МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
Кафедра биомедицинской информатики

ИССЛЕДОВАНИЕ МЕТОДОВ АГРЕГАЦИИ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ С СОХРАНЕНИЕМ ПРИВАТНОСТИ МЕДИЦИНСКИХ ИЗОБРАЖЕНИЙ

Курсовой проект

Зеленковского Виктора Петровича
студента 4 курса,
специальность «информатика»

Научный руководитель:
доцент кафедры БМИ
Ковалёв В.А.

Минск, 2023

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

В последнее время машинное обучение стало одним из наиболее перспективных направлений при работе с медицинскими данными. Наиболее эффективными и точными являются глубокие нейросетевые модели, но для их обучения требуются большие объемы информации. Это является частой проблемой в случае медицинских данных, особенно изображений, так как их создание требует больших затрат.

Одним из решений, позволяющим улучшить качество глубоких нейросетевых моделей без увеличения обучающего набора данных, является агрегация моделей. Но возникает проблема с сохранением конфиденциальности медицинских изображений. Например, если одна модель обучается на изображении, содержащем информацию о конкретном пациенте, то другие модели, участвующие в агрегировании, могут также получить доступ к этой информации. В результате, информация о конкретном пациенте может быть раскрыта.

Пытаясь решить проблему, описанную выше, данная работа направлена на исследование и разработку методов агрегации моделей машинного обучения с сохранением приватности медицинских изображений, в частности методов федеративного обучения.

ГЛАВА 1

ОСНОВЫ АГРЕГАЦИИ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ

1.1 Предпосылки к агрегации моделей

В современном мире глубокие нейронные сети являются одним из наиболее мощных инструментов для анализа медицинских данных, поскольку они способны извлекать сложные взаимосвязи между различными признаками.

Но одной из основных проблем в обучении больших и сверхбольших нейронных сетей является вычислительные ограничения при обучении. Это происходит из-за того, что для обучения глубоких нейронных сетей используются алгоритмы обратного распространения ошибки, которые требуют большого количества итераций для достижения хорошего качества обучения. А с увеличением размера нейронной сети, количество итераций, необходимых для обучения, увеличивается экспоненциально.

Таким образом, обучение больших нейронных сетей становится весьма сложным и требует больших вычислительных ресурсов.

Одним из способов решения данной проблемы является агрегация.

Агрегация моделей машинного обучения – это процесс объединения нескольких моделей в одну более мощную и эффективную.

1.2 Преимущества и недостатки агрегации

Среди **преимуществ** агрегации моделей можно выделить следующие:

- повышение точности модели: объединение нескольких моделей позволяет получить более полную картину данных и снизить вероятность ошибок, связанных с недостатками одной модели;
- снижение риска переобучения: при агрегации моделей происходит объединение различных подходов к обучению, что позволяет снизить риск переобучения одной модели на конкретном наборе данных;
- улучшение устойчивости модели: агрегированные модели более устойчивы к изменениям входных данных, так как они учитывают различные точки зрения на проблему.

Однако у данного подхода обучения существуют некоторые **недостатки**:

- увеличение времени обучения: объединение моделей может потребовать больше времени на обучение, так как необходимо, во-первых, учитывать особенности каждой модели, а во-вторых дополнительное время для агрегации;

- сложность настройки параметров: при агрегации моделей необходимо учитывать множество различных параметров и их взаимосвязи, что может усложнить процесс настройки модели;

- риск утечки данных: так как при агрегации моделей используются данные от разных источников, существует риск утечки конфиденциальной информации.

1.3 Другие способы улучшения качества модели

Классические методы агрегации, такие как bagging, boosting и stacking, действительно позволяют улучшить качество без особого увеличения вычислительных затрат. Но это не совсем верно в случае глубоких нейронных сетей, для которых количество обучаемых параметров слишком велико. Возникает проблема обучения одной глубокой нейронной сети на одном устройстве, не говоря уже об агрегации нескольких обученных моделей.

Традиционный подход для решения данной проблемы - это использование центрального сервера (локального или облачного) с большими вычислительными ресурсами, на котором размещается и/или обучается модель.

Основным недостатком данной архитектуры является то, что все данные, собранные локальными устройствами, отправляются на центральный сервер для обработки, а затем возвращаются обратно на устройства. Данный способ обмена информацией ограничивает способность модели учиться в режиме реального времени и увеличивает риск утечки конфиденциальной информации.

