Software Engineering and Applications 软件工程与应用, 2023, 12(3), 463-473
Published Online June 2023 in Hans. https://www.hanspub.org/journal/sea
https://doi.org/10.12677/sea.2023.123046

Hans汉斯

基于元学习的MIMO接收机学习方法设计

汪流航，王保云

南京邮电大学，通信与信息工程学院，江苏 南京

收稿日期：2023年5月13日；录用日期：2023年6月7日；发布日期：2023年6月19日

摘　要

基于数据驱动的接收机通常以提高训练成本为代价提升性能。本文对基于深度神经网络(Deep Neural Network, DNN)的多输入多输出(Multiple Input Multiple Output, MIMO)接收机的学习方式进行设计，旨在降低基于DNN的接收机的训练成本并提升性能。解决包括周期性发送导频信号导致网络被频繁的训练，即需要大量的训练数据和训练时间以及网络收敛慢等问题。根据元学习的学习方式，对系统进行基于任务的构建。提出了包括元训练、元适应和测试三阶段的接收机学习算法。并在训练的过程中对学习过程进行了优化，包括梯度简化和参数继承，提升了网络的收敛速度和泛化能力。在实验设计中与传统的接收机在不同信道模型下的性能测试结果表明所提出的学习算法在收敛性和系统性能上的提升。

关键词

MIMO接收机，深度神经网络，数据驱动，元学习

Design of MIMO Receiver Learning Method Based on Meta Learning

Liuhang Wang, Baoyun Wang

School of Communication and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing Jiangsu

Received: May 13th, 2023; accepted: Jun. 7th, 2023; published: Jun. 19th, 2023

Abstract

Data-driven receivers often improve performance at the expense of training. This paper designs the learning method of Multiple Input Multiple Output (MIMO) receiver based on Deep Neural Network (DNN), aiming to reduce the training cost and improve the performance for the receiver. Solve problems such as periodic transmission of pilots that cause the network to be trained frequently, require a large amount of training data, training time, and slow network convergence.

文章引用: 汪流航, 王保云. 基于元学习的 MIMO 接收机学习方法设计[J]. 软件工程与应用, 2023, 12(3): 463-473.
DOI: 10.12677/sea.2023.123046

According to the learning idea of meta-learning, the system model is constructed task-based. A receiver learning algorithm including meta-training, meta-adaptation and testing is proposed. In the process of training, the learning process is optimized, including gradient simplification and parameter inheritance, which improves the convergence speed and generalization ability of the network. For the experimental design, the simulation results of the traditional receiver under different channel models show that the proposed learning algorithm for the receiver improves the convergence and system performance.

Keywords

MIMO Receiver, Deep Neural Network, Data-Driven, Meta Learning

Copyright © 2023 by author(s) and Hans Publishers Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/
Open Access

1. 引言

现代通信系统受制于不断增长的吞吐量需求。为了满足该需求，基站(Base Station, BS)通常配备多天线，同时与多个用户通信，以提高频谱效率[1]。这种多用户多输入多输出(MIMO)系统的主要挑战之一是如何进行符号检测，即实现在 BS 端进行由上行通道传输的多个符号的恢复工作。然而，一些基于深度展开方法提出的接收机通常都假设有高斯噪声的线性信道，其中信道状态信息(Channel State Information, CSI)是提前假设已知的[2]或者是从导频信号[3]中估计出的。因此，尽管机器学习在这些接收机的设计中取得了较好性能的应用，但是这些方法并没有捕捉到机器学习独立于模型的潜力，并且只适用于特定的信道模型和相关的信道设置。

文献[4]设计了一种数据驱动的接收机，从标记数据集中实现已有的软迭代干扰消除符号检测算法[5]。然而该模型中 DNN 构建块的学习是独立的，当其数量很多时，所需的训练时间成本会大幅增加，系统收敛速度慢。这一特性在 MIMO 系统中则会显得尤为突出。近期的一项工作[6]针对含智能反射面(Reconfigurable Intelligent Surface, RIS)辅助的 MIMO 系统，提出了一种联合优化 RIS 和 DNN 接收机网络参数的贝叶斯机器学习框架，然而由于贝叶斯优化中迭代周期的存在和周期性的发送导频的特点会进一步增加网络的训练成本。

针对上述问题，本文提出了一种基于元学习[7] [8]的接收机学习算法。该算法结合现有的接收机结构采用元学习的学习方式，对系统进行基于任务的划分。在任务内和任务间的更新采用独立的数据集，使得网络在训练的过程中网络参数会逐步向初始更优的方向靠拢，在新任务的学习上将更快的适应，并提升系统的性能。

2. 系统模型

在由 RIS 辅助的上行的 MIMO 系统中，考虑为蜂窝网络的单小区，其中装配了 N 个天线的基站为 K 个用户终端服务，RIS 含有 P 个单元。该系统由三个基本部分组成，包括基站，用户终端和 RIS。对系统基于任务的划分后，用户终端与环境(RIS)相关的符号集合可视为独立的任务(Task)，系统模型可简化为图 1。这里假设基站端装配了 DNN 接收机[4]，通过训练 DNN 接收机实现符号检测。在每个时隙 t 中，用户终端发送已知的导频信号，记为 $\boldsymbol{s}_t \triangleq \left[s_{1,t} s_{2,t} \cdots s_{K,t}\right] \in S^{K\times 1}$，其中 $\mathcal{S}$ 表示为离散的星座集，其大小记为 M (符号的种类数)。对于线性高斯信道，基站端接收的符号可表示为[9]：

$$\boldsymbol{y}_t \triangleq \left(\boldsymbol{H}_2\boldsymbol{\Phi}\boldsymbol{H}_1+\boldsymbol{G}\right)\boldsymbol{s}_t+\boldsymbol{n}_t \tag{1}$$

