

引用格式:许晨,杨少石,谭景升.一种基于元学习的自适应调制编码策略[J].中国传媒大学学报(自然科学版),2024,31(03):42-48.  
文章编号:1673-4793(2024)03-0042-07

# 一种基于元学习的自适应调制编码策略

许晨<sup>1,2</sup>,杨少石<sup>1,2\*</sup>,谭景升<sup>1,2</sup>

(1.北京邮电大学信息与通信工程学院,北京100876;2.泛网无线通信教育部重点实验室,北京100876)

**摘要:**针对现有基于深度学习的自适应调制编码算法在信道环境改变时出现的模型泛化能力下降的问题,提出了一种基于元学习的自适应调制编码策略。该方法利用元学习算法快速适应新任务的优势,使得模型仅需通过新环境下的少量样本微调就能获得良好的性能。仿真结果和讨论都表明,本文所提算法比基线算法在性能上表现更为优越。

**关键词:**元学习;自适应调制编码;泛化能力

**中图分类号:**TN92 **文献标识码:**A

## An adaptive modulation and coding strategy based on Meta-learning

XU Chen<sup>1,2</sup>, YANG Shaoshi<sup>1,2\*</sup>, TAN Jingsheng<sup>1,2</sup>

(1.School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China;2. Key Laboratory of Universal Wireless Communications, Ministry of Education, Beijing 100876, China)

**Abstract:** A meta-learning-driven adaptive modulation coding strategy was proposed for dealing with the model generalization capability degradation problem encountered by the existing deep learning algorithms when the channel environment was changed. The proposed approach employed the Model-Agnostic Meta-Learning (MAML) algorithm to predict the modulation and coding schemes based on channel characteristics. Initially, two neural network models were proposed and trained. Subsequently, a small number of samples from new channel scenarios were used to fine-tune the trained model parameters, thus enabling rapid adaptation to new environments. Both simulation results and discussions demonstrate that the proposed algorithm outperforms the baseline algorithms in term of throughput performance and model generalization capability.

**Keywords:** Meta-learning; adaptive modulation and coding; generalization capability

## 1 引言

随着信息通信产业的发展,通信业务和信号传播环境的多样性对通信质量提出了更高的要求。面向低延时和高频谱效率的技术在5G、5G-A、6G等移动通信系统,以及物联网、下一代Wi-Fi等系统中将扮演

越来越重要的角色。自适应调制编码技术基于信道状态信息动态调整发送端的调制编码方案,可以在保证系统可靠性的前提下最大化系统容量。另一方面,随着近年来人工智能的发展,机器学习技术在自适应调制编码技术领域的应用也得到了广泛研究。

Zhang等<sup>[1]</sup>提出将多实体贝叶斯网络(Multi-

基金项目:北京市自然科学基金重点项目(Z220004);北京市科委新一代信息通信技术创新项目(Z221100007722036)

作者简介(\*为通讯作者):许晨(1998-),女,硕士研究生,主要从事基于机器学习的自适应调制编码技术研究。E-mail:chenx@bupt.edu.cn;杨少石(1983-),男,教授,博士生导师,主要从事B5G/6G和分布式通信-感知-计算-智能融合理论与技术研究。E-mail:shaoshi.yang@bupt.edu.cn;谭景升(1999-),男,博士研究生,主要从事无线分布式机器学习理论与技术研究。Email:jingsheng.tan@bupt.edu.cn

