Distributed Multi-Agent Learning is More Effectively than Single-Agent

Shuya Ke

Kunming University of Science and Technology

Wenqi Liu (✉ liuwenq2215@sina.com)

Kunming University of Science and Technology

Research Article

Keywords: multi-agent system, opinion dynamics, stationary distribution, distributed system, time-series forecasting

Posted Date: November 11th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-1070129/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Distributed Multi-Agent Learning is More Effectively than Single-Agent

Shuya Ke^{1,2} and Wenqi Liu^{1,2*}

¹Data Science Research Center, Kunming University of Science and Technology, Kunming, 650500, China.

^{2*}Faculty of Science, Kunming University of Science and Technology, Kunming, 650500, China.

*Corresponding author(s). E-mail(s): liuwenq2215@sina.com;
Contributing authors: keshuya0816@sina.com;

Abstract

Interpretable distributed group intelligence techniques have emerged as an essential topic in artificial intelligence. The mathematical interpretability of prediction outcomes is critical for improving the reliability of machine learning, especially in random scenes. Although some experimental results published so far show that the prediction of group intelligence is better than individual intelligence, establishing a mathematical foundation for the superiority of distributed group intelligence is still a challenging problem for enhancing the interpretability of learning systems. Through the Radermacher complexity principle, we proved mathematically that the learning quality of group machine intelligence is better than its subset machine intelligence with a high probability, significantly better than any individual among them if the number of individuals in the group is large enough. We proposed a multi-agent distributed learning method for time series forecasting by incorporating multi-agent cooperation in cognitive processes into machine learning. In addition, since the way of cooperative interaction between multi-agent affects the training effect of the model, we provide a generalized interaction approach and prove its convergence. We conduct sufficient experiments on predicting time series for classically chaotic systems, and the results indicate that distributed group intelligence significantly improves the prediction accuracy of individual intelligence. The experiments result shows that the prediction error reduces substantially as the number of agents increases, confirming

the theoretical accuracy and the model's validity. This work provides new ideas for theoretically exploring how group intelligence emerges.

Keywords: multi-agent system, opinion dynamics, stationary distribution, distributed system, time-series forecasting

1 Introduction

Recently, multi-agent learning has gained popularity due to the development of generative adversarial networks and their application to complex systems [1–4]. Organisms utilize their variety to show their adaptability to the uncertainty of the real world. The real world is complete only when the whole group is integrated. In multi-agent cognitive learning, the group comprises individuals, who share a common objective but have distinct parameters. The differences exhibited by individuals through learning can explain world uncertainty and biodiversity. The emergence of group learning is a phenomenon that occurs as a consequence of individuals' continuous mutual learning.

Time series data has attracted intensive attention from academia and industry, continuously produced in industrial, environmental, social and healthcare applications. Acquiring accurate and reliable future trends is becoming a perpetual endeavour in many fields, providing the basis for various applications such as production planning, control, optimization, prevention, etc [5–8]. Traditional time series forecasting technologies such as autoregressive integrated moving average (ARIMA) [9–11], filtering-based approaches[12, 13], support vector machines[14–16], and others are restricted in their capacity to handle high-dimensional data and correctly represent complicated functions. Deep learning has shown significant benefits in this area, with various models based on long short-term memory (LSTM) [17] and recurrent neural networks (RNN) successfully applied to time series forecasting fields, including remote sensing[18], the atmosphere[19], language processing, and speech recognition[20, 21].

Most models learn the representation of potential features mainly at higher and more abstract multiple levels[22]. However, it is difficult for the upper layers to extract correct features based on the prior learning in the deep model, when the bottom layers do not collect sufficient and trustworthy data. Moreover, the learning result of each layer and layer-to-layer connection patterns will highly influence the final effect. Dan Schechtman thinks that group learning is the most efficient method of acquiring information for lack of prior knowledge[23]. To increase the model's learning efficiency, we integrated the multi-agent conceptual cognitive learning process seen in social networks into the model training process. The diverse social roles of cognitive agents create information asymmetry in social concept cognition. Reinforcement learning synthesizes social roles, and unknowable elements outside the specific cognitive

agent into an “environment”[24]. This approach can compensate for the shortcomings of the previous “open-loop learning” of cognitive agents. However, it cannot accurately reflect the complexity of the cognitive learning process for social ideas. A dopamine-based distributed reinforcement learning method confirms that the brain does not encode potential future rewards in a mean-valued way, but effectively and in parallel with multiple future rewards in a set-valued manner[25]. Their discovery demonstrates the feasibility of distributed reinforcement learning on neural networks.

In contrast to group learning, dopamine-based distributed reinforcement learning does not reflect cooperative learning behaviours, social role distinctions, and agent learning inclinations. Thus, the research of complicated multi-agent distributed social concept cognition theories and methods is urgently required to develop artificial intelligence. Swarm Learning, a decentralized machine learning method that surpasses collaborative learning, have developed to rapidly and accurately identify patients with severe and diverse illnesses. Four different illness cases show that Swarm Learning classifiers outperform single-agent classifiers, showing that multi-agent distributed learning beats single-agent learning[26]. This also verifies the profundity of the Chinese classic “The Book of Rites and Learning”, which says “one who studies alone and without companions is alone and uninformed”. Similarly, the literature[27] indicates that when the task is complex, the group solves the problem as fast as the fastest individual, and the efficiency will be better than the most efficient individual in the group. However, most of the results above are based on experimental evidence rather than theoretical proofs, and their mathematical foundations have become essential concerns in machine intelligence.

