

УДК 519.68
DOI: 10.15827/0236-235X.137.037-044

Дата подачи статьи: 16.10.21, после доработки: 25.10.21
2022. Т. 35. № 1. С. 037–044

Формальная модель многоагентных систем для федеративного обучения

Г.П. Юлейси¹, аспирант, yuleisy2688@gmail.ru
И.И. Холод¹, д.т.н., доцент, iiholod@mail.ru

¹ Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»
им. В.И. Ульянова (Ленина), г. Санкт-Петербург, 197022, Россия

В статье представлена формальная модель многоагентных систем для федеративного обучения. Концепция федеративного обучения очень близка к многоагентным системам, поскольку агенты позволяют обучать модели машинного обучения на локальных устройствах, сохраняя при этом конфиденциальную информацию. Возможности агентов взаимодействовать друг с другом позволяют обобщать (агрегировать) такие модели и повторно их использовать.

В работе описываются взаимодействие и координация агентов, которые должны осуществляться с учетом стратегий обучения: последовательно, когда модель обучается по очереди на каждом узле; централизованно, когда модели обучаются параллельно на каждом узле и агрегируются на центральном сервере; децентрализованно, когда обучение и агрегация выполняются на каждом из узлов.

Выделены основные типы агентов, необходимые для выполнения полного цикла федеративного обучения: принимающий задачу от пользователя, собирающий информацию о среде, выполняющий планирование обучения, выполняющий обучение на узле с данными, предоставляющий информацию и доступ к данным, осуществляющий агрегацию моделей. Для каждого из агентов определены основные действия и типы сообщений, которыми они обмениваются. Проанализированы и описаны конфигурации размещения агентов для каждой из стратегий федеративного обучения.

На основе предложенной формальной модели можно осуществлять разработку многоагентных систем, используемых для задач федеративного обучения, а на основе выделенных типов агентов и видов сообщений – платформы агентов, сами агенты и протоколы их взаимодействия.

Ключевые слова: агент, федеративное обучение, коммуникация, мультиагентные системы, модель.

Основными проблемами, возникающими при проектировании *многоагентных систем* (МАС), представляющих собой распределенные сетевые приложения, являются размещение ее элементов на сетевых узлах и организация взаимодействия между ними. От этого во многом зависят такие характеристики системы, как занимаемые на узлах ресурсы, время выполнения и сетевой трафик операций. Для МАС эти проблемы усложняются возможным динамическим изменением местоположения агентов, между которыми необходимо обеспечивать взаимодействие.

В свою очередь, при анализе данных, находящихся на распределенных узлах, приходится сталкиваться с такими проблемами, как приватность данных, ограничение пропускной способности каналов связи, обработка больших объемов информации и др. Для их решения была предложена концепция федеративного обучения (Federated Learning – FL) [1–3], которая предусматривает обучение без передачи данных от источников. Выделяют центра-

лизованные и децентрализованные топологии FL, в которых возможны разные стратегии обучения: циклическое, федеративное и роевое.

Для реализации такого обучения хорошо подходят МАС. Состав агентов, их размещение и взаимодействие могут зависеть от топологии и стратегии FL. МАС предоставляют возможность адаптации агентов с учетом этих условий.

Концепции МАС и FL имеют определенные формальные модели, каждая из которых описывает ее элементы, ресурсы и то, как она связана с окружающей средой. В данной статье описываются формальная модель МАС для FL и ее адаптация для реализации разных стратегий.

Формальная модель МАС

Формальное определение МАС можно представить в виде выражения, описывающего каждый из составляющих его элементов [4, 5]: $MAS = \{A^t, E^t\}$, где $A^t = \{a_1^t, a_2^t, \dots, a_n^t\}$ – мно-

жество всех агентов в MAC на момент времени t ; $E^t = \{N^t, R^t, ORG^t, ACT^t, COM^t\}$ – отношение агентов с окружающей средой на момент времени t ; $N^t = \{n_1^t, n_2^t, \dots, n_j^t\}$ – множество всех узлов в MAC на момент времени t ; $R_{AN}^t = A^t \times N^t$ – отношение, описывающее размещение агентов множества A на узлах множества N , в момент времени t ; ORG^t – множество типов агентов в момент времени t (состав зависит от прикладной задачи); $ACT^t = \{ACT_1^t, ACT_2^t, \dots, ACT_x^t\}$ – конечное множество действий, выполняемых различными типами агентов в MAC, в момент времени t (состав зависит от прикладной задачи); COM^t – коммуникативные действия, выполняемые между агентами, узлами и средой через ACL или протокол связи (состав зависит от прикладной задачи).