Именно последний недостаток является ключевым при рассмотрении возможности использовать традиционный подход обучения моделей машинного обучения, используемых для медицинских изображений.

ГЛАВА 2

АГРЕГАЦИИ МОДЕЛЕЙ НА ОСНОВЕ ФЕДЕРАТИВНОГО ОБУЧЕНИЯ

2.1 Определение и основные понятия федеративного обучения

Для поддержания конфиденциальности данных во время обучения в 2017 году компанией Google был предложен подход к обучению моделей машинного обучения, называемый федеративное обучение[1].

Данный метод позволяет обучать на сервере единую глобальную модель, используя несколько источников (клиентов) с различными наборами данных без их передачи между источниками. То есть федеративное обучение относится к распределенному процессу машинного обучения, развернутому на нескольких клиентах.

Пусть в процессе построение глобальной статистической модели участвуют N клиентов $\{C_1, C_2, ..., C_N\}$ каждый со своим собственным набором данных $\{D_1, D_2, ..., D_N\}$. Набор данных для каждого клиента хранится локально, и данные не могут быть переданы между клиентами или собраны третьей стороной.

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 ${w_k}^t \leftarrow {w_k}^{t-1}(\eta, D_k)$ и отправляет их на сервер.

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

$$w^t_{global} \leftarrow Aggregation({w_k}^t; k \in [1, 2, ..., N]) \tag{2.1}$$

Целью всего процесса является минимизация целевой функции, которая отражает желаемые характеристики модели и может быть записана следующей формулой:

$$\min_w \sum_{k=1}^{N} p_k F_k(w) \tag{2.2}$$

, где F_k - локальная целевая функция для k-ого клиента, p_k - величина, отражающая относительное влияние каждого клиента, причем $p_k \geqslant 0$ и $\sum_{k=1}^{N} p_k = 1$.

То есть на шаге локального обучения каждый k-ый клиент обновляет веса своей модели в соответствии с локальной целевой функцией, а далее отправляет их серверу для обновления глобальной модели с помощью определенных методов агрегации.

Более подробно весь процесс федеративного обучения отображен на рисунке ниже:

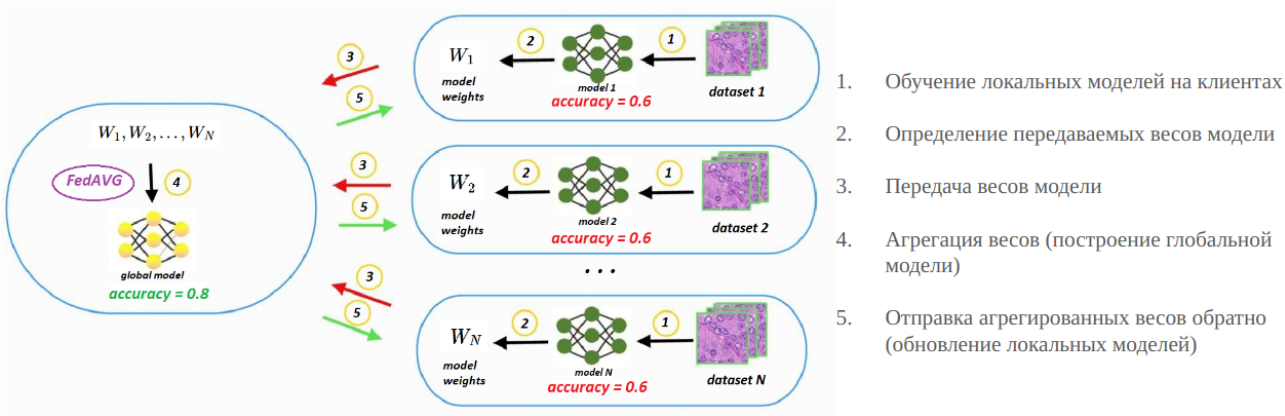


Рисунок 2.1 --- Процесс федеративного обучения

Таким образом, федеративное обучение позволяет нескольким клиентам совместно работать над обучением одной модели, не обмениваясь своими данными, что особенно важно для сохранения приватности медицинских изображений.