Cooperation is a common phenomenon in group learning, dolphins hunt in groups[28], humans cooperate to win[29], cooperation exists in all biological species and human cultures[30, 31]. Mathematical studies have shown that when the leader does not pursue a dominating strategy, the system co-evolves and stays in a cooperative condition. This phenomenon occurs in the Snowdrift and the Staghunt game as well[32]. This research offers mathematical evidence for cooperative learning in the leader-follower model to reach win-win outcomes. Cooperation may take many forms, including specialization, sacrifice, and coordination. The evolutionary population model consists of two interacting individuals that reproduce under stochastic environmental conditions. Reducing the correlation of fecundity between replication units is an evolutionary selection mechanism for collaboration, which explains how cooperation evolved[31]. Additionally, Swami Iyer et al. demonstrated that structured groups based on complex networks display a more excellent cooperative behaviour than unstructured groups[33]. That is, when the interaction between individuals creates a network, the network topology significantly affects the outcome of group learning. In this paper, we integrate cooperative learning and network structure into a novel machine learning framework, the multi-agent distributed LSTM, to improve the prediction accuracy of network models.

In recent years, numerous academics have investigated the mathematical foundations behind machine learning. For time series prediction, the interpretability of the prediction results is essential to improve the reliability of machine learning systems. Therefore, the research of interpretable machine learning methods has emerged as a hot topic for future research. In the previous work, we established a new condition for generator and discriminator data distribution agreement. Additionally, we propose a multi-agent distributed generative adversarial network (MADGAN) to address the multi-agent cognitive consistency problem in complicated distributed systems[34]. Chen et al. break through a critical nonlinear partial differential equation closely related to the optimal transport theory familiar in machine learning. For example, W-GAN belongs to optimal transmission theory[35]. Multimodal learning outperforms learning with its modal subsets, since the former provides a more accurate estimation of latent space representation[36]. In this paper, we show that utilizing multi-agent learning reduces overall risk compared to using single-agent learning when the size of the dataset is enough. We developed a multi-agent distributed learning framework based on LSTM networks to verify the validity of the theory. The experimental results corroborate the theories, demonstrating that reciprocal learning among agents may compensate for the cognitive process's lack of individual sample size and expertise.

In this paper, multi-agent learning outperforms subset learning is shown by minimizing empirical risk. The multi-agent distributed LSTM network (MAD-LSTM) is proposed as a new prediction framework. MAD-LSTM significantly increase prediction accuracy by simulating multi-agent conceptual cognitive learning and incorporating cooperative learning. The agents in multi-agent cognitive learning consist of LSTM networks with the same goal but distinct parameters, and they interact via exchanging parameter information. Notably, the interaction between agents may be regarded as information sharing between neurons at each layer. We evaluate the results using prominent benchmark datasets and real-world datasets. The experimental results demonstrate that when sufficient data is available, multi-agent distributed learning beats learning with a subset of agents.

2 Results

Notation This section first introduces the knowledge of Markov chains. $\mathbb{R}$, $\mathbb{N}$ denote the real and natural number spaces, respectively. The Markov chain state-space can be represented as $I = \{1, 2 \cdots, m\}$ when m agents learn from each other. $G = (V, E, W)$ is an m -order weighted directed graph with $V = \{1, 2, \cdots, m\}$ being the set of all agents. $E = \{e_{ij} \mid i, j \in V\}$ is the set of edges, which represents the information interaction or conceptual influence relationship between agents in the network. The edge e_{ij} indicates that agent j can receive information from agent i . The weighted adjacency matrix is $W = [w_{ij}]_{m \times m}$, where w_{ij} denotes the weight corresponding to edge e_{ji} , and also represents the one-step transfer probability from state i to state j . If

and only if $e_{ji} \in E$, there is $w_{ij} > 0$, otherwise, $w_{ij} = 0$. $W^{(n)}$ denotes the adjacency matrix of multi-agent interactions at n th time. When any times W are the same, then $W^{(n)} = W^n$. $N_i = \{j \in V : w_{ij} > 0\}$ denotes the collection of neighbors of node i . Additionally, if $w_{ii} > 0$, node i has a self-loop. $w_{ij}^{(n)}$ denotes the n th transfer probability from state i to state j . If the set $\{n : n \geq 1, w_{ii}^{(n)} > 0\}$ is not empty, then the greatest common divisor d of the set is called the period of state i . Specifically, when $d = 1$, state i is non-periodic. If states i and j interwork, that is, $i \leftrightarrow j$, then states i and j have the same period. If every state is interworking in state-space $I = \{1, 2, \dots, m\}$, it is an irreducible closed set. The irreducible aperiodic Markov chain with finite state must have a single stationary distribution $\{\pi_j, j \in I\}$, which is equivalent to the limit distribution [37, 38], denoted $\lim_{n \rightarrow \infty} w_{ij}^{(n)} = \pi_j, j \in I$.

2.1 Mathematical theorems on the excellence of distributed group learning

Multi-agent learning outperforms single-agent learning mathematically. Suppose X denotes the input domain and Y denotes the target domain. Consider a group of m agents, who are trained using the same data $x \in X$. Let $M = \{1, 2, \dots, m\}$ denote the set of all agents and $|M| = m$. K be a subset of all agents set M . F is the function class, $f^* : X \rightarrow Y$ denotes the true mapping from the input domain to the target domain. $f_i : X \rightarrow Y, i = 1, 2, \dots, m$ denotes the mapping of the agent i . Multi-agent learn from each other according to the adjacency matrix W . According to the condition for the existence of a stationary distribution [37], when the number of training $n \rightarrow \infty$, there is a stationary distribution $\{\pi_j, j \in M\}$, and $\sum_{j=1}^m \pi_j = 1$.

That is, $\lim_{n \rightarrow \infty} w_{ij}^{(n)} = \pi_j, j \in M$. Through mutual reference between agents, the learning result of agent i at the $n + 1$ th time is defined for input x , as $f_i^{(n+1)}(x) = \sum_{j=1}^m w_{ij} f_j^{(n)}(x), i = 1, 2, \dots, m$. Because of the existence of stationary distribution for opinion shifts, group learning is consistent, namely, $|f_i^{(n)}(x) - f_j^{(n)}(x)| \rightarrow 0 (n \rightarrow \infty)$ for input x (see Methods for details). Then, the consistency of group learning is defined as

$$f_M(x) = \lim_{n \rightarrow \infty} f_i^{(n+1)}(x) = \sum_{j=1}^m \pi_j f_j^{(0)}(x), i, j = 1, 2, \dots, m$$

where $f_i^{(n+1)}$ denotes the mapping function when agent i is trained $n + 1$ th time and $f_j^{(0)}$ denotes the initial mapping function of agent j . Let $f_M = \sum_{j=1}^m \pi_j f_j^{(0)}$ denote the mapping function eventually learned by the m agents.