Формальная модель FL

В случае размещения данных $D = \{d_1, \dots, d_z\}$ на разных узлах множества N для их анализа применяют два подхода [6, 7]:

- распределенный анализ, предполагающий сбор всех наборов данных $d = d_1 \cup \dots \cup d_z$ вместе и использование объединенного набора d для обучения модели m_{DL} алгоритмом α : $m_{DL} = \alpha(d)$;

- FL, предполагающее построение модели m_{FL} алгоритмом агрегации agg из моделей m_1, \dots, m_z , обученных алгоритмом α на данных d_1, \dots, d_z , хранящихся на узлах без передачи их куда-либо (рис. 1): $m_{FL} = agg(m_1, m_2, \dots, m_z)$, где $m_1 = \alpha(d_1), m_2 = \alpha(d_2), \dots, m_z = \alpha(d_z)$.

Так, FL эффективно, если точность m_{FL} , обозначенная как $A(m_{FL})$, очень близка к точности m_{DL} , обозначенной как $A(m_{DL})$: $|A(m_{FL}) - A(m_{DL})| < \delta$, где δ – неотрицательное действительное число, обозначающее приемлемую потерю точности.

Таким образом, формально систему FL можно описать с помощью двух множеств:

- множества клиентов, выполняющих алгоритм α на данных d_i и строящих модель $m_i = \alpha(d_i)$: $C = \{c_1, \dots, c_z\}$;

- множества серверов, выполняющих агрегацию agg моделей $agg(m_1, m_2, \dots, m_z)$: $S = \{s_1, \dots, s_y\}$.

В зависимости от состава серверов, используемых для решения конкретной задачи, выделяют две топологии FL: централизованная (включает только один сервер: $S = \{s\}$) и децентрализованная (предполагает серверы на каждом узле с данными: $S = \{s_1, \dots, s_z\}, |S|=|D|$).

Стратегия FL – это план обучения, который определяет, как модель обучается на клиентах и