其中 $\boldsymbol{\Phi}\triangleq \mathrm{diag}\left[\mathrm{e}^{j\phi_1},\mathrm{e}^{j\phi_2},\cdots,\mathrm{e}^{j\phi_P}\right]\in\mathcal{C}^{P\times P}$，$\varphi_p\in[0,2\pi]$ 表示相移矩阵且下标 p 代表第 p 个 RIS 单元。$\boldsymbol{H}_1\triangleq\left[\boldsymbol{h}_{1,1}\boldsymbol{h}_{1,2}\cdots\boldsymbol{h}_{1,K}\right]\in\mathcal{C}^{P\times K}$ 表示 RIS 单元与用户之间的无线信道增益矩阵($\boldsymbol{h}_{1,k}\in\mathcal{C}^{P\times 1}$，$k=1,2,\cdots,K$，该向量表示第 k 个用户到 RIS 之间的信道)。$\boldsymbol{H}_2\in\mathcal{C}^{N\times P}$ 表示基站和 RIS 之间的信道。$\boldsymbol{G}\triangleq\left[\boldsymbol{g}_1\boldsymbol{g}_2\cdots\boldsymbol{g}_K\right]\in\mathcal{C}^{N\times K}$ 表示基站和用户终端之间的直接链路信道($\boldsymbol{g}_k\in\mathcal{C}^{P\times 1}$ 表示第 k 个用户到基站的直接链路信道)。此外 $\boldsymbol{n}\in\mathcal{C}^{N\times 1}$ 表示方差为 σ^2 的加性高斯白噪声(Additive White Gaussian Noise, AWGN)向量，通常被建模为均值为零且其协方差矩阵为 $\sigma^2\boldsymbol{I}_N$。

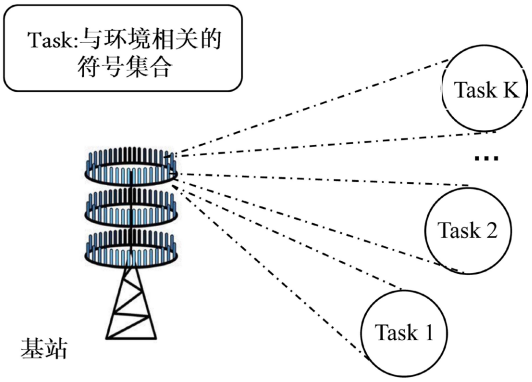

Figure 1. Task-based system model
图 1. 基于任务的系统模型

基站端装配 DNN 接收机结构如图 2，该接收机实现了传统的迭代干扰消除方法，利用深度展开的方式，将原有的算法流程中的部分复杂计算利用 DNN 进行替代，利用深度学习进行建模并扩展为与信道模型无关的实现。

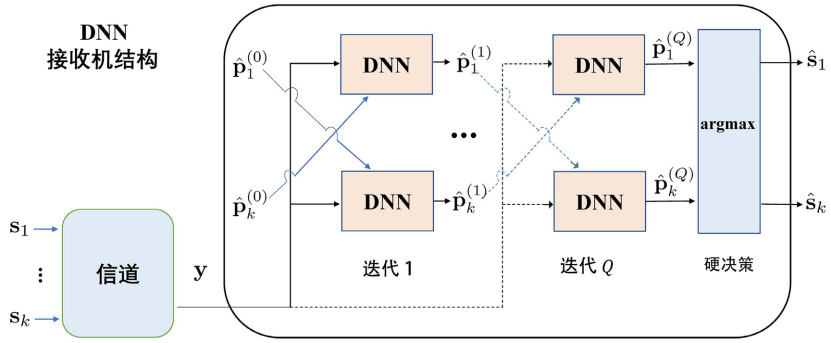

Figure 2. Receiver structure diagram
图 2. 接收机结构图

该接收机采用迭代方式来实现干扰消除，对于第 k 个用户来说，将其他用户传输的符号视干扰符号。符号检测器以迭代进行操作：在每次迭代 $q\in\{1,2,\cdots,Q\}\triangleq\mathcal{Q}$ 中，对于给定信道输出 $\boldsymbol{y}$，关于用户符号 $\boldsymbol{s}_k$ 的条件概率分布的估计用 $\boldsymbol{p}_k^{(q)}$ 表示，对每个用户 $k\in\mathcal{K}\triangleq\{1,2,\cdots,K\}$ 来说，使用前一次迭代中得到的干扰符号 $\{\boldsymbol{s}_l\}_{l\neq k}$ 的估计作为下一次迭代的输入。这里，用 $\boldsymbol{s}_k$ 表示为在每个时隙中由第 k 个用户传输的符号。符号间的干扰是通过连续的迭代进行消除，其中最后一次迭代的 DNN 输出用于硬判决的方式译码。具体的