2.2 Основные методы федеративного обучения

После локального обучения модели клиентов объединяются с использованием определенного алгоритма агрегации. В литературе описано множество таких методов. Ниже перечислены наиболее известные и часто используемые из подходов для задач компьютерного зрения:

- агрегация усреднением;
- прогрессивная агрегация Фурье;
- FedGKT.

Рассмотрим несколько методов, основанных на агрегация усреднением.

2.2.1 FedAvg

Данный метод[2] является одним из самых ранних и наиболее часто используемых методов федеративного обучения.

Идея подхода заключается в следующем: на этапе агрегирования случайным образом выбирается группа клиентов, далее для получения глобальной модели

параметры каждого клиента усредняются с весовым коэффициентом, равным доле объема данных данного клиента.

$$w_{global}^{t} \leftarrow \sum_{C_k \in S_t} \frac{n_k}{n} {w_k}^t \tag{2.3}$$

, где S_t - группа клиентов выбранных на круге агрегации t, n_k - объем данным k-ого клиента, n - объем данных всех клиентов.

При этом количество локальных обновлений клиентов перед агрегацией является настраиваемым параметром.

Данный подход не позволяет клиентам выполнять различные объемы локальной работы в зависимости от вычислительных возможностей каждого клиента и расходится в случае, когда данные распределяются между клиентами неодинаково.

2.2.2 FedProx

Данный метод[3] является усовершенствованием предыдущего и направлен на смягчение проблемы локальной оптимизации, присущей подходам, основанным на стохастическом градиентном спуске.

Данная проблема заключается в следующем: выполнение многочисленных локальных итеративных шагов обучения в FedAvg может привести к тому, что каждый клиент будет уделять приоритетное внимание достижению своей локальной цели, а не глобальной, что приведет к неоптимальной сходимости или расхождению моделей.

Решение, предлагаемое данным методом, заключается в добавлении к целевой функции слагаемого $\frac{\mu}{2}||{w_k}^t - w_{global}^t||^2$ для регулирования влияния локальных моделей и обеспечения гарантий сходимости. При $\mu = 0$ FedProx аналогичен FedAvg, то есть последующая агрегация модели и глобальное обновление выполняются по тому же принципу, что и в FedAvg.

2.2.3 FedNova

В алгоритме FedNova[4] изменен этап агрегации моделей алгоритма FedAvg с целью устранения неидентичности моделей. Алгоритм перед обновлением глобальной модели использует нормализацию и масштабирование локальных обновлений от каждого клиента на основе его локального номера итерации.

2.3 Преимущества и недостатки существующих методов

К основными **преимуществам** федеративного обучения можно отнести следующие[5]:

- 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 у рассмотренных методов имеются следующие существенные **недостатки**, ограничивающие их использование для медицинских изображений:

- потеря приватности данных: на этапе агрегации сервер получает все веса моделей от клиентов, что приводит к потере конфиденциальности данных. Это обосновывается тем фактом, что, 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.

ГЛАВА 3

АГРЕГАЦИИ МОДЕЛЕЙ С СОХРАНЕНИЕМ ПРИВАТНОСТИ ИЗОБРАЖЕНИЙ

3.1 Постановка задачи

Ниже будет использоваться терминология из главы 2.

Данная глава посвящена методам, разработанным для решения проблем, ограничивающим использование базовых и наиболее известных подходов федеративного обучения для медицинских изображений.

Основной целью описанных ниже алгоритмов является улучшение качества моделей машинного обучения путем их агрегации с сохранением приватности медицинских изображений, использованных во время обучения.

3.2 Агрегирование с помощью части весов

3.2.1 Описание метода

Данный метод отличается от других подходов, описанных в главе 2, 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 состоит из следующих этапов:

1. Клиенты отправляют серверу определенную часть (не все) обучаемых весов $\{w_k^t, k \in [1, 2, ..., N]\}$.

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, находящегося в едином пространстве.

$$w^{t'}{}_k \leftarrow F(w_k^t) \tag{3.1}$$

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

$$w_{global}^t \leftarrow G(w^{t'}{}_k) \tag{3.2}$$

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

Преобразования F, G, F^{-1} выше обучаются моделью сервером.