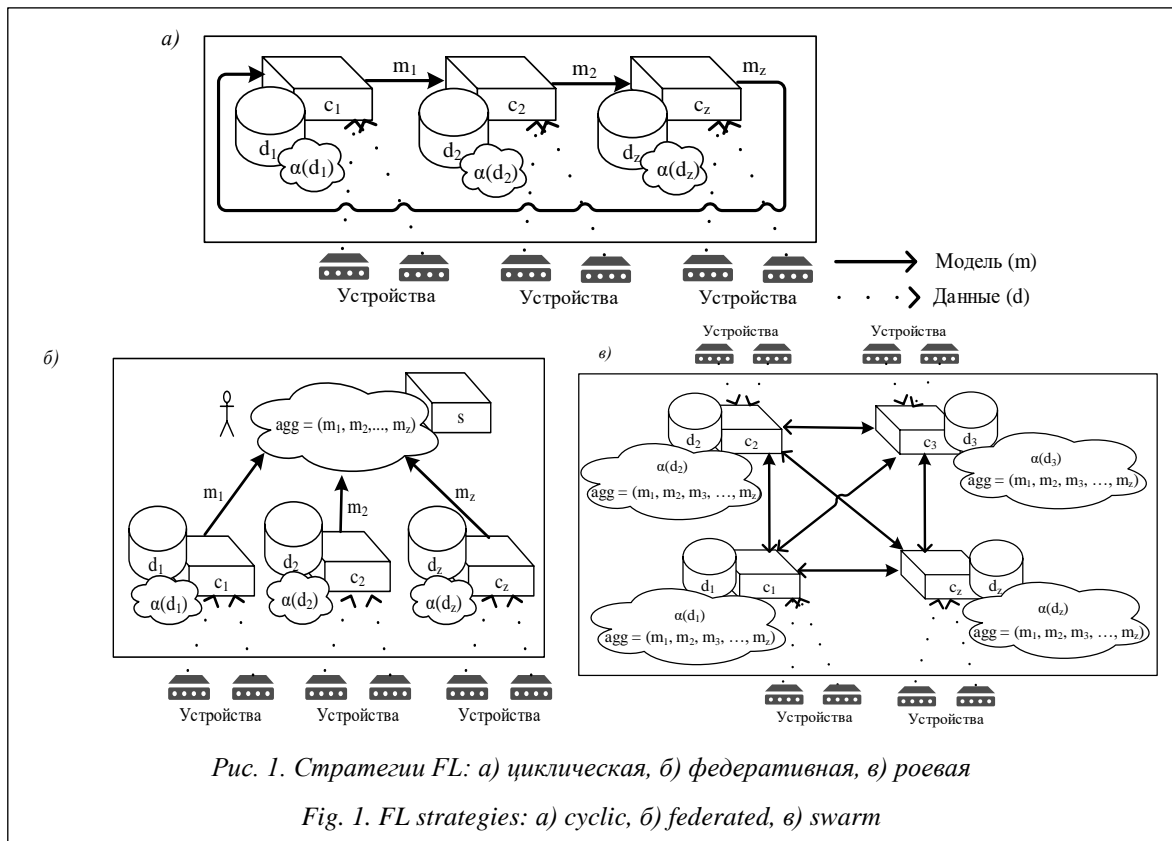


Рис. 1. Стратегии FL: а) циклическая, б) федеративная, в) роевая

Fig. 1. FL strategies: а) cyclic, б) federated, в) swarm

как организована связь между ними и/или сервером. В работе [8] выделены три типа стратегий: циклическая, федеративная, роевая (рис. 1). Циклическая модель обучается на каждом клиенте последовательно и передается следующему клиенту (в этом случае агрегирование модели не требуется). Федеративная модель обучается локально на каждом клиенте параллельно и агрегируется на сервере. Роевая модель обучается локально на каждом клиенте и распространяется по остальным клиентам, на каждом из которых выполняется агрегирование.

При выполнении *FL* выделяют 5 этапов [9].

Этап 1. Сбор информации о среде (опционально) – для пользователя собирается информация о данных, узлах, на которых они размещены, и т.п.

Этап 2. Постановка задачи – пользователь указывает, какие данные должны быть проанализированы, каким алгоритмом, с какими параметрами и т.п.

Этап 3. Рассылка модели (опционально, если обучение осуществляется параллельно на разных узлах) – передается модель на каждый из узлов с данными.

Этап 4. Обучение – на каждом из узлов выполняется обучение на локальных данных, на которых они размещаются.

Этап 5. Агрегация (опционально, если обучение осуществляется параллельно на разных узлах) – объединяются модели, обученные на каждом узле.

Формальная модель MAC для FL

Традиционно рассматривают два варианта применения MAC к машинному обучению [10]:

– параллельная работа нескольких агентов, когда агенты обучают локальные модели на данных и передают их агенту-мастеру для объединения;

– последовательная работа одного агента, когда он перемещается от источника к источнику, дообучая модель на каждом из узлов.

Стратегии обучения иллюстрирует рисунок 2.

Формально опишем MAC для реализации *FL* как систему агентов, каждый из которых выполняет обучение, обмениваясь моделями друг с другом, формируя общую модель.

Поведение каждого агента определяется выполнением различных задач в соответствии с определенной ролью. Среди основных задач *FL* можно выделить сбор информации о среде выполнения, обучение модели алгоритмом $\alpha(d)$, объединение моделей функцией агрегации $agg(m_1, m_2, \dots, m_z)$.

Уточним некоторые сущности в формальной модели MAC для *FL* [11].

Множество ролей:

$ORG = \{UA, FA, TA, DMA, DA, AA\}$, где

– агент пользователя (*UA* – User Agent) взаимодействует с пользователем и принимает от него задачу на обучение: $UA = \{ua\}$;

– агент-посредник (*FA* – Facilitator Agent) собирает информацию о среде выполнения: $FA = \{fa\}$;

– агент задачи (*TA* – Task Agent) координирует работу всех агентов для выполнения задачи пользователя: $TA = \{ta\}$;

– агенты интеллектуального анализа данных (*DMA* – Data Mining Agent) выполняют алгоритм обучения на данных: $DMA = \{dma_1, dma_2, \dots, dma_z\}$;