Let F_M denote the mapping function class from the domain X to the domain Y containing m agents, then as follows:

$$F_M \triangleq \left\{ f_M : X \rightarrow Y \mid f_M(x) = \sum_{j=1}^m \pi_j f_j^{(0)}(x), f_j \in F_j, j = 1, 2, \dots, m, \right\}$$

The data pair $(x, y) \in X \times Y$ is sampled from the total data in the learning task, obeying the unknown distribution D , that is

$$P_D(x, y) \triangleq P_{y|x}(y \mid f^*(x))P_x(x) \quad (1)$$

Given a dataset $Q = ((x_i, y_i))_{i=1}^s$, where (x_i, y_i) is obtained by independent sampling from the total dataset. Based on the Empirical Risk Minimization (ERM) principle[39], the goal of learning is to find the mapping function learned by the m agents $f_M \in F_M$ to jointly minimize the empirical risk, that is:

$$\min \hat{r}(f_M) \triangleq \frac{1}{s} \sum_{i=1}^s l(f_M(x_i), y_i) = \frac{1}{s} \sum_{i=1}^s l\left(\sum_{j=1}^m \pi_j f_j^{(0)}(x_i), y_i\right) \quad (2)$$

where $l(\cdot, \cdot)$ denotes the loss function. Given $\hat{r}(f_M)$, we can define the corresponding population risk as

$$r(f_M) = E_{(x_i, y_i) \sim D} [\hat{r}(f_M)] \quad (3)$$

As in the previous research [40, 41], we utilized the population risk to measure the performance of learning.

Representation learning quality was introduced to measure the goodness of the learned representations.

Definition 1 Given a data distribution of the form (1), for any learned representation mapping $f \in F$, the representation learning quality is defined as

$$\varphi(f) = |r(f) - r(f^*)|$$

$\varphi(f)$ can measure the loss due to the distance between f and f^* .

The analytical proof leads us to the following Theorem 1. The detailed proof of Theorem 1 is provided in Methods.

Theorem 1 Let $Q = ((x_i, y_i))_{i=1}^s$ be a dataset of s samples independently sampled from the distribution D , and the size of the sample is sufficiently big. M, K represent the set of m, k agents, where $K \subset M$. Assume we have trained the optimum empirical minimal risk on m and k agents to get $\hat{f}_M = \sum_{j=1}^m \pi_j \hat{f}_j^{(0)}$ and $\hat{f}_K = \sum_{j=1}^k \pi_j \hat{f}_j^{(0)}$. Then, for all $1 > \delta > 0$, at least with probability $1 - \delta$ satisfies

$$\varphi(\hat{f}_M) \leq \varphi(\hat{f}_K) \quad (4)$$

The former has better representation learning than the latter, thus indicating that the learning efficiency is better in m agents than in k agents.

$\hat{f}_M$ has a more sufficient space exploration than $\hat{f}_K$, as shown in Fig. 1c. In addition, our experimental results also show that multi-agent learn better than their subsets.

Fig. 1 Multi-agent representation learning and the architecture of MAD-LSTM. **a**, The architecture of MAD-LSTM with three agents. The three agents learn from each other according to the adjacency matrix W . **b**, The architecture of LSTM. **c**, The comparison of m agents and k agents representation learning, where $k < m$ and $k > 1$. Y_1 , Y_K , Y_M , Y^* are high-dimensional vectors on the target domain Y . $\hat{f}_1$, $\hat{f}_K$, $\hat{f}_M$, $\hat{f}^*$ denote the mapping functions from the input domain X to the target domain Y .

2.2 Experimental results of distributed predictive models

MAD-LSTM approach. The long short-term memory (LSTM) model is the basic network to describe the multi-agent distributed learning process (Fig. 1b). The LSTM is regarded as an agent (physical or abstract object), and the training process for the LSTM parameters is disregarded. The agent can receive and transmit information with the surrounding individuals in a limited range. The training process of multiple LSTM networks is the process of conceptual state transfer between agents. The interaction between agents forms a network, and the weights of the network (edges) indicate the degree of influence between two agents. Multi-agent learn from each other with adjacency matrix W (Fig. 1a).

Multi-agent learning is better than single-agent. Assuming $m = 3$, we construct a multi-agent distributed LSTM (MAD-LSTM) network, which composes three agents with identical network structures. Three agents learn from each other with a random matrix

$$W = \begin{pmatrix} 0.9 & 0.05 & 0.05 \\ 0.1 & 0.8 & 0.1 \\ 0.025 & 0.025 & 0.95 \end{pmatrix}$$

The random matrix W is determined empirically and is retained throughout the training procedure. To verify that multi-agent learning outperforms single-agent, we trained MAD-LSTM and LSTM using Mackey-Glass, Lorenz chaotic time series and Beijing PM2.5 concentration datasets, respectively. More details about training data are described in appendix A. We provide the experimental results in Table 1. The results indicate that MAD-LSTM has a superior training effect to a single LSTM.

Fig. 2 Multi-agent distributed learning training loss is lower than that of a single agent. Results of MAD-LSTM experiments more than ten runs on three datasets. The number of agents m highly affects the results of MAD-LSTM network experiments.

Table 1 Experimental results of LSTM and MAD-LSTM on three datasets. The experimental results are the mean of more than ten runs.

Datasets	Method	MAE
Mackey-Glass	LSTM	0.0027
Mackey-Glass	MAD-LSTM	0.0011
Lorenz	LSTM	0.0374
Lorenz	MAD-LSTM	0.0287
PM2.5	LSTM	15.18
PM2.5	MAD-LSTM	11.39

Further, we explore the relationship between the number of agents and the experimental results. We use three datasets to train MAD-LSTM networks containing different numbers of agents, respectively. Table 2 provides the experimental results for different numbers of agents. Fig. 2 shows that distributed learning with two agents is better than single agent. Similarly, distributed learning with five agents is better than with two agents, thus verifying the validity of Theorem 1. Detailed analysis of the experimental results is provided in Appendix Fig. A1.