DNN 结构如图 3 所示的全连接型神经网络，其输入包括信道的输出(复数需要进行实部和虚部的拼接处理)和上一次迭代的 DNN 输出，优化器采用 Adam。

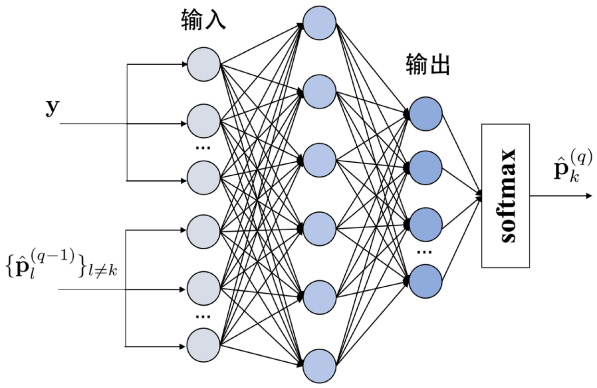

Figure 3. Task-based system model
图 3. 基于任务的系统模型

对于 N_{tr} 对信道的输入和相应输出的数据集 $\left\{\tilde{\boldsymbol{s}}_j,\tilde{\boldsymbol{y}}_j\right\}_{j=1}^{N_{\mathrm{tr}}}$，将 $\hat{\boldsymbol{p}}_k^{(q)}\left(\boldsymbol{y},\left\{\hat{\boldsymbol{p}}_l^{(q-1)}\right\}_{l\neq k},\alpha;\theta_k^{(q)}\right)$ 表示为对应的 $\hat{\boldsymbol{p}}_k^{(q)}$ 关于输入 $\alpha\in\mathcal{S}$、第 k 个 DNN 在迭代次数 q 时的网络参数 $\theta_k^{(q)}$ 以及上一次迭代过程中的输入 $\left(\boldsymbol{y},\left\{\hat{\boldsymbol{p}}_l^{(q-1)}\right\}_{l\neq k}\right)$ 的表达式，则交叉熵损失可以表示为：

$$\mathcal{L}\left(\theta_k^{(q)}\right)=\frac{1}{N_{\mathrm{tr}}}\sum_{j=1}^{N_{\mathrm{tr}}}-\log_{\mathrm{e}}\hat{\boldsymbol{p}}_k^{(q)}\left(\tilde{\boldsymbol{y}}_j,\left\{\hat{\boldsymbol{p}}_{j,l}^{(q-1)}\right\}_{l\neq k},\left(\tilde{\boldsymbol{s}}_j\right)_k;\theta_k^{(q)}\right)\tag{2}$$

其中，$\left\{\hat{\boldsymbol{p}}_{j,l}^{(q-1)}\right\}_{l\neq k}$ 表示与 $\tilde{\boldsymbol{y}}_j$ 相关的前一次迭代的符号条件概率。

3. 学习过程与算法设计

3.1. 学习过程

系统基于任务的划分后，第 k 个 Task 用 $\{\boldsymbol{s}_k,\boldsymbol{y}_k\}$ 表示其数据集。接收机的学习过程可分为三个阶段。分别是元训练、元适应、和测试阶段。下面首先介绍一些基础定义和数据集的具体划分，然后介绍学习的各个阶段。

1) 基础定义与数据划分

对于第 k 个 Task 数据集，需要为元训练阶段提供两个数据集，即支持数据集(Support Set)和查询数据集(Query Set)，分别表示为：$\mathbb{D}_{\mathrm{sup}}(k)$ 和 $\mathbb{D}_{\mathrm{que}}(k)$。数据集满足 $\mathbb{D}_{\mathrm{sup}}\cap\mathbb{D}_{\mathrm{que}}=\varnothing$，以提升网络的泛化能力。对于元学习来说，最终目的是寻找到适用于各任务的初始元网络参数 ω_{meta}。在元网络参数更新完毕后需进行元测试过程，为避免与机器学习中数据测试相混淆，用元适应来表示这一过程，其含义为：由元参数出发适应到各个任务的最优参数。元适应阶段最后将得到各任务训练完毕的网络参数，然后进行性能的评估验证，即进行测试过程。适应集(Adaption Set)和测试集(Test Set)分别表示为 $\mathbb{D}_{\mathrm{ad}}$ 和 $\mathbb{D}_{\mathrm{te}}$，其具体划分与元训练阶段的数据集相同，且来自与同一分布(即数据集来自于各个用户的符号集合)。

2) 元训练过程

在元训练阶段，元学习算法的目标是学习一个能够有效适应新任务的网络初始化参数 ω。DNN 的参数初始是由随机生成的，然后通过两个迭代过程进行更新，即任务内部更新和任务外部更新。在任务内

部更新时，对于任务 k 来说，其独有的关于 DNN 的网络参数用 θ_k 表示。内部更新目标是通过在数据集 $\mathbb{D}_{\text{sup}}$ 上最小化损失函数(遵循式 2)，用 $\mathcal{L}_{\mathbb{D}_{\text{sup}}}(\theta_k)$ 表示任务 k 的损失，通过内部迭代更新参数 θ_k。其中任务 k 的网络参数 θ_k 初始化为元网络参数 ω_{meta}，然后通过下式进行梯度下降以更新内部网络参数：

$$\theta_k \leftarrow \theta_k - \alpha \nabla_{\theta_k} \mathcal{L}_{\mathbb{D}_{\text{sup}}}(\theta_k) \tag{3}$$