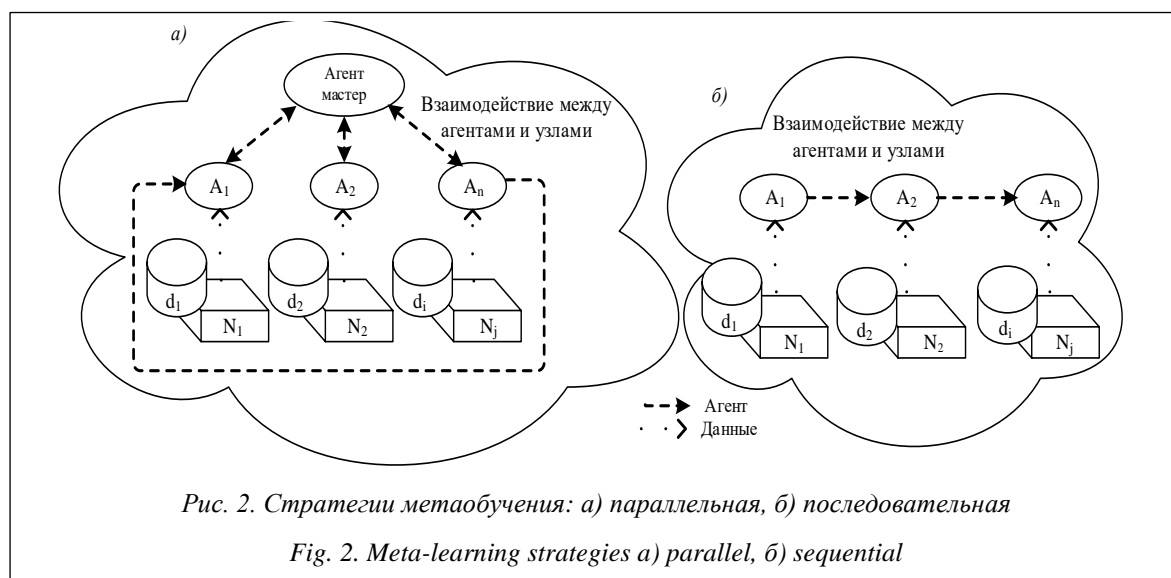


Рис. 2. Стратегии метаобучения: а) параллельная, б) последовательная

Fig. 2. Meta-learning strategies a) parallel, б) sequential

– агент данных (DA – Data Agent) взаимодействует с данными: $DA = \{da_1, da_2, \dots, da_z\}$;

– агент агрегации (AA – Aggregate Agent) объединяет модели: $AA = \{aa\}$.

• Конечное множество действий, выполняемых каждым типом агента, в момент времени t :

$ACT^t = \{ACT_{UA}^t, ACT_{FA}^t, ACT_{TA}^t, ACT_{DMA}^t, ACT_{DA}^t, ACT_{AA}^t\}$, где

$ACT_{UA}^t = \{r_query, c_task, s_task\}$ – действия агента UA , где r_query – получить запрос пользователя; c_task – интерпретировать запрос пользователя в задачу для агента; s_task – передать задачу агентам;

$ACT_{FA}^t = \{a_agent, req_info, r_info, s_info\}$ – действия агента FA , где a_agent – зарегистрировать агентов; req_info – запросить информацию о данных на каждом клиенте; r_info – получить информацию о данных с каждого клиента; s_info – передать информацию о данных на клиентах среды;

$ACT_{TA}^t = \{r_task, r_info, c_plan, s_plan\}$ – действия агента TA , где r_task – получить задачу пользователя; r_info – получить информацию о данных с каждого из узлов; c_plan – запланировать работу по выполнению обучения; s_plan – отправить план обучения;

$ACT_{DMA}^t = \{r_plan, req_data, c_a, s_models\}$ – действие агента DMA , где r_plan – получить план выполнения; req_data – запросить данные; c_a – выполнить алгоритм интеллектуального анализа данных, используя значения, содержащиеся в запросе; s_models – отправить обученную модель для агрегирования;

$ACT_{DA}^t = \{r_info, r_data, s_info, s_data\}$ – действия агента DA , где r_info – получить запрос на информацию о данных; r_data – получить запрос на доступ к данным; s_info – поддерживать информацию метаданных об источнике данных; s_data – передать доступ к данным;

$ACT_{AA}^t = \{r_model, a_models, s_models\}$ – действия, выполняемые агентами AA , где r_model – получить модель; a_models – агрегировать полученные модели; s_models – отправить агрегированные модели.

• Сообщения, которые перемещаются между агентами по протоколу ACL [11]:

$COM^t = \{task, qinfo, info, plan, qdata, data, model\}$, где $task$ – задача пользователя; $qinfo$ – запрос информации о клиенте; $info$ – информация о данных на клиенте; $plan$ – план выполнения задачи; $qdata$ – запрос на доступ к данным; $data$ – доступ к данным; $model$ – обученная модель.

Конфигурации MAC для стратегий FL

Выполнение FL начинается, когда UA получает задачу от пользователя, интерпретирует ее и передает ($task$) TA . Параллельно FA запрашивает информацию от DA о данных, доступных на узлах, и тоже передает эту информацию ($info$) TA . Таким образом, TA получает всю информацию, необходимую для выполнения FL : задачу пользователя и информацию о среде, в которой выполняется анализ. На основе этой информации TA генерирует план ($plan$) выполнения FL .

После этого в зависимости от используемой стратегии будут выполнены разные действия.

В циклической стратегии (рис. 3) сгенерированный план ($plan$) передается для выполнения DMA . Он, получив план, обращается к DA ($qdata$) и с его помощью получает доступ к данным ($data$). Используя эти данные, DMA обучает модель ($model$). На следующем узле эта модель дообучается.

Таким образом, не требуется агрегация моделей и, следовательно, множество типов агентов не будут включать в себя AA [11]:

$ORG^t_{seq} = \{UA, FA, TA, DMA, DA\}$.