The effectiveness of the MAD-LSTM network structure. The literature [33] shows that groups with a network structure exhibit higher levels of cooperative behavior. The MAD-LSTM network describes the process of

Table 2 MAE (mean absolute error) obtained by training with different datasets for different number of agents. m denotes the number of agents. The experimental results are the mean of more than ten runs.

Datasets	m = 1	m = 2	m = 3	m = 4	m = 5
Mackey-Glass	0.0027	0.0013	0.0012	0.0011	0.001
Lorenz	0.0374	0.0304	0.0287	0.0259	0.0233
PM2.5	15.18	12.1	11.39	11.3	10.78

Table 3 Experimental results for different models. The experimental results are the mean of more than ten runs.

Datasets	Method	MAE
Mackey-Glass	LSTM-1	0.0032
	LSTM-2	0.0027
	LSTM-3	0.0029
	NC-LSTM	0.0029
	MAD-LSTM	0.0012
Lorenz	LSTM-1	0.036
	LSTM-2	0.0367
	LSTM-3	0.0374
	NC-LSTM	0.0367
	MAD-LSTM	0.0287
PM2.5	LSTM-1	16.54
	LSTM-2	15.18
	LSTM-3	16.73
	NC-LSTM	16.15
	MAD-LSTM	11.39

multi-agents learning from each other and eventually reaching a consensus on mapping functions. The weights connecting the agents represent their degree of impact on one another. Therefore, we can construct a adjacency matrix based on the degree of influence between the agents. Assuming $m = 3$, we train three independent agents (LSTM-1, LSTM-2 and LSTM-3), MAD-LSTM and Non-cooperative MAD-LSTM (NC-LSTM) networks with three datasets, respectively. NC-LSTM indicates that the three agents do not share parameters based on the adjacency matrix, but average the training results to produce the final prediction. Table 3 shows that the training loss of MAD-LSTM is lower than that of the other two networks, demonstrating the effectiveness of the multi-agent distributed learning architecture.

3 Discussion

This work demonstrates that multi-agent learning outperforms single-agent learning when the size of the dataset is enough, as the former leads to better representation learning. This paper establishes a theoretical foundation for distributed multi-agent learning. The same conclusion can be obtained in practice even if the training samples are different for each agent. For example, multimodal learning is better than single, and multi-view learning is better than single-view[40]. Multimodal and multi-view learning can essentially be understood as multi-agent learning separately. In summary, our conclusions are generalizable.

Collaboration is a common phenomenon in group learning. In this paper, through merging cooperative learning and network architecture into a machine learning framework, we construct a multi-agent distributed LSTM. In multi-agent cognitive learning, any agent is composed of an LSTM network with the same goals but different parameters. We propose a new machine learning framework, multi-agent distributed LSTM (MAD-LSTM), to enhance prediction accuracy. We evaluate the Multi-agent Distributed LSTM (MAD-LSTM) framework on Mackey-Glass, Lorenz chaotic time series data and Beijing PM2.5 concentration datasets to verify theoretical correctness and model validity. PM2.5 concentration belongs to a complex weather environment system with specific chaotic nature. The experimental results show that multi-agent learning outperforms its subset for classically chaotic systems.

In particular, agent interaction learning methods significantly impact the final learning outcome in multi-agent learning. There are various methods of interaction between multi-agent, but the most prevalent are described in this paper. We analyze the convergence of a multi-agent distributed learning framework based on the existence of a stationary distribution of Markov chains in the Degroot [42] model. In fact, the adjacency matrix can be replaced with a time-varying matrix, and the multi-agent interaction can be changed, which are worth further exploration.

4 Methods

4.1 Rademacher complexity principle and proof of theorem 1

First, we provide a few assumptions. Assumption 1 : The loss function $l(\cdot, \cdot)$ is L-smooth concerning the first coordinate and is bounded by a constant C . Assumption 2 : The true mapping f^* is included in function class F . Assumption 3 : For any set $K \subset M$, we have $F_k \subset F_m \subset F$. In theoretical analysis, Assumption 1 is the regularity condition for loss functions. Assumption 2 is the realizability condition in representation learning, which guarantees that the function class we optimize contains the true mapping[40].

Next, we introduce the Rademacher complexity, which describes the hypothesis space complexity and considers the data distribution. We use it

to quantify the population risk performance based on different agents. Specifically, let F_M be function class containing m agents $\mathbb{R}^d \rightarrow [a, b]^X$. Let $x_1, \dots, x_s$ be an independent random variable obeying some distribution D on $\mathbb{R}^d$. $Q = (x_1, \dots, x_s)$ denotes the sample set. The Empirical Rademacher complexity[43] of F_M with respect to the sample set Q is

$$\hat{R}_Q(F_M) = E_\sigma \left[\sup_{f \in F} \frac{1}{s} \sum_{i=1}^s \sigma_i f_M(x_i) \right] = E_\sigma \left[\sup_{f \in F} \frac{1}{s} \sum_{i=1}^s \sum_{j=1}^m \sigma_i \pi_j f_j^{(0)}(x_i) \right] \quad (5)$$

where $\sigma = (\sigma_1, \dots, \sigma_s)^T$, σ_i is an independent random variable taking values of $\{-1, 1\}$, also known as the Rademacher variable. The Empirical Rademacher complexity measures the correlation between the function class F_M and the random noise σ in the sample set Q . The Rademacher complexity of the function class F_M is define by that

$$R_s(F_M) = E_Q \left[\hat{R}_Q(F_M) \right] \quad (6)$$

In addition, we introduce McDiarmid's inequality and the property of Radmacher complexity in the following lemma.