内部更新的学习率用 α 表示。在任务外部更新时，所依赖的数据集为 $\mathbb{D}_{\text{que}}$，进行的是元网络参数 ω 的更新。统计各个任务基于查询集的总损失，可表示为：$\sum_{k\in\mathcal{K}} \mathcal{L}_{\mathbb{D}_{\text{que}}}(\theta_k)$。其中 $\mathcal{K}$ 表示任务集合(不同用户的符号集合)。由此得到外部更新的方式为：

$$\omega \leftarrow \omega - \beta \nabla_{\omega} \sum_{k\in\mathcal{K}} \mathcal{L}_{\mathbb{D}_{\text{que}}}(\theta_k) \tag{4}$$

外部更新的学习率设置为 β。式 4 的更新涉及高阶导数的计算，为降低计算成本，进行梯度简化，现考虑内部网络进行了 m 次梯度下降(为简单起见，忽略损失对下标的依赖，且任务内部损失用 $\mathcal{L}^{(0)}$ 表示，外部损失用 $\nabla_{\omega}\mathcal{L}(\omega)$ 表示)，网络参数变化如下式：

$$\begin{aligned}
&\theta_0 = \omega \\
&\theta_1 = \theta_0 - \alpha \nabla_{\theta} \mathcal{L}^{(0)}(\theta_0) \\
&\theta_2 = \theta_1 - \alpha \nabla_{\theta} \mathcal{L}^{(0)}(\theta_1) \\
&\quad \vdots \\
&\theta_m = \theta_{m-1} - \alpha \nabla_{\theta} \mathcal{L}^{(0)}(\theta_{m-1})
\end{aligned} \tag{5}$$

然后对外部损失 $\nabla_{\omega}\mathcal{L}(\omega)$ 运用链式法则展开，得到：

$$\begin{aligned}
\nabla_{\omega}\mathcal{L}(\omega) &= \nabla_{\theta}\mathcal{L}(\theta_m) \\
&= \nabla_{\theta_m}\mathcal{L}(\theta_m) \cdot \left(\nabla_{\theta_{m-1}}\theta_m\right) \cdots \left(\nabla_{\theta_0}\theta_1\right) \cdot \left(\nabla_{\theta}\theta_0\right) \\
&= \nabla_{\theta_m}\mathcal{L}(\theta_m) \cdot \left(\prod_{i=1}^{m} \nabla_{\theta_{i-1}}\theta_i\right) \cdot I \\
&= \nabla_{\theta_m}\mathcal{L}(\theta_m) \cdot \prod_{i=1}^{m} \nabla_{\theta_{i-1}} \left(\theta_{i-1} - \alpha \nabla_{\theta} \mathcal{L}^{(0)}(\theta_{i-1})\right) \\
&= \nabla_{\theta_m}\mathcal{L}(\theta_m) \cdot \prod_{i=1}^{m} \left(I - \alpha \nabla_{\theta_{i-1}} \left(\nabla_{\theta} \mathcal{L}^{(0)}(\theta_{i-1})\right)\right)
\end{aligned} \tag{6}$$

由此可见，元网络参数 ω 的更新涉及二阶求导。采用一阶导数来简化二阶导数的计算，用来降低神经网络的计算成本。文献[8]中指出大多数二阶梯度的计算中，其结果近似于零，因此可忽略二阶项，用一阶导数代替，即外部更新中关于 ω 的梯度可近似为：$\nabla_{\omega}\mathcal{L}(\omega) \approx \nabla_{\theta_m}\mathcal{L}(\theta_m)$。因此式 4 中的梯度可近似为 $\nabla_{\theta_m} \sum_{k\in\mathcal{K}} \mathcal{L}_{\mathbb{D}_{\text{que}}}(\theta_k)$。梯度简化操作在性能上略微降低但同时提升了训练速度。

3) 元适应和测试过程

在元适应阶段，元学习算法旨在对适应集 $\mathbb{D}_{\text{ad}}$ 上的网络进行微调，使网络能够尽可能准确的在测试集 $\mathbb{D}_{\text{te}}$ 取得良好的性能。对于第 k 个任务，将元训练阶段学习到的元网络参数 ω 赋值给任务 k 独有的 DNN 网络参数，然后通过梯度下降在适应集上进行参数微调。该步骤的梯度更新由下式决定：

$$\theta_k \leftarrow \theta_k - \gamma \nabla_{\theta_k} \mathcal{L}_{\mathbb{D}_{\text{ad}}}(\theta_k) \tag{7}$$

其中适应阶段的学习率用 γ 表示。当适应阶段完成后，便获得任务 k 的最优参数 θ_k^*。最后使用测试集 $\mathbb{D}_{\text{te}}$ 评估 DNN 网络性能，即通过误比特率(Bit Error Rate, BER)的大小判断该 DNN 进行符号检测的能力。