Entity Bayesian Network, MEBN)学习作为一种新的自适应调制编码框架,该框架在选择调制编码方案(Modulation and Coding Scheme, MCS)方面具有更高的有效性。Daniels等<sup>[2]</sup>推荐利用KNN(K-Nearest Neighbors, KNN)算法,来建立具体的信道和MCS之间的对应关系。同时, Daniels等<sup>[3]</sup>进一步提出了一种基于信噪比排序的降维策略,以应对在频率选择性衰落信道中样本维度增加的问题。随着神经网络的发展,深度学习网络表现出比传统机器学习算法更强大的特征提取和曲线拟合能力,因此许多人基于深度学习算法在自适应调制编码领域展开研究。Yigit等<sup>[4]</sup>提出了一种前馈神经网络(Feedforward Neural Network, FNN)框架作为正交频分复用(Orthogonal Frequency Division Multiplexing, OFDM)系统中基于自适应调制编码的链路自适应机器学习技术。为了进一步地提高OFDM系统在衰落信道下的吞吐量, Dong等<sup>[5]</sup>通过机器学习算法提出了多输入多输出(Multiple-Input Multiple-Output, MIMO)系统中的链路自适应方案,以最大限度地提高频谱效率。Tato等<sup>[6]</sup>利用多层前馈神经网络进行信噪比的估计,并详细地探讨了神经网络各参数的影响。Hussien等<sup>[7]</sup>采用卷积神经网络(Convolutional Neural Network, CNN)模型并设计了

一个简单但功能强大的损失函数,减少了由于较多的MCS分类错误而导致的重传次数。Elwekeil等<sup>[8]</sup>讨论了不同机器学习技术对自适应调制编码性能的影响,考虑了MIMO-OFDM接收机在实际应用中的非理想特性,将估计的信道状态信息和噪声标准差作为CNN的输入特征,来预测合适的调制和编码方案。

现有基于深度学习的自适应调制编码方法,存在信道场景变化时,已训练模型与信道场景不一致导致模型泛化能力下降的问题。因此本文提出一种基于元学习的自适应调制编码策略,采用模型无关元学习(Model-Agnostic Meta-Learning, MAML)方法,根据信道特征预测MCS。首先提出基于CNN和FNN的元学习模型,并通过训练获得神经网络初始化参数,然后在新的信道场景下采集数据并微调网络,从而快速适应新的信道场景。仿真结果表明本文所提算法相比现有基于深度学习的方法提高了信道场景变化时的传输吞吐量性能以及模型泛化能力。

## 2 系统模型

在本文中,考虑一个基于IEEE 802.11ah标准的单入单出(Single-Input Single-Output, SISO)OFDM系统模型,如图1所示。

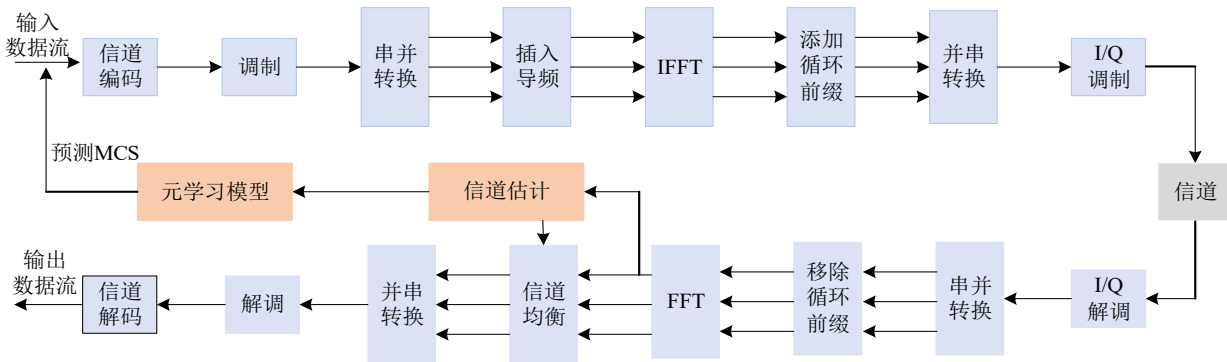


图1 系统模型

在发射端,表示信息数据的比特流经过信道编码、符号调制和串并变换得到待传输符号向量,在待传输符号向量中插入导频,然后对其进行逆快速傅里叶变换(Inverse Fast Fourier Transform, IFFT),并添加循环前缀得到OFDM时域信号。之后对OFDM时域信号进行并串变换和I/Q调制,得到面向信道传输的发射信号。在接收端,对信号的处理可视为发送端的一个逆过程。接收端收到信号以后,首先对信号进行I/Q解调,然后进行串并变换,去除循环前缀之后的OFDM符号通过快速傅里叶变换(Fast Fourier Trans-

form, FFT),获得相应的频域信号,然后基于导频进行信道估计,并将估计后的结果作为信道特征信息输入给已训练的元学习模型,得到预测的MCS,同时得到的频域信号和信道特征信息输入信道均衡器,得到已发送符号向量,将得到的符号向量进行并串变换,并依次进行符号解调和信道解码,最终得到解码的二进制数据。