Lemma 1.[43] Let the random variables $x, \{x_i\}_{i=1}^s$ be a set of samples independently sampled from the space X , and $F \subseteq [a, b]^X$ be a set of bounded functions. Then

$$E \left[\sup_{f \in F} (E[f(x)] - \frac{1}{s} \sum_{i=1}^s f(x_i)) \right] \leq 2R_s(F) \quad (7)$$

Lemma 2.[43] Let F and H be real function classes, then for $R_s(F)$ satisfies: 1. If $H \subseteq F$, $R_s(H) \leq R_s(F)$. 2. If $\phi: \mathbb{R} \rightarrow \mathbb{R}$ is Lipschitz with constant L_ϕ and satisfies $\phi(0) = 0$, then $R_s(\phi \circ F) \leq 2L_\phi R_s(F)$. 3. For any uniformly bounded function c , $R_s(F + c) \leq R_s(F) + \|c\|_\infty / \sqrt{s}$.

Lemma 3 (McDiarmid's Inequality).[43] If $x_1, x_2, \dots, x_s$ is an independent random variable taken from the set A , and for any $1 \leq i \leq s$, the function $f: A^n \rightarrow [a, b]^X$ satisfies

$$\sup_{x_1, \dots, x_s, x'_i} |f(x_1, \dots, x_s) - f(x_1, \dots, x_{i-1}, x'_i, x_{i+1}, \dots, x_s)| \leq c_i$$

For any $\varepsilon > 0$, then

$$\Pr [f(x_1, \dots, x_s) - E[f(x_1, \dots, x_s)] \geq \varepsilon] \leq \exp^{-2\varepsilon^2 / \sum_{i=1}^s c_i^2}$$

Proof of Theorem 1

$$\varphi(\hat{f}_M) = |r(\hat{f}_M) - r(f^*)| \leq |r(\hat{f}_M) - \hat{r}(\hat{f}_M)| + |\hat{r}(\hat{f}_M) - \hat{r}(f^*)| + |\hat{r}(f^*) - r(f^*)|$$

Let $J_1 = |r(\hat{f}_M) - \hat{r}(\hat{f}_M)|$, $J_2 = |\hat{r}(\hat{f}_M) - \hat{r}(f^*)|$, $J_3 = |\hat{r}(f^*) - r(f^*)|$, where J_2 is the centering empirical risk, and

$$J_1 + J_3 \leq \sup_{f_M \in F_M} |r(f_M) - \hat{r}(f_M)| + \sup_{f \in F} |r(f) - \hat{r}(f)|$$

where

$$\begin{aligned} & \sup_{f_M \in F_M} |\hat{r}(f_M) - r(f_M)| \\ &= \sup_{f_M \in F_M} \left| \frac{1}{s} \sum_{i=1}^s l(f_M(x_i), y_i) - E_{(x'-y') \sim D} [l(f_M(x'), y')] \right| \end{aligned} \quad (8)$$

By Assumption 1, l is bounded by the constant C , and for any (x, y) we have $0 \leq l(f_M(x), y) \leq C$. If a pair of data (x_i, y_i) is changed, the above equation is changed by at most $\frac{2C}{s}$. According to McDiarmid's inequality, with probability $1 - \delta/2$ we get the equation:

$$\begin{aligned} & \sup_{f_M \in F_M} |\hat{r}(f_M) - r(f_M)| \\ & \leq E_{(x_i, y_i) \sim D} \sup_{f_M \in F_M} \left| \frac{1}{s} \sum_{i=1}^s l(f_M(x_i), y_i) - E_{(x'-y') \sim D} [l(f_M(x'), y')] \right| \quad (9) \\ & + C \sqrt{\frac{2 \ln(2/\delta)}{s}} \end{aligned}$$

$$\begin{aligned} \text{Let } J_{11} &= E_{(x_i, y_i) \sim D} \sup_{f_M \in F_M} \left| \frac{1}{s} \sum_{i=1}^s l(f_M(x_i), y_i) - E_{(x'-y') \sim D} [l(f_M(x'), y')] \right|, \\ J_{12} &= C \sqrt{\frac{2 \ln(2/\delta)}{s}}. \end{aligned}$$

Consider the function class containing m agents

$$l_{F_M} = \{(x, y) \mapsto l(f_M(x), y) \mid f_M \in F_M\}$$

Suppose $F = l_{F_M}$ in Lemma 1, then the upper bound of J_{11} is $2R_s(l_{F_M})$. To utilize the hypothesis function class directly, we need to decompose the Rademacher term consisting of the loss function class. We center the function $l'(f_M(x), y) = l(f_M(x), y) - l(\vec{0}, y)$. According to Lemma 2(3), we know that

$$R_s(l_{F_M}) \leq R_s(l'_{F_M}) + \frac{C}{\sqrt{s}}$$

Because l' is Lipschitz's first coordinate with constant L and $l'(f_M(\vec{0}), y) = 0$. According to Lemma 2(2) we can get:

$$R_s(l'_{F_M}) \leq 2LR_s(F_M)$$

Combining the above Rademacher complexity analysis, we can get

$$J_{11} \leq 2R_s(l_{F_M}) \leq 2R_s(l'_{F_M}) + \frac{2C}{\sqrt{s}} \leq 2 \cdot 2LR_s(F_M) + \frac{2C}{\sqrt{m}}$$

i.e.

$$\sup_{f_M \in F_M} |\hat{r}(f_M) - r(f_M)| \leq 4LR_s(F_M) + \frac{2C}{\sqrt{s}} + C \sqrt{\frac{2 \ln(2/\delta)}{s}}$$

similarly,

$$\sup_{f \in F} |\hat{r}(f) - r(f)| \leq 4LR_s(F) + \frac{2C}{\sqrt{s}} + C \sqrt{\frac{2 \ln(2/\delta)}{s}}$$

Therefore, we can obtain with probability $1 - \delta$:

$$J_1 + J_3 \leq 4LR_s(F_M) + 4LR_s(F) + \frac{4C}{\sqrt{s}} + 2C\sqrt{\frac{2\ln(2/\delta)}{s}}$$

In summary, the probability of at least $1 - \delta$ satisfies:

$$\eta(\hat{f}_M) \leq 4LR_s(F_M) + 4LR_s(F) + \frac{4C}{\sqrt{s}} + 2C\sqrt{\frac{2\ln(2/\delta)}{s}} + \hat{L}(\hat{f}_M, Q)$$

where $\hat{L}(\hat{f}_M, Q) \triangleq |\hat{r}(\hat{f}_M) - \hat{r}(f^*)|$ is the empirical loss of the center.