3.2. 算法设计与分析

Table 1. Receiver training algorithm based on metalearning
表 1. 基于元学习的接收机训练算法

算法 1：基于元学习的接收机训练算法
输入：初始符号数据集，任务数，迭代次数，学习率
输出：适应的 DNN 网络参数 $\theta=\left\{\theta_k^{(q)}\right\}_{k,q=1}^{K,Q}$，元参数 ω，误码率 BER
1：For $t=1,2,\cdots$ do
2：生成关于任务 k 的数据集 $\mathbb{D}_{\text{sup}}(k)$ 和 $\mathbb{D}_{\text{que}}(k)$
3：**For** $k\in\mathcal{K}$ **do**
4：设置 $\theta_k\leftarrow\theta$
5：**For** $n=1,\cdots,N_{\text{in}}$ **do**
6：由式 3 进行内部更新
7：**End**
8：将 $\left\{\tilde{\boldsymbol{y}}_j,\left\{\hat{\boldsymbol{p}}_{j,l}^{(t-1)}\right\}_{l\neq k}\right\}_{j=1}^{N_{\text{sup}},N_{\text{que}}}$ 输入已训练的网络并输出 $\left\{\hat{\boldsymbol{p}}_{j,k}^{(t)}\right\}_{j=1}^{N_{\text{sup}},N_{\text{que}}}$
9：**End**
10：由式 4 进行外部更新
11：**End**
12：获取元参数：$\omega_{\text{meta}}\leftarrow\theta$
13：**For** $q=1,\cdots,Q$ **do**
14：生成适应集 $\left\{\mathbb{D}_{\text{ad}}(k)\right\}_{k=1}^{K}$
15：**For** $k\in\mathcal{K}$ **do**
16：**If** $q=1$ **then** 设置 DNN 参数 $\theta_k^{(q)}\leftarrow\omega_{\text{meta}}$
17：**Else** 设置 DNN 参数 $\theta_k^{(q)}\leftarrow\theta_k^{(q-1)}$
18：由式 7 进行 N_{ad} 步适应更新
19：将 $\left\{\tilde{\boldsymbol{y}}_j,\left\{\hat{\boldsymbol{p}}_{j,l}^{(q-1)}\right\}_{l\neq k}\right\}_{j=1}^{N_{\text{ad}}}$ 输入到已训练的网络并输出 $\left\{\hat{\boldsymbol{p}}_{j,k}^{(q)}\right\}_{j=1}^{N_{\text{ad}}}$
20：**End**
21：**End**
22：输入 $\mathbb{D}_{\text{te}}$ 进行测试，输出 BER，元网络参数 ω_{meta} 和已训练的网络参数 θ_k

结合数据集的划分、各学习过程和 DNN 接收机的结构，提出的基于元学习的三阶段的学习算法如表 1。执行之前需要进行初始化，即初始随机生成网络参数 θ 适用于各任务并设置各任务初始的输入条件概率为均匀分布即 $\left\{\hat{\boldsymbol{p}}_k^{(0)}\right\}_{k=1}^{K}=M^{-1}$ 且 $\hat{\boldsymbol{p}}_{j,k}^{(0)}=\hat{\boldsymbol{p}}_k^{(0)}$。算法主要包括内部更新，外部更新和适应阶段。在步骤 3 中，所有的任务是并行执行的。在步骤 8 中数据集 $\left\{\boldsymbol{y},\boldsymbol{p}\right\}_{j=1}^{N_{\text{sup}},N_{\text{que}}}$ (简写形式)表示由内部更新的网络得到的组合数据，N_{sup} 和 N_{que} 分别表示支持集和查询集中数据对的数目。步骤 1~12 中由元训练阶段更新的元网络参数 ω_{meta} 将作为步骤 13~21 (适应阶段)的初始网络参数，由此出发在适应集上得到该任务的最优参数。

步骤 16~17 称为“继承”。该步骤的目的是在接收机训练迭代中，第 q 次迭代产生的网络参数可作

为第 $q+1$ 次迭代的初始参数，以便更快的适应网络，这是因为该接收机结构是基于迭代的干扰消除方法，第 q 次迭代所得到的网络可以认为是含有干扰消除的相关知识，因而可被下一次迭代的网络利用。元训练阶段中其内部更新的步数通常设置的不高，原因是元更新的目的更看重初始化的能力，而非让所有任务均取得最优(容易导致过拟合)。下面将以元训练阶段外部更新的损失(即 $\mathcal{L}_{\mathbb{D}_{\text{que}}}$)解释元学习的泛化能力：

$$\begin{aligned}\mathcal{L}_{\mathbb{D}_{\text{que}}} &= \sum_{k\in\mathcal{K}} \mathcal{L}_{\mathbb{D}_{\text{que}}(k)}\left(\theta_k^{(t)}\right)\\ &= \sum_{k\in\mathcal{K}} \mathcal{L}_{\mathbb{D}_{\text{que}}(k)}\left(\omega^{(t-1)} - \alpha\nabla\mathcal{L}_{\mathbb{D}_{\text{que}}(k)}\left(\omega^{(t-1)}\right)\right)\\ &\approx \sum_{k\in\mathcal{K}} \mathcal{L}_{\mathbb{D}_{\text{que}}(k)}\left(\omega^{(t-1)}\right) - \sum_{k\in\mathcal{K}} \alpha\nabla\mathcal{L}_{\mathbb{D}_{\text{que}}(k)}\left(\omega^{(t-1)}\right)\nabla\mathcal{L}_{\mathbb{D}_{\text{que}}(k)}\left(\omega^{(t-1)}\right)\end{aligned} \tag{8}$$

这里用上标(t)表示第 t 次外部更新。为便于分析，设置内部更新的迭代次数为 1 (即 $N_{\text{in}}=1$)。外部更新损失进行一阶泰勒展开，得到式 8。该式的第一项为查询集的总损失，第二项为支持集和查询集上梯度的负内积和。当两个梯度的方向越接近时，则负内积越小，因而总损失越小。这说明在元训练阶段外部更新的过程中，试图最大化查询集与支持集之间梯度的相似性。这增强了查询集与支持集之间的泛化能力，因此当更新的元网络参数应用于新任务时，将会更快的适应。