Возможны два варианта циклической стратегии:

– использование одного мобильного агента dma , который будет перемещаться вместе с моделью, поэтому не требуется отправка модели:

$DMA^t_{seq_mob} = \{dma\}$,

$R^t_{ANseq_mob} = \{ \langle ua, fa, ta \rangle, n_1 \rangle, \langle dma, n^t \rangle, \dots, \langle da_1, n_1 \rangle, \dots, \langle da_z, n_z \rangle \} \subset A^t \times N^t$,

$COM^t_{seq_mob} = COM^t / \{model\}$,

$ACT^t_{seq_mob} = ACT^t / \{ACT_{AA}^t\}$,

$ACT_{DMA}^t = \{r_plan, req_data, c_a\}$ – действие агента DMA , где r_plan – получить план выполнения; req_data – запросить данные; c_a – выполнить алгоритм интеллектуального анализа данных, используя значения, содержащиеся в запросе;

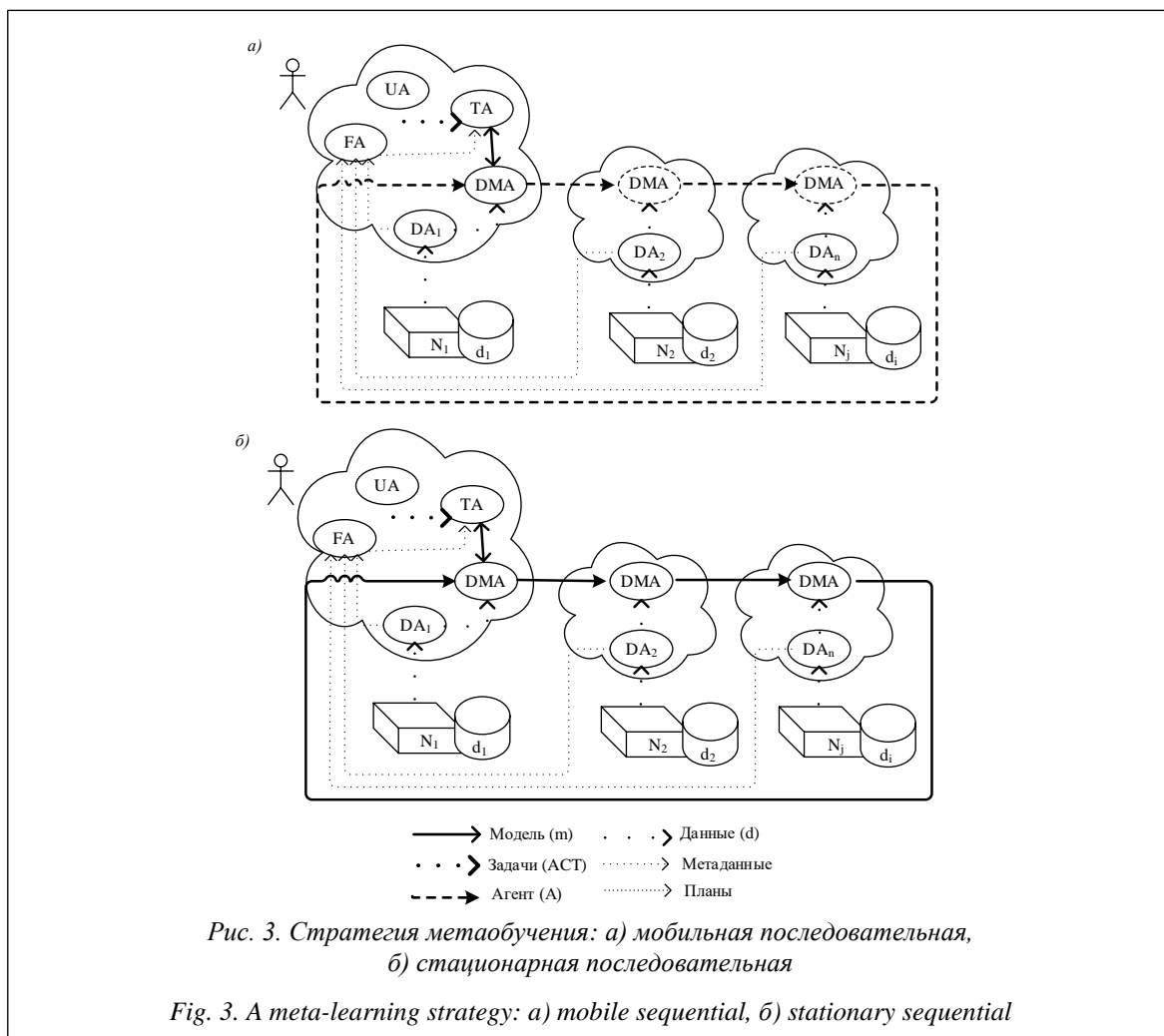
– использование стационарных агентов DMA на каждом из узлов, которые по очереди будут передавать модель друг другу:

$DMA^t_{seq_st} = \{dma_1, dma_2, \dots, dma_n\}$,

$R^t_{ANseq_st} = \{ \langle ua, fa, ta \rangle, n_s \rangle, \langle dma_1, n_1 \rangle, \dots, \langle dma_z, n_z \rangle, \langle da_1, n_1 \rangle, \dots, \langle da_z, n_z \rangle \} \subset A^t \times N^t$,

$COM^t_{seq_st} = COM^t, ACT^t_{seq_st} = ACT^t / \{ACT_{AA}^t\}$.