假设一个待发送的OFDM帧 $\mathbf{X}$ 的大小是一个 $N_s \times N_c$ 的矩阵,信道传输矩阵是 $\mathbf{H}$ , $\mathbf{N}$ 为加性高斯白噪声,最终在接收端获得的OFDM帧表示为 $\mathbf{Y}$ 。对于

第  $n$  个 OFDM 符号的第  $m$  个子载波,输入输出关系可以表示为:

$$Y[m,n] = H[m,n]X[m,n] + N[m,n] \quad (1)$$

其中  $m \in \{1,2,\dots,N_s\}, n \in \{1,2,\dots,N_c\}, H[m,n]$  是  $\mathbf{H} \in \mathbb{C}^{N_s \times N_c}$  中的元素,表示第  $n$  个 OFDM 符号在第  $m$  个子载波上的频域复信道增益。为了估计信道,频域信道响应表示为  $\mathbf{H} = \{\mathbf{h}[1], \mathbf{h}[2], \dots, \mathbf{h}[N_c]\}$  其中  $\mathbf{h}[n]$  是第  $n$  个 OFDM 符号对应的信道频率响应。

### 3 基于元学习的自适应调制编码算法

#### 3.1 MAML 方法的基本原理

MAML 是元学习领域中的一种杰出实现方法。其核心理念是通过对神经网络初始化参数的学习,使模型具备迅速适应各种新任务的能力。通过从大量的任务中学习来寻找优化的初始化参数,这样模型在面对新任务时将不需要从随机初始化参数开始学习,而是在先前学习到的最优参数上,通过少量梯度更新,便能使模型收敛。

基于 MAML 的神经网络初始化参数的学习过程由两个学习器共同辅助完成。两个学习器分别称为元学习器和基学习器。相应地, MAML 分为两层优化,内层优化负责更新基学习器的参数,外层优化负责更新元学习器的参数。面对新任务时,通过对参数的微调,神经网络即可实现在新环境下高效的性能表现。

MAML 算法<sup>[9]</sup>的参数优化方式如图 2 所示。

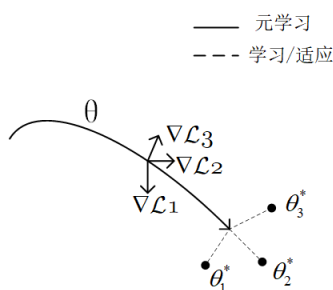


图2 MAML 参数优化方式

MAML 定义一个元学习模型  $a = f_{\theta}(x)$ , 其中  $\theta$  是模型参数。在模型训练过程中,元学习沿不同方向根据梯度下降法进行搜索,获得具有一定先验知识的神经网络初始化参数。元学习模型经过不断训练以后,可以凭借已经学习到的经验快速生成适应新环境的模型参数,这使得元学习模型在面对各种基于梯度下降的具体问题时,比如回归问题、分类问题和强化学习等,都能够展现出快速适应的能力,而不会过拟合,