Consider the multi-agent set $K \subset M$. According to Assumption 3, it is known that optimizing on a larger class of function yields a smaller empirical risk. Therefore

$$\hat{L}(\hat{f}_M, Q) \leq \hat{L}(\hat{f}_K, Q)$$

The Rademacher complexity $R_s(F)$ with sample size s is bounded by $\sqrt{C(F)/s}$, where $C(F)$ represents the intrinsic property of function class F . Thus, the boundary of $R_s(F_M)$ is $\sqrt{C(F_M)/s}$ and the boundary of $R_s(F_K)$ is $\sqrt{C(F_K)/s}$. According to Lemma 2(1), we can obtain $C(F_K) \leq C(F_M)$. Then

$$\varphi(\hat{f}_M) - \varphi(\hat{f}_K) \leq 0$$

i.e.

$$\hat{L}(\hat{f}_K, Q) - \hat{L}(\hat{f}_M, Q) \geq \sqrt{\frac{C(F_M)}{s}} - \sqrt{\frac{C(F_K)}{s}}$$

The above equation shows that: (1) when the size of data s increases, the effect of the intrinsic complexity of the function class decreases. (2) Using more agents to learn can effectively optimize the empirical risk and thus improve the representation learning quality. When $K \subset M$ and the size of training sample s is big enough, $\varphi(\hat{f}_M) \leq \varphi(\hat{f}_K)$ is fulfilled with at least $1 - \delta$ probability, indicating that m agents learn better than single agent. □

4.2 Construction of distributed systems and proof of consistency (convergence)

Long short-term memory (LSTM). Theoretically, recurrent neural networks can handle sequence inputs of arbitrary length, but they are not well trained for lengthy sequences due to gradient disappearance, gradient explosion and lack of computational resources. To address this problem, Hochreiter and Schmidhuber et al. proposed a new network topology, the long short-term memory (LSTM) model, in 1997. The LSTM overcomes this limitation by incorporating a specific structure of “cellular cells” into the network. This complex mechanism effectively captures long-term relationships between cells without significantly increasing the amount of parameters.

Long and short-term memory models are a sort of recurrent structure suited for processing and forecasting significant events in time series with extremely long intervals and delays. The classical long and short-term memory models include input gates, output gates, forgetting gates, and cellular cell structures. Fig. 1b depicts one of the more prevalent long and short-term memory models.

Model architecture of MAD-LSTM. The LSTM is regarded as an agent. The training process of multiple LSTM networks is the process of conceptual state transfer between agents. Agents can be classified as “leaders” or “followers” based on their degree of influence on one another. Multi-agent learn from each other with a random matrix W with row sum 1. All agents first learn the same dataset separately to form their own mapping function. Then each agent updates their mapping functions based on the random matrix W . The weighted average of all agents’ new mapping functions is the output of the whole network system. Finally, the network system updates parameters based on losses. The mapping function of an agent is materialized as a network parameter or a high-dimensional feature vector in machine learning. Fig. 1a shows a multi-agent distributed LSTM network structure with three agents.

Let $f_i^{(n)}$ denote the mapping function of agent i at the n th time in an influence network $G = (V, E, W)$ composed of m agents, where $n \in \mathbb{N}$. The mapping function of agent i at the $n+1$ th time is determined by the weighted average of its own and its neighboring agents’ mapping functions at the n th time [42], specifically,

$$f_i^{(n+1)} = \sum_{j=1}^m w_{ij} f_j^{(n)} \quad (10)$$

where w_{ij} is an element of the adjacency matrix W , and satisfies $\sum_{j=1}^m w_{ij} = 1$.

Let $F^{(n)} = (f_1^{(n)}, f_2^{(n)}, \dots, f_m^{(n)})^T \in \mathbb{R}^n$ denote the vector of all agents’ mapping functions at the n th time. The matrix form of equation (10) is

$$F^{(n+1)} = WF^{(n)} \quad (11)$$

where W is a random matrix, i.e., a non-negative matrix with a row sum equal to 1. Network G depicts multi-agent interactions in a distributed learning system.

MAD-LSTM convergence analysis. Participant role heterogeneity is mentioned as important for cooperation [32]. Agents are divided into “leaders” and “followers” based on mutual influence. Specifically, any node $k \in N_i$, satisfies $w_{ik} = \max_{j \in N_i} w_{ij}$, i.e., node i receives the most information from node k or is most influenced by it in the set N_i of nodes. Then designate node k as the leader of node i and node i as its follower. In fact, the number of leaders influences network convergence [34]. Considering the complexity of the network, this paper focuses on the multi-agent distributed network model with a single leader.

In an influence network $G = (V, E, W)$ consisting of m agents, it is assumed that W remains constant throughout the learning process. The initial mapping function of agent i is $f_i^{(0)}$, where $i \in V$. Next, the mapping function of agent i is updated according to equation (10), i.e., $f_i^{(1)} = \sum_{j=1}^m w_{ij} f_j^{(0)}$, written as a matrix of the form $F^{(1)} = WF^{(0)}$. According to equation (11), all agents’

mapping functions are updated, thus

$$\begin{aligned}
 F^{(1)} &\leftarrow WF^{(0)} \\
 F^{(2)} &\leftarrow WF^{(1)} = W^2F^{(0)} \\
 &\vdots \\
 F^{(n+1)} &\leftarrow WF^{(n)} = W^{n+1}F^{(0)}
 \end{aligned} \tag{12}$$

According to the condition for the existence of a stationary distribution [37], when $n \rightarrow \infty$, there is a stationary distribution, that is, $\lim_{n \rightarrow \infty} w_{ij}^{(n)} = \pi_j$, $j \in M$. Then $\lim_{n \rightarrow \infty} f_i^{(n+1)} = \lim_{n \rightarrow \infty} \sum_{j=1}^m w_{ij}^{(n)} f_j^{(0)} = \sum_{j=1}^m \pi_j f_j^{(0)}$.