4. 实验设计与分析

本节首先验证上述基于元学习的接收机(以下将简称为 MetaDNN)算法，首先对该接收机算法进行测试，而后在不同信道模型下和不同的接收机进行性能比较测试。

4.1. 级联信道模型下接收机性能

当信道模型为级联模型(服从式 1 的相关建模)并设置用户、RIS 和基站的坐标分别为[0, 0]，[30, 30]和[60, 0]，单位(m)。其中各信道 H_1, H_2, G 的衰落因子分别设置为 0.7，0.8 和 0.9，信噪比定义为 $1/\sigma^2$ 。将 $\boldsymbol{H}_1$ 和 $\boldsymbol{G}$ 建模为瑞利信道，其每一列分别为 $\boldsymbol{h}_{1,k}=\gamma\tilde{\boldsymbol{h}}_{1,k}$ 和 $\boldsymbol{g}_k=\gamma\tilde{\boldsymbol{g}}_k$ ，其中 γ 为路径损耗因子 $\tilde{\boldsymbol{h}}_{1,k}, \gamma\tilde{\boldsymbol{g}}_k \sim \mathcal{CN}\left(0, \boldsymbol{I}_P\right)$ (每一列均由复高斯分布随机产生)。信道矩阵 $\boldsymbol{H}_2$ 可建模为莱斯信道：

$$\boldsymbol{H}_2 = \beta\left(\sqrt{\frac{\kappa}{1+\kappa}}\tilde{\boldsymbol{H}}_2^{\text{LOS}} + \sqrt{\frac{1}{1+\kappa}}\tilde{\boldsymbol{H}}_2^{\text{NLOS}}\right) \tag{9}$$

其中 κ 和 β 分别表示莱斯因子(10 dB)和路径损耗。上标“LOS”和“NLOS”表示信道的视距和非视距成分，且该信道分量均由标准的高斯信道矩阵随机产生。整个系统为 10 × 10 的 MIMO，星座集采用 QPSK，RIS 单元数量 P 为 256。仿真环境为 Python，测试框架为 Pytorch。算法 1 中各主要参数设置如表 2 所示：

Table 2. Parameter settings
表 2. 参数设置

参数名	设置值
迭代数 Q 和任务数 K	4，10
学习率 α, β, γ	1e−3，1e−2，5e−3
训练集 $\mathbb{D}_{\text{sup}}$ 和 $\mathbb{D}_{\text{que}}$ 样本量	1000，1000
适应集 $\mathbb{D}_{\text{ad}}$ 和测试集 $\mathbb{D}_{\text{te}}$ 样本量	500，20,000
内部更新步数 N_{in}	3

该算法的性能如图 4，信噪比(Signal Noise Ratio, SNR)为 11 dB 的不同方案的性能比较。由随机初始网络参数方案中有无继承可以得出，继承的使用将加快网络收敛速度，在初始随随机的方案中，网络收敛从 500 步开始，而使用继承后收敛从 250 步开始。从元学习的对比也可发现，在元学习的方案中收敛最初从 50 步开始，相较于随机初始方案，收敛的速度得到很大的提升。然而继承的使用在性能上的提升不明显，当迭代的步数足够大时，无论有无继承，性能收敛值 BER 均接近 10^{-3}。综合比较可以得出元学习算法在网络收敛能力上提升较大，且在系统性能上有一定提升。

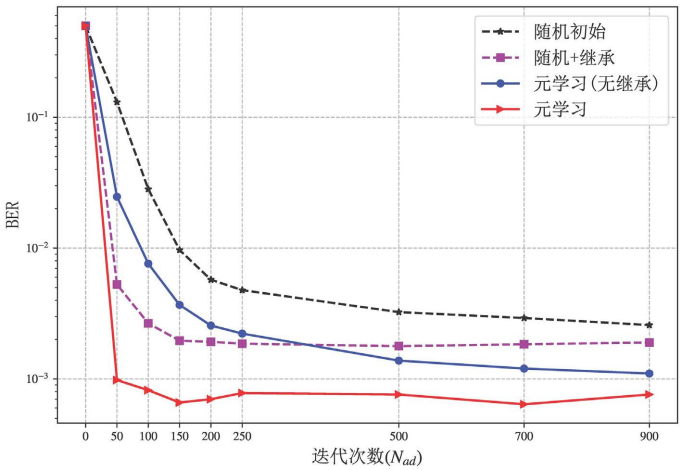

Figure 4. Comparison of performance for different simulations
图 4. 不同方案性能比较

接下来考虑基于元学习算法的接收机和其他传统接收机的性能比较。即比较基准包含迫零检测器(Zero Forcing, ZF)、最小均方准则检测器(Minimum Mean Square Error, MMSE) [10]和球体解码器(Sphere Decoder, SD) [11]。需要注意的是，接收机 ZF，MMSE 和 SD 作为对比方案时，具有完整的 CSI，与译码规则有关，且 SD 是一种次优的解码器，是最大似然译码方法的高效实现。基于 DNN 的接收机不需要 CSI，用信道输入输出的符号对来进行数据驱动。