Таким образом, во время агрегации веса клиента C_k преобразовываются следующим образом:

$$w^t_k \to F^{-1}(G(F(w^t_k))) \tag{3.3}$$

3.2.2 Преимущества и недостатки

Описанный метод обладает теми же преимуществами, что и базовые подходы федеративного обучения, о которых говорилось в разделе 2.3:

- масштабируемость;
- упрощение модели.

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** 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.

ГЛАВА 4
ЭКСПЕРИМЕНТЫ И РЕЗУЛЬТАТЫ

В данной главе представлены и анализированы результаты проверки работы метода, разработанного в разделе 3.2.

Решаемая задача: классификация изображений.

Набор исходных данных: 8400 гистологических изображений, разбитых на 2 класса: изображения с раковыми клетками и без них.

Размер исходных изображений: 256 × 256 пикселей.

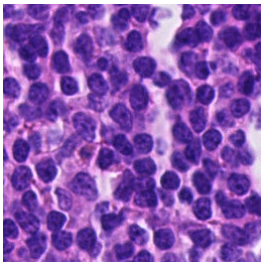


Рисунок 4.1 --- Пример исходных данных 1-ого класса

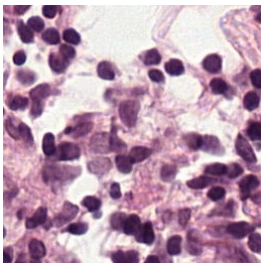


Рисунок 4.2 --- Пример исходных данных 2-ого класса

Количество клиентов: 4.

Размер набора данных для обучения для каждого клиента и сервера: 1176.

Размер тестового набора данных: 2520.

Круг обучения(отправка весов от клиентов к серверу, их агрегация и отправка обратно) будет производится каждые 3 эпохи обучения. Количество таких кругов равно 10.

Веса для агрегации будут отправляться от каждого клиента.

Таким образом итоговое количество эпох обучения для каждого клиента равно 30.

Веса моделей для обмена: матрица весов линейного слоя сети размерности 1024x1024 (для всех клиентов).

Оценка метода будет производится 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.