In general, the multi-agent distributed learning model will converge when n is big enough. Simultaneously, the loss function guides the whole network to optimal.

4.3 Hyperparameter setting.

All agents are three hidden layer LSTM networks with the same number of neurons in each layer. We utilize Adam[44] as the optimizer, MAE (mean absolute error) as the loss, a batch size of 120, and default values for other hyperparameters. In addition, we monitor the “val-loss” via the keras callback class function (keras.callbacks). If the value of “val-loss” does not drop after three epochs, we multiply it by 0.5 to reduce the learning rate. Set the minimum value of the learning rate to 0.00001 and do not decrease it afterwards. When using the Mackey-Glass and Lorenz datasets, the hidden layer of the neural network was set to 32 neurons, and the first 200 data were used to predict the next one. When using the PM 2.5 dataset, the hidden layer of the neural network was configured with 48 neurons, and all monitoring data from the previous five days (220 hours) were used to predict the PM2.5 concentration one day (24 hours) into the future, i.e. the PM2.5 concentration in the early hours of day 7. The Lorenz dataset was trained 300 times, whereas the other datasets were trained 200 times.

Appendix A Several chaotic systems and real datasets

Dataset. To validate our model, we employ two benchmark time series datasets and a real-world dataset. The following is a brief overview of the datasets.

(1) Mackey-Glass: Mackey-Glass time series forecasting is a well-known non-linear fitting problem that serves as the test data for many research. The series data is generated by the following time lag differential equation:

$$\frac{dx(t)}{dt} = \frac{\alpha x(t - \tau)}{1 + x^\beta(t - \tau)} + \gamma x(t), \tag{A1}$$

where $x(t)$ is the generated data, τ represents a time delay parameter, α , β , and γ are the parameters. Setting $\alpha = 0.2$, $\beta = 0.1$, $\gamma = 10$, $\tau = 17$, we get the training data.

(2) Lorenz: In the area of chaos, the Lorenz model has become the widely accepted model. It is a fully deterministic system of third-order ordinary differential equations proposed by the American meteorologist Lorenz's research of atmospheric motion via simplifying the convection model. The equation takes the following form:

$$\begin{aligned}\frac{dx}{dt} &= \sigma(y - x) \\ \frac{dy}{dt} &= \rho x - y - xz \\ \frac{dz}{dt} &= xy - \beta z\end{aligned}\tag{A2}$$

where σ denotes the Planter number, ρ is the Rayleigh number, and β denotes the angle ratio. Setting $\sigma = 10$, $\rho = 28$, and $\beta = 3$, We obtained 1000 3D evolutionary trajectory data.

(3) PM 2.5: The Beijing air PM2.5 concentration dataset is another widely-used public benchmark dataset. This paper only considers a sample of data from Beijing, observed from January 1, 2010, to December 31, 2014. The dataset contains PM2.5 concentrations (pm2.5), dew points (DEWP), temperatures (TEMP), atmospheric pressure (PRES), wind direction (cbwd), wind speed (lws), cumulative snowfall (ls), and accumulated rainfall (lr). The dataset includes 43,800 observations in total. We trained on 80% of the data and validated on 20%. The goal of the experiment was to use the first 200 data to predict PM2.5 concentration data after an interval of 24 hours.

Fig. A1 Distribution of experimental results for more than ten runs. The points in the graph above are the experiment results mean of more than ten runs. The graphs below show the results of all experiments. The experimental results easily show that the multi-agent distributed learning outperforms its subsets.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 61573173).

Author contributions

All content covered in this paper was done by both authors.

Competing interests

The authors declare no competing interests.

Data availability

The source data involved in the experiments can be found in the Appendix A. To facilitate the reproducibility of this work, the data required for the experimental runs are publicly available on GitHub (<https://github.com/ka-ka-girl/MAD-LSTM-tensorflow-master/tree/main/datasets>).

Code availability

The code used in the research is publicly available from the GitHub repository: <https://github.com/ka-ka-girl/MAD-LSTM-tensorflow-master>.

References

- [1] Brown, N. & Sandholm, T. Superhuman AI for multiplayer poker. *Science* **365**, 885-890 (2019).
- [2] Yang, C., Yuan, K., Zhu, Q., Yu, W. & Li, Z. Multi-expert learning of adaptive legged locomotion. *Science Robotics* **5**, 49 (2020).
- [3] Novati, G., de Laroussilhe, H. L. & Koumoutsakos, P. Automating turbulence modelling by multi-agent reinforcement learning. *Nature Machine Intelligence* **3**, 87-96 (2021).
- [4] Bichler, M., Fichtl, M., Heidekrüger, S., Kohring, N. & Sutterer, P. Learning equilibria in symmetric auction games using artificial neural networks. *Nature Machine Intelligence* **3**, 687-695 (2021).
- [5] Box, G. E., Jenkins, G. M., Reinsel, G. C. & Ljung, G. M. *Time series analysis: forecasting and control* (John Wiley & Sons, 2015).
- [6] Brockwell, P. J., Brockwell, P. J., Davis, R. A. & Davis, R. A. *Introduction to time series and forecasting* (springer, 2016).