В случае с федеративной стратегией сгенерированный план ($plan$) передается для исполнения AA и DMA . Каждый DMA , получив план,

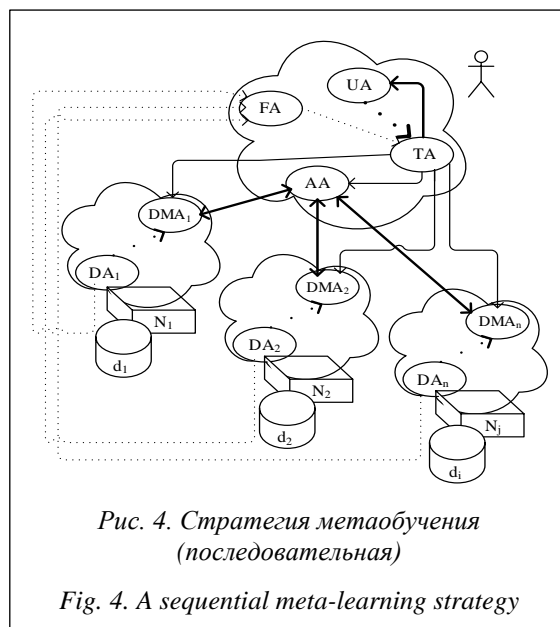


обращается к *DA* за данными (*qdata*) и с его помощью получает доступ к данным (*data*). На этих данных *DMA* обучает модель (*model*) и передает ее *AA*. После получения от *DMA* всех моделей *AA* объединяет их и возвращает *DMA*. Обмен между *AA* и *DMA* осуществляется до завершения обучения (рис. 4). Таким образом, элементы среды имеют следующий вид:

$$\begin{aligned}
 ORG_{fed}^t &= \{UA, FA, TA, DMA, AA, DA\}, \\
 DMA_{fed}^t &= \{dma_1, dma_2, \dots, dma_n\}, \\
 R_{ANfed}^t &= \{ \langle ua, fa, ta, aa \rangle, n_s, \langle dma_1, n_1 \rangle, \dots, \\
 &\langle dma_z, n_z \rangle, \langle da_1, n_1 \rangle, \dots, \langle da_z, n \rangle \} \subset A^t \times N^t, \\
 COM_{fed}^t &= COM^t, ACT_{fed}^t = ACT^t.
 \end{aligned}$$

В случае роевой стратегии на главном сервере (с которым взаимодействует пользователь) размещаются все типы агентов. Сгенерированный план (*plan*) передается для исполнения в *AA* и *DMA*, которые размещаются на каждом из узлов. Каждый *DMA*, получив план, обращается (*qdata*) к *DA* и с его помощью получает доступ к данным (*data*). На этих данных *DMA* обучает модель (*model*) и передает ее

всем *AA*. После получения моделей от всех *DMA* каждый *AA* объединяет их и возвращает



локальному DMA. Обмен между AA и DMA осуществляется до завершения обучения (рис. 5).

$$\begin{aligned}
 ORG^t_{swarm} &= \{UA, FA, TA, DMA, AA, DA\}, \\
 DMA^t_{swarm} &= \{dma_1, dma_2, \dots, dma_n\}, \\
 AA^t_{swarm} &= \{aa_1, aa_2, \dots, aa_n\}, \\
 R^t_{ANswarm} &= \{ \langle ua, fa, ta \rangle, n_s \rangle, \\
 &\langle aa_1, n_1 \rangle, \dots, \langle aa_z, n_z \rangle, \langle dma_1, n_1 \rangle, \dots, \\
 &\langle dma_z, n_z \rangle, \langle aa_1, n_1 \rangle, \dots, \langle aa_z, n_z \rangle \} \subset A^t \times N^t, \\
 COM^t_{swarm} &= COM^t, ACT^t_{swarm} = ACT^t.
 \end{aligned}$$

Заключение

Исследование позволило определить формальную модель MAC для FL. В модели выделены основные типы агентов, их действия и сообщения, которыми они обмениваются. Определение предложенных агентов позволило понять, как будут осуществляться коммуникация и координация в MAC для разных стратегий FL.