4.1 Модели одинаковой архитектуры

Модель для всех клиентов - простая нейронная сеть со следующей архитектурой:

```
        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 4.3 --- Архитектура нейросети SimpleModel

Модель для сервера - предобученная сеть resnet18, дообученая на 1176 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 4.4 --- Последний измененный слой сети resnet18

Для агрегации весов используется линейный слой aggregate, выходные значения которого - модифицированные веса каждого клиента.

Графики, позволяющие сделать следующие выводы о работе данного метода на простых моделях одинаковой архитектуры, находятся в Приложении А:

- 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.

4.2 Модели различной архитектуры

Далее для клиентов будут использоваться следующие предобученные нейросети:

Клиент 1: SimpleModel(см. предыдущий раздел);
Клиент 2: mobilenetv3 large;
Клиент 3: mobilenetv3 small;
Клиент 4: densenet121.

Для каждого клиента последний слой нейросети 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 4.5 --- Измененный последний слой моделей клиентов

Передаваемые веса - веса линейного слоя shared.

Модель для сервера аналогична модели из предыдущего раздела.

Графики, позволяющие сделать следующие выводы о работе данного метода на предобученных моделях различной архитектуры, находятся в Приложении В:

- 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 в случае применения метода;
- точность предсказания с применением метода в среднем не изменилась.

ЗАКЛЮЧЕНИЕ

В ходе работы были изучены базовые и наиболее часто применяющиеся методы федеративного обучения: FedAvg, FedProx, FedNova. Рассмотрены их преимущества и недостатки, на основе чего был сделан вывод о невозможности использования данных методов в условиях приватности медицинских изображений.