- [7] Ravuri, S. et al. Skilful precipitation nowcasting using deep generative models of radar. *Nature* **597**, 672–677 (2021).
- [8] DeFelice, N. B., Little, E., Campbell, S. R. & Shaman, J. Ensemble forecast of human West Nile virus cases and mosquito infection rates. *Nature Communications* **8**, 1-6 (2017).
- [9] Saboia, J. L. M. Autoregressive integrated moving average (ARIMA) models for birth forecasting. *Journal of the American Statistical Association* **72**, 264-270 (1977).
- [10] Thiruchelvam, L., Dass, S. C., Asirvadam, V. S., Daud, H. & Gill, B. S. Determine neighboring region spatial effect on dengue cases using ensemble ARIMA models. *Scientific Reports* **11**, 1-9 (2021).
- [11] Klimek, P., Poledna, S. & Thurner, S. Quantifying economic resilience from input–output susceptibility to improve predictions of economic growth and recovery. *Nature communications* **10**, 1-9 (2019).
- [12] Wu, X. & Wang, Y. Extended and Unscented Kalman filtering based feed-forward neural networks for time series prediction. *Applied Mathematical Modelling* **36**, 1123-1131 (2012).
- [13] Joo, T. W. & Kim, S. B. Time series forecasting based on wavelet filtering. *Expert Systems with Applications* **42**, 3868-3874 (2015).
- [14] Han, Z., Liu, Y., Zhao, J. & Wang, W. Real time prediction for converter gas tank levels based on multi-output least square support vector regressor. *Control Engineering Practice* **20**, 1400-1409 (2012).
- [15] Chen, Y. M., Lin, P., He, J. Q., He, Y. & Li, X. L. Combination of the manifold dimensionality reduction methods with least squares support vector machines for classifying the species of sorghum seeds. *Scientific reports* **6**, 1-10 (2016).
- [16] Ertekin, S., Bottou, L. & Giles, C. L. Nonconvex online support vector machines. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **33**, 368-381 (2010).
- [17] Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural computation* **9**, 1735-1780 (1997).
- [18] Zhang, L., Zhang, L. & Du, B. Deep learning for remote sensing data: A technical tutorial on the state of the art. *IEEE Geoscience and Remote Sensing Magazine* **4**, 22-40 (2016).

- [19] Bandara, K., Bergmeir, C. & Hewamalage, H. LSTM-MSNet: Leveraging forecasts on sets of related time series with multiple seasonal patterns. *IEEE transactions on neural networks and learning systems* **32**, 1586-1599 (2020).
- [20] Tsai, S. T., Kuo, E. J. & Tiwary, P. Learning molecular dynamics with simple language model built upon long short-term memory neural network. *Nature communications* **11**, 1-11 (2020).
- [21] Heelan, C., Lee, J., O'Shea, R., Lynch, L., Brandman, D. M., Truccolo, W. & Nurmikko, A. V. Decoding speech from spike-based neural population recordings in secondary auditory cortex of non-human primates. *Communications biology* **2**, 1-12 (2019).
- [22] LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436-444 (2015).
- [23] Vinyals, O. et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, **575**, 350-354 (2019).
- [24] Saggio, V. et al. Experimental quantum speed-up in reinforcement learning agents. *Nature* **591**, 229-233 (2021).
- [25] Dabney, W., Kurth-Nelson, Z., Uchida, N., Starkweather, C. K., Hassabis, D., Munos, R. & Botvinick, M. A distributional code for value in dopamine-based reinforcement learning. *Nature* **577**, 671-675 (2020).
- [26] Warnat-Herresthal, S. et al. Swarm Learning for decentralized and confidential clinical machine learning. *Nature* **594**, 265-270 (2021).
- [27] Almaatouq, A., Alsobay, M., Yin, M. & Watts, D. J. Task complexity moderates group synergy. *Proceedings of the National Academy of Sciences* **118**, 1-9 (2021).
- [28] Matrai, E., Kwok, S. T., Boos, M. & Pogány, Á. Group size, partner choice and collaborative actions in male Indo-Pacific bottlenose dolphins (*Tursiops aduncus*). *Animal Cognition*, 1-15 (2021).
- [29] Milinski, M. Tit for tat in sticklebacks and the evolution of cooperation. *Nature* **325**, 433-435 (1987).
- [30] Wedekind, C. & Milinski, M. Human cooperation in the simultaneous and the alternating Prisoner's Dilemma: Pavlov versus Generous Tit-for-Tat. *Proceedings of the National Academy of Sciences* **93**, 2686-2689 (1996).
- [31] Koduri, N. & Lo, A. W. The origin of cooperation. *Proceedings of the National Academy of Sciences* **118**, 1-7 (2021).

- [32] Mu, Y. & Guo, L. How cooperation arises from rational players?. *Science China Information Sciences* **11**, 1-9 (2013).
- [33] Iyer, S. & Killingback, T. Evolution of cooperation in social dilemmas on complex networks. *PLoS computational biology* **12**, e1004779 (2016).
- [34] Ke, S. & Liu, W. Consistency of multiagent distributed generative adversarial networks. *IEEE Transactions on Cybernetics*, 1-11 (2020). Preprint at <https://ieeexplore.ieee.org/document/9233959>.
- [35] Chen, S., Liu, J. & Wang, X. J. Global regularity for the Monge-Ampère equation with natural boundary condition. Preprint at <https://arxiv.org/abs/1802.07518> (2018).
- [36] Huang, Y., Du, C., Xue, Z., Chen, X., Zhao, H. & Huang, L. What Makes Multimodal Learning Better than Single (Provably). Preprint at <https://arxiv.org/abs/2106.04538> (2021).
- [37] Parzen, E. *Stochastic processes* (SIAM, 1999).
- [38] Ross, S. M. et al. *Stochastic processes*. (Wiley New York, 1996).
- [39] Mohri, M., Rostamizadeh, A. & Talwalkar, A. *Foundations of machine learning*. (MIT press, 2018).
- [40] Amini, M. R., Usunier, N. & Goutte, C. Learning from multiple partially observed views-an application to multilingual text categorization. *Advances in neural information processing systems* **22**, 28-36 (2009).
- [41] Tripuraneni, N., Jordan, M. I. & Jin, C. On the theory of transfer learning: The importance of task diversity. Preprint at <https://arxiv.org/abs/2006.11650> (2020).
- [42] DeGroot, M. H. Reaching a consensus. *Journal of the American Statistical Association* **69**, 118-121 (1974).
- [43] Bartlett, P. L. & Mendelson, S. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research* **3**, 463-482 (2002).
- [44] Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. Preprint at <https://arxiv.org/abs/1412.6980> (2014).