Авторы заявляют об отсутствии конфликта интересов.

REFERENCES

1. H. Brendan McMahan, E. Moore, D. Ramage, S.Hampson, B. Agüera y Arcas, "Communication-efficient learning of deep networks from decentralized data," Artificial Intelligence and Statistics, 1273–1282 (2017).

2. P. Qi, D. Chiaro, A. Guzzo, M. Ianni, G. Fortino, F. Piccialli "Model aggregation techniques in federated learning: A comprehensive survey". Future Generation Computer System, 150, 272-293 (2023).