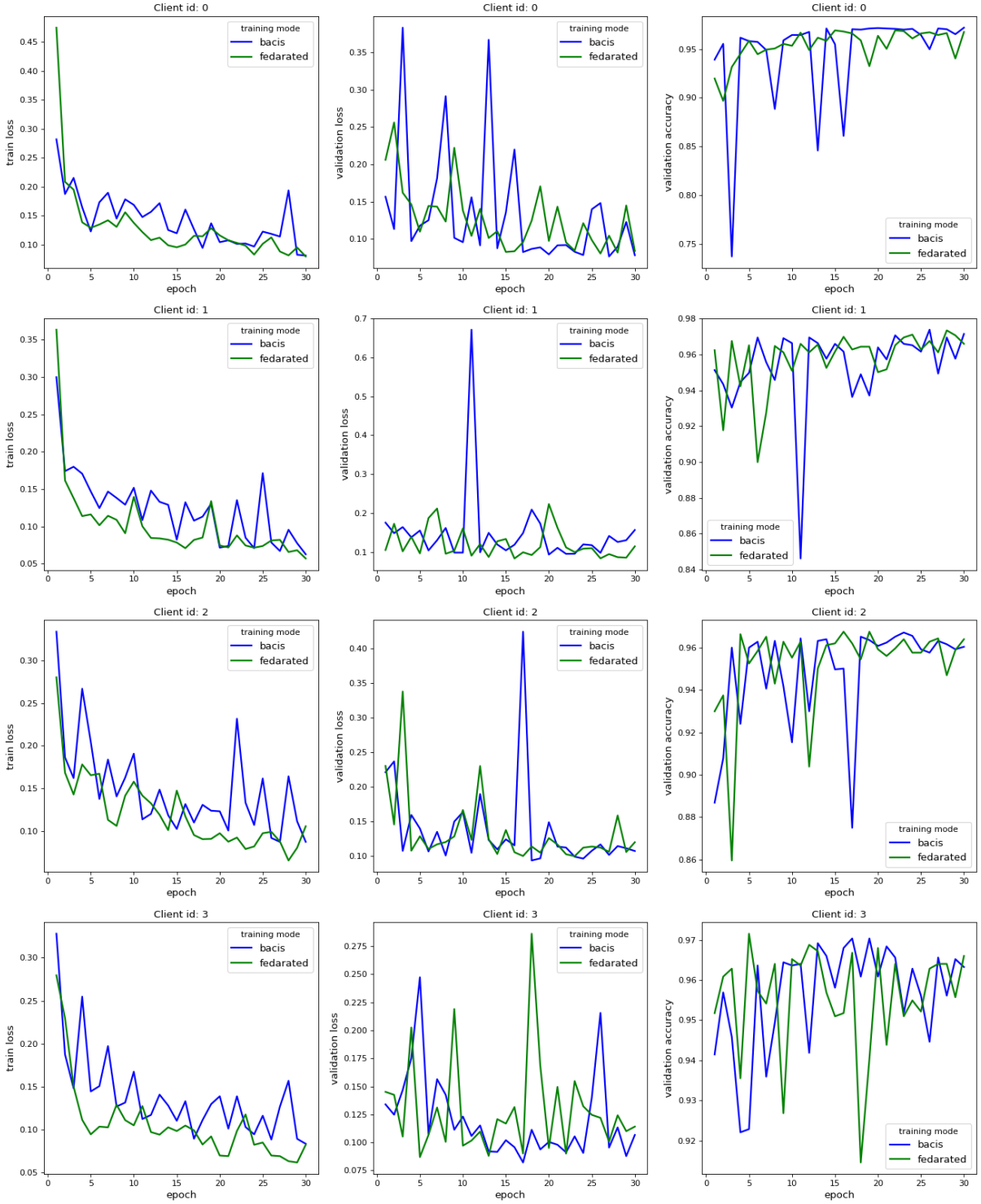
Далее был разработан метод агрегации моделей машинного обучения с сохранением конфиденциальности набора данных, используемого при обучении. Рассмотрены его потенциальные преимущества и недостатки.

Разработан алгоритм, реализующий работу данного метода для моделей одинаковой и различной архитектуры.

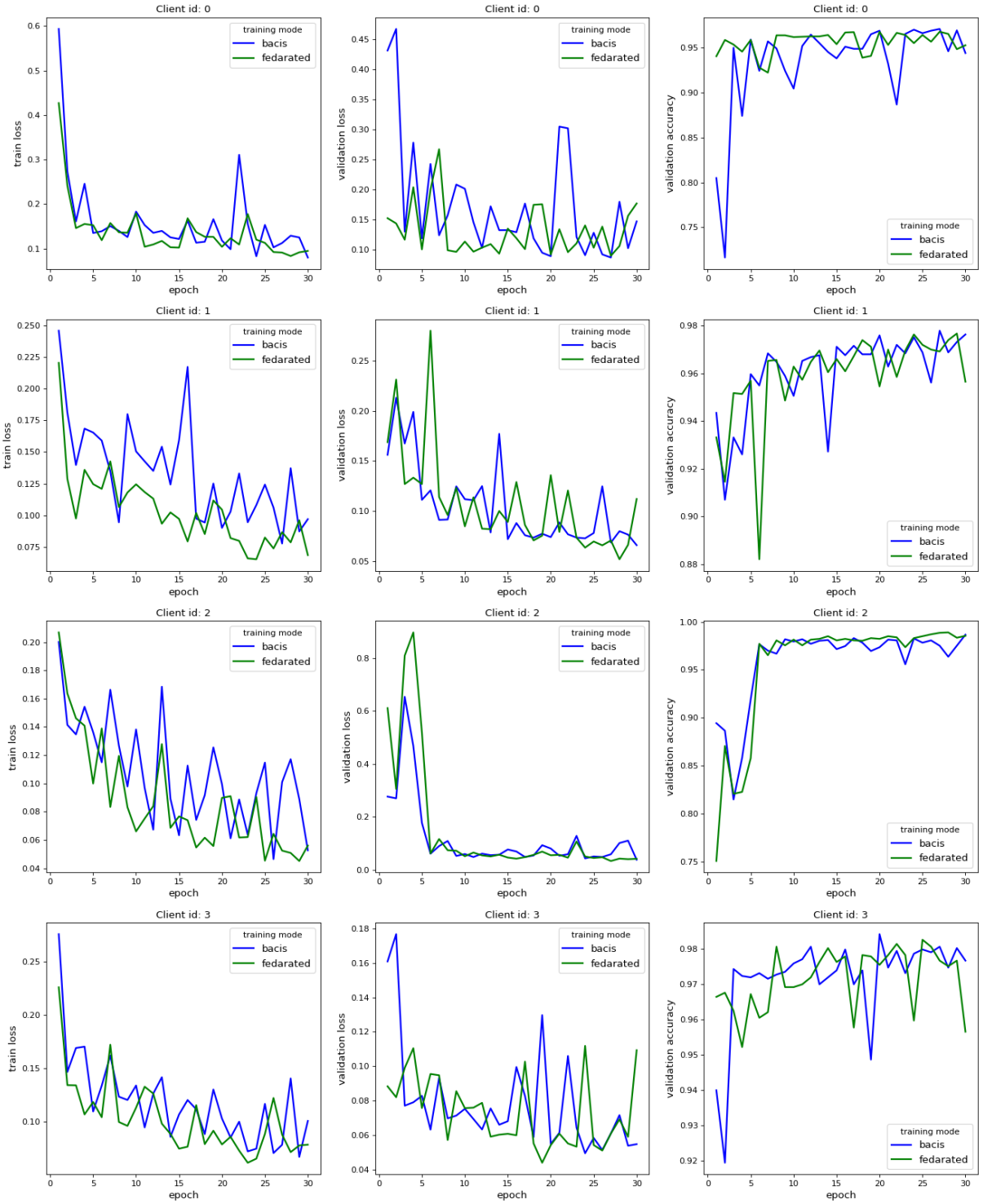
Получены необходимые результаты, проанализировав которые, был сделан вывод, что разработанный метод позволяет улучшить стабильность обучения моделей машинного обучения.

Логическим продолжением выполненной работы является разработка схожих методов, основанных на агрегации лишь определенной части обучаемых весов.

Графики сравнения качества моделей с одинаковыми архитектурами



Графики сравнения качества моделей с различными архитектурами



СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Federated Learning Collaborative Machine learning without Centralized Training Data / V.A. Abhishek[et al.] // International Journal of Engineering Technology and Management Sciences. - 2022. - Vol. 6. - P. 255-259.

2. Communication-efficient learning of deep networks from decentralized data / B. McMahan [et al.] // Artificial Intelligence and Statistics. - 2017. - P. 1273–1282.

3. Federated optimization in heterogeneous networks / T. Li [et al.] // Proceedings of Machine Learning and Systems - 2020. - Vol. 2. - P. 429–450.

4. Tackling the objective inconsistency problem in heterogeneous federated optimization / J. Wang [et al.] // NeurIPS - 2020. - Vol. 33. - P. 7611–7623.

5. Model aggregation techniques in federated learning: A comprehensive survey / Pian Qi [et al.] // Future Generation Computer System - 2023. - Vol. 150. - 272-293.