Анализ, проведенный на основе построенной формальной модели MAC_FL для разных стратегий FL, показал, что в зависимости от выбранной топологии и стратегии FL состав множеств модели MAC меняется несущественно. Для федеративной и роевой стратегий добавляется тип агента AA, выполняющий агрегацию моделей. При этом протокол взаимодействия для всех стратегий остается неизменным, за исключением последовательного обучения, при котором не требуется передача модели (model).

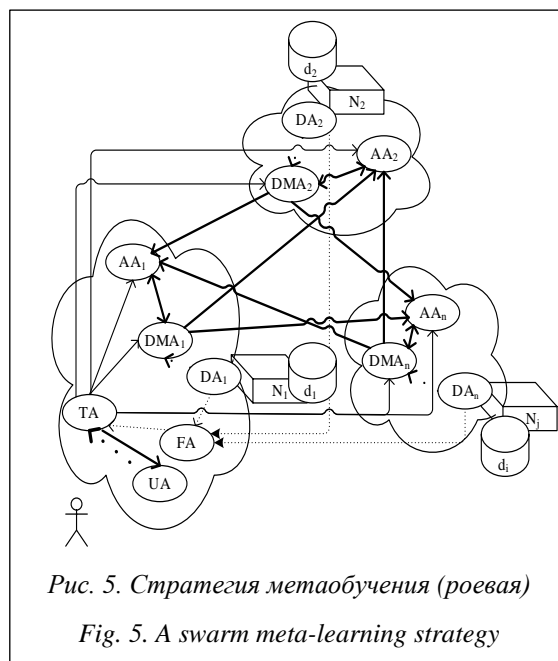


Рис. 5. Стратегия метаобучения (роевая)

Fig. 5. A swarm meta-learning strategy

Предложенная формальная модель может служить основой для разработки MAC, используемых для задач FL. На основе выделенных типов агентов и видов сообщений будут разработаны платформа агентов, сами агенты с определенными жизненными циклами, а также предложены протоколы их взаимодействия. Это позволит строить гибкие и масштабируемые системы, реализующие методы FL на основе MAC.

Литература

1. McMahan H.B., Moore E., Ramage D., Hampson S., Arcas B.A. Communication-efficient learning of deep networks from decentralized data. AISTATS, 2017, pp. 1273–1282.
2. Li L., Fan Y., Tse M., Lin K.Y. A review of applications in federated learning. Computers & Industrial Engineering, 2020, vol. 149, art. 106854. DOI: 10.1016/j.cie.2020.106854.
3. Li Q., Wen Z., Wu Z., Hu S., Wang N., Li Y., Liu X. et al. A Survey on federated learning systems: Vision, hype and reality for data privacy and protection. IEEE Transactions on Knowledge and Data Engineering, 2021, p. 1. DOI: 10.1109/TKDE.2021.3124599.
4. Яровой С.В. Имитационное моделирование распределенных динамических процессов на поверхности земли на основе агентного подхода: автореф. дис. Красноярск: СибГУ им. М.Ф. Решетнева, 2019. 23 с.
5. Душутина Е., Столяренко Ю.А. Формальная модель временного обязательства в многоагентных системах реального времени // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. 2013. № 2. С. 9–16.
6. Yang Q., Liu Y., Chen T., Tong Y. Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology, 2019, vol. 10, no. 2, pp. 1–19. DOI: 10.1145/3298981.
7. Kholod I., Yanaki E., Fomichev D., Shalugin E., Novikova E., Filippov E., Nordlund M. Open-source federated learning frameworks for IoT: A comparative review and analysis. Sensors, 2021, vol. 21, no. 1, art. 167. DOI: 10.3390/s21010167.
8. Rieke N., Hancox J., Li W., Milletari F., Roth H., Albarqouni Sh. et al. The future of digital health with federated learning. npj Digit. Med., 2020, vol. 3, art. 119. DOI: 10.1038/s41746-020-00323-1.
9. Kairouz P., McMahan B.H., Avent B. et al. Advances and open problems in federated learning. Foundations and Trends® in Machine Learning, 2021, vol. 14, no. 1-2, pp. 1–210. DOI: 10.1561/22000000083.

10. Юлейси П.Г., Холод И.И. Анализ многоагентных систем для анализа данных // XXIII Междунар. конф. по мягким вычислениям и измерениям. 2020. Т. 1. С. 273–276.

11. Yuleisy P.G., Kholod I.I. Protocol for the interaction of agents solving the problem of distributed data analysis. Proc. 2021 II Int. Conf. NeuroNT, 2021, pp. 47–50. DOI: 10.1109/NeuroNT53022.2021.9472793.