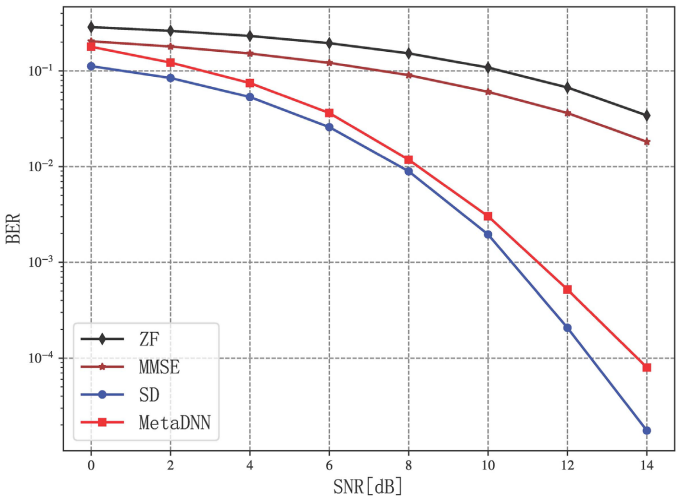

Figure 5. Comparison of different receiver performance
图 5. 不同接收机性能比较

如图 5 所示，传统的接收机 ZF 和 MMSE 在具有 CSI 的情况下，其性能在 SNR 较小时均较差，然而基于元学习的接收机性能虽然比次优的 SD 性能稍差，但在 SNR 相同时其性能差异不大，例如同时取得 BER 为 10^{-3} 时，其所需的 SNR 差异接近 0.6 dB。当 SNR 逐渐增大时，其性能差异逐渐显现。这是由于当 SNR 较大时，噪声的影响逐渐减小，相较于利用 CSI 的方式，由数据驱动的学习方式对噪声变化不敏感，因而，改善的幅度不如利用 CSI 的方式。

4.2. 非级联信道模型下接收机性能

本节中考虑的是一个更复杂的信道模型，并验证 MetaDNN 接收机的性能。与级联模型不同，这里采用了一种更复杂的物理衰落信道模型 PhysFad [12]，它是一种基于物理层的由 RIS 参数化辅助且衰落可调的端到端模型。由 RIS 辅助的无线信道通常由收发天线、RIS 单元和散射环境三部分组成。PhysFad 将这三个实体建模为具有特殊属性的偶极子或一组偶极子的集合，根据这些偶极子组合成一个信道模型并生成端到端的信道矩阵。值得注意的是，该建模方法用于二维(2D)场景，系统由偶极子描述，该偶极子用一组参数 $\left(\boldsymbol{r}_i, f_{\mathrm{res},i}, \chi_i, \Gamma_i\right)$ 表征。本文不涉及该参数的专业物理学含义的解释，关于详细参数的物理含义参考原文献。

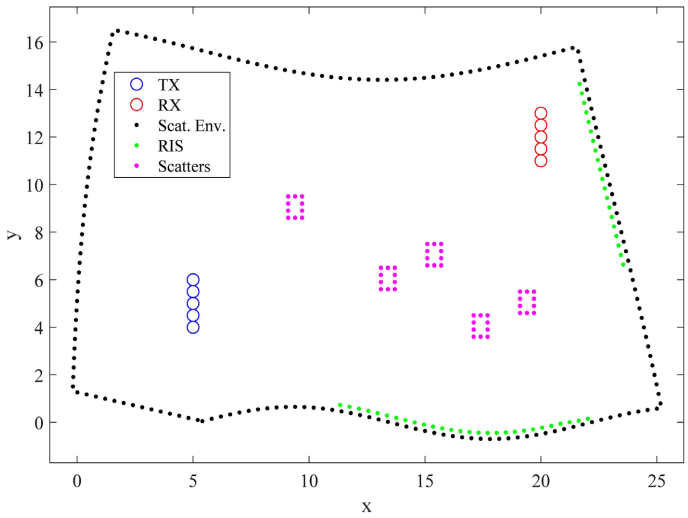

Figure 6. 2D system modeling based on PhysFad
图 6. 基于 PhysFad 的 2D 系统建模

图 6 是基于文献中的建模方法所设计的 2D FhysFad 模型简图(单位：m)。通信系统中的实体均被建模为偶极子，系统包括发送天线(TX)、接收天线(RX)、散射环境(Scat.Env)、散射体(Scatters)和 RIS。该图为一个小型的 5 × 5 MIMO 系统(具有金属外壳的内部环境，RIS 分布在墙壁两侧)。在该模型中对系统中的各个实体进行了固定，即采用文献中提到的默认参数，产生莱斯因子为 10 dB 的端到端的莱斯分布信道矩阵。由此得到的仿真结果如图 7，由此可见非级联模型下的性能趋势下和级联模型的仿真结果的趋势基本一致。然而，在信噪比均定义为的情况下，由于信道环境的改变，需要更大的信噪比才能获取较好的性能增益。

观察到当 SNR 在 12~18 dB 时 MetaDNN 和 SD 的性能接近，然后性能差异开始扩大，这是基于 SD 接收机拥有完美 CSI，而 MetaDNN 通过数据驱动的方式，当无 CSI 可用时，MetaDNN 在少量样本的情况下依然可以达到较好的性能。