Software & Systems
DOI: 10.15827/0236-235X.137.037-044

Received 16.10.21, Revised 25.10.21
2022, vol. 35, no. 1, pp. 037–044

A formal model of multiagent systems for federated learning

G.P. Yuleisy¹, *Postgraduate Student, yuleisy2688@gmail.ru*

I.I. Kholod¹, *Ph.D. (Engineering), Associate Professor, iiholod@mail.ru*

¹ *St. Petersburg Electrotechnical University "LETI", St. Petersburg, 197022, Russian Federation*

Abstract. Recently, the concept of federated learning has been actively developing. This is due to the tightening of legislation in the field of working with personal data. Federated learning involves performing data training directly on the nodes where the data is stored. As a result, there is no need to transfer data anywhere, and they remain with the owners. To generalize the trained models, they are sent to the server that performs the aggregation.

The concept of federated learning is very close to a multi-agent system, since agents allow training machine learning models on local devices while maintaining confidential information. The ability of agents to interact with each other makes it possible to generalize (aggregate) such models and reuse them.

Taking into account the tasks that are solved by the federated learning methods, there are several learning strategies. Learning be carried out as follows: sequentially when the model is trained in turn at each node; centrally when models are trained in parallel at each node and aggregated on a central server; or decentralized where training and aggregation is performed on each of the nodes. Interaction and coordination of agents should be carried out taking into account these learning strategies.

This article presents a formal model of multi-agent systems for federated learning. It highlights the main types of agents required to complete the full cycle of federated learning: an agent that accepts a task from a user; an agent that collects information about the environment; an agent performing training planning; an agent performing training on a data node; an agent providing information and access to data; an agent performing model aggregation. For each of them, the paper defines the main actions and types of messages exchanged by such agents. It also analyzes and describes the configurations of agent placement for each of the federated learning strategies.

Keywords: agent, federated learning, communication, multi-agent systems, model.

References

1. McMahan H.B., Moore E., Ramage D., Hampson S., Arcas B.A. Communication-efficient learning of deep networks from decentralized data. *AISTATS*, 2017, pp. 1273–1282.
2. Li L., Fan Y., Tse M., Lin K.Y. A review of applications in federated learning. *Computers & Industrial Engineering*, 2020, vol. 149, art. 106854. DOI: 10.1016/j.cie.2020.106854.
3. Li Q., Wen Z., Wu Z., Hu S., Wang N., Li Y., Liu X. et al. A Survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering*, 2021, p. 1. DOI: 10.1109/TKDE.2021.3124599.
4. Yarovoy S.V. *Simulation Modeling of Distributed Dynamic Processes on the Earth's Surface Based on the Agent-Based Approach. Ph.D. Thes.*, Krasnoyarsk, Reshetnev Siberian State University of Science and Technology, 2019, 23 p. (in Russ.).
5. Dushutina E., Stolyarenko Yu.A. Formal model of time commitment in multi-agent real-time systems. *St. Petersburg Polytech. Univ. J. Computer Science. Telecommunication and Control Systems*, 2013, vol. 2, pp. 9–16 (in Russ.).
6. Yang Q., Liu Y., Chen T., Tong Y. Federated machine learning: Concept and applications. *ACM Trans. on Intelligent Systems and Technology*, 2019, vol. 10, no. 2, pp. 1–19. DOI: 10.1145/3298981.

7. Kholod I., Yanaki E., Fomichev D., Shalugin E., Novikova E., Filippov E., Nordlund M. Open-source federated learning frameworks for IoT: A comparative review and analysis. *Sensors*, 2021, vol. 21, no. 1, art. 167. DOI: 10.3390/s21010167.
8. Rieke N., Hancox J., Li W., Milletari F., Roth H., Albarqouni Sh. et al. The future of digital health with federated learning. *npj Digit. Med.*, 2020, vol. 3, art. 119. DOI: 10.1038/s41746-020-00323-1.
9. Kairouz P., McMahan B.H., Avent B. et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 2021, vol. 14, no. 1-2, pp. 1–210. DOI: 10.1561/22000000083.
10. Yuleisy P.G., Kholod I.I. Analysis of multiagent system for data analysis. *Proc. XXIII Int. Conf. SCM*, 2020, vol. 1, pp. 273–276 (in Russ.).
11. Yuleisy P.G., Kholod I.I. Protocol for the interaction of agents solving the problem of distributed data analysis. *Proc. 2021 2nd Int. Conf. NeuroNT*, 2021, pp. 47–50. DOI: 10.1109/NeuroNT53022.2021.9472793.

Для цитирования

Юлейси Г.П., Холод И.И. Формальная модель многоагентных систем для федеративного обучения // Программные продукты и системы. 2022. Т. 35. № 1. С. 037–044. DOI: 10.15827/0236-235X.137.037-044.

For citation

Yuleisy G.P., Kholod I.I. A formal model of multiagent systems for federated learning. *Software & Systems*, 2022, vol. 35, no. 1, pp. 037–044 (in Russ.). DOI: 10.15827/0236-235X.137.037-044.