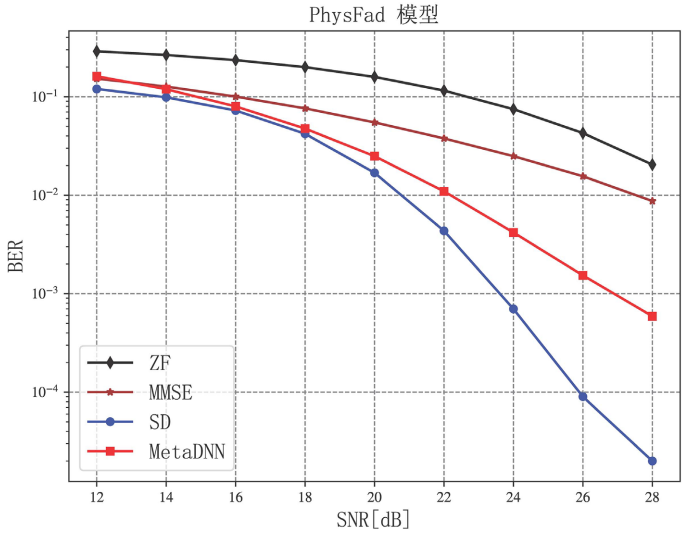

Figure 7. Comparison of receiver performance for the PhysFad model
图 7. PhysFad 模型下各接收机性能比较

5. 结论

本文提出一种基于元学习的 DNN 接收机(MetaDNN)学习算法，该算法在通过多阶段的学习得到适用于新任务即符号集合的初始网络参数，能够快速适应新的任务，并以少量的迭代次数使得网络快速收敛并同时提升系统性能。仿真结果表明相较于传统依赖 CSI 的检测器 ZF、MMSE，MetaDNN 在无需 CSI 的情况下依然有更好的性能且性能接近次优的 SD 检测器。在级联和非级联模型中均取得了相似的结果表明该算法对于不同信道模型的有效性。

参考文献

[1] Marzetta, T.L. (2010) Noncooperative Cellular Wireless with Unlimited Numbers of Base Station Antennas. *IEEE Transactions on Wireless Communications*, **9**, 3590-3600. https://doi.org/10.1109/TWC.2010.092810.091092

[2] Liu, L., Chi, Y., Yuen, C., *et al*. (2019) Capacity-Achieving MIMO-NOMA: Iterative LMMSE Detection. *IEEE Transactions on Signal Processing*, **67**, 1758-1773. https://doi.org/10.1109/TSP.2019.2896242

[3] He, H., Wen, C.K., Jin, S., *et al*. (2018) A Model-Driven Deep Learning Network for MIMO Detection. *Proceedings of* 2018 *IEEE Global Conference on Signal and Information Processing*, Anaheim, 26-29 November 2018, 584-588. https://doi.org/10.1109/GlobalSIP.2018.8646357

[4] Shlezinger, N., Fu, R. and Eldar Y.C. (2020) DeepSIC: Deep Soft Interference Cancellation for Multiuser MIMO Detection. *IEEE Transactions on Wireless Communications*, **20**, 1349-1362. https://doi.org/10.1109/TWC.2020.3032663

[5] Choi, W.J., Cheong, K.W. and Cioffi, J.M. (2000) Iterative Soft Interference Cancellation for Multiple Antenna Systems. *Proceedings of* 2020 *IEEE Wireless Communications and Networking Conference Record*, Chicago, 23-28 September 2000, 304-309. https://doi.org/10.1109/WCNC.2000.904647

[6] Wang, L., Shlezinger, N., Alexandropoulos, G.C., Zhang, H., Wang, B. and Eldar, Y.C. (2021) Jointly Learned Symbol Detection and Signal Reflection in RIS-Aided multi-User MIMO Systems. *Proceedings of* 55*th Asilomar Conference on Signals*, *Systems and Computers*, Pacific Grove, 31 October-3 November 2021, 715-721. https://doi.org/10.1109/IEEECONF53345.2021.9723108

[7] Hospedales, T., Antoniou, A., Micaelli, P. and Storkey, A. (2021) Meta-Learning in Neural Networks: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **44**, 5149-5169. https://doi.org/10.1109/TPAMI.2021.3079209

[8] Finn, C., Abbeel, P. and Levine, S. (2017) Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *Proceedings of the* 34*th International Conference on Machine Learning*, Sydney, 6-11 August 2017, 1126-1135.

[9] Huang, C., Zappone, A., Alexandropoulos, G.C., Debbah, M. and Yuen, C. (2019) Reconfigurable Intelligent Surfaces

for Energy Efficiency in Wireless Communication. *IEEE Transactions on Wireless Communications*, **18**, 4157-4170. https://doi.org/10.1109/TWC.2019.2922609

[10] Riadi, A., Boulouird, M. and Hassani, M. (2021) ZF and MMSE Detectors Performances of a Massive MIMO System Combined with OFDM and M-QAM Modulation. *Wireless Personal Communications*, **116**, 3261-3276. https://doi.org/10.1007/s11277-020-07848-4

[11] Ghasemmehdi, A. and Agrell, E. (2011) Faster Recursions in Sphere Decoding. *IEEE Transactions on Information Theory*, **57**, 3530-3536. https://doi.org/10.1109/TIT.2011.2143830

[12] Faqiri, R., Saigre-Tardif, C., Alexandropoulos, G.C., Shlezinger, N., Imani, M.F. and del Hougne, P. (2023) PhysFad: Physics-Based End-to-End Channel Modeling of RIS-Parametrized Environments with Adjustable Fading. *IEEE Transactions on Wireless Communications*, **22**, 580-595. https://doi.org/10.1109/TWC.2022.3196834