同时,元学习模型的适应能力并不受限于具体实例模型,这一特性也体现出 MAML 算法与具体模型无关的特点。

#### 3.2 模型架构设计

在本文中,基于 MAML 算法的研究考虑两种神经网络,即 FNN 和 CNN,如图 3 所示。

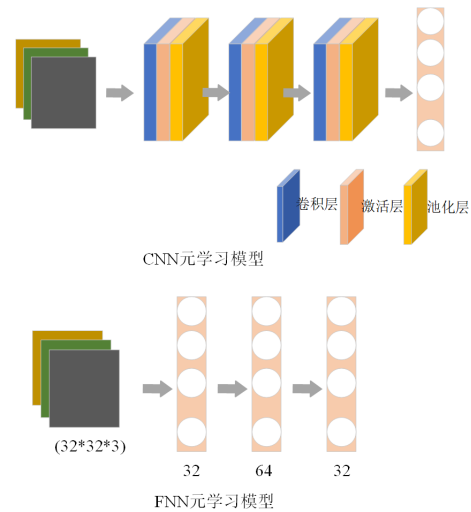


图3 元学习模型结构图

FNN 采用三个全连接层,前两个全连接层分别包含 32 和 64 个神经元,最后一层是输出层。其中信道复数矩阵的实部和虚部与信噪比值共同构成一个三维信道特征图像,三维的信道特征图像会先经过扁平化处理为一维向量作为输入。并在前两个全连接层采用 ReLU 作为激活函数,表示如下:

$$ReLU(x) = \max(x, 0) \quad (2)$$

在输出层采用 softmax 作为激活函数进行分类:

$$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^N e^{x_j}} \quad (i = 1, 2, \dots, N) \quad (3)$$

其中,  $x_i$  代表 softmax 函数的输入信号,也即输出层输出向量的第  $i$  个分量,  $x_j$  代表输出层输出向量的第  $j$  个分量,输出值  $f(x_i)$  代表样本属于第  $i$  类别的概率。

CNN 包含 3 个卷积模块和一个用来分类的全连接层,每个卷积模块由卷积层、ReLU 函数和池化层组成。首个卷积模块的卷积层有 64 个卷积核,卷积核大小为  $3 \times 3$ ,执行卷积操作过程的步长为 1。之后两个卷积模块都包含 32 个卷积核,每个卷积核的尺寸为  $3 \times 3$ ,步长为 1。前 2 个卷积模块中的池化层大小为  $2 \times 2$ ,最后一个卷积模块中池化层大小为  $2 \times 1$ 。在每个卷积模块的卷积层后都添加一个 ReLU 激活函数。信道参

数被卷积模块提取特征后,得到的特征图被展平为一维向量。将所得到的一维向量输入至输出层,并将结果输入至 softmax 激活函数,最终得到信道状态信息对应的 MCS。

### 3.3 算法流程设计

元学习和传统机器学习的不同点在于,元学习是以任务为单位进行训练,而在传统的机器学习中,模型优化的基本单位是单个数据样本。在元学习中,训练的过程则分为内外两层循环。首先,在元学习的内层训练,每个任务对应的数据由训练数据和测试数据组成,分别用支撑集和查询集来表示,一个训练批次由多个任务组成。其次,每个任务的支撑集用来作为内层循环的训练单位,查询集作为外层循环的训练单位。尽管传统机器学习和元学习的目标都是找到一个优选的函数,但对应的两个函数的作用有很大不同。在传统机器学习中,这个函数用于刻画输入数据(即特征)和输出数据之间的直接关联。但是,元学习是基于找到的这个函数,来产生新的函数,后者会被应用到新的任务中。具体而言,假设任务集合为  $T = \{T_1, \dots, T_k, \dots, T_U\}$ , 每个任务  $T_k$  的数据进一步被划分为支撑集  $\mathbb{R}_k^{su} = \{(y_k^{su}, H_k^{su})\}$  和查询集  $\mathbb{Q}_k^{qu} = \{(y_k^{qu}, H_k^{qu})\}$ , 其中  $H_k^{su}$  表示任务  $T_k$  中支撑集的信道特征数据,  $y_k^{su}$  表示离线采集的信道数据集里  $H_k^{su}$  对应的 MCS 取值。在训练过程中,使用支撑集  $\mathbb{R}_k^{su}$  来更新内循环中基学习器的参数,使用查询集  $\mathbb{Q}_k^{qu}$  来更新外循环中元学习器的参数。将所有任务对应的输入数据分为元训练数据集和元测试数据集,将其分别表示为  $D_{train}$  和  $D_{test}$ 。

图4描绘了基于元学习的自适应调制编码算法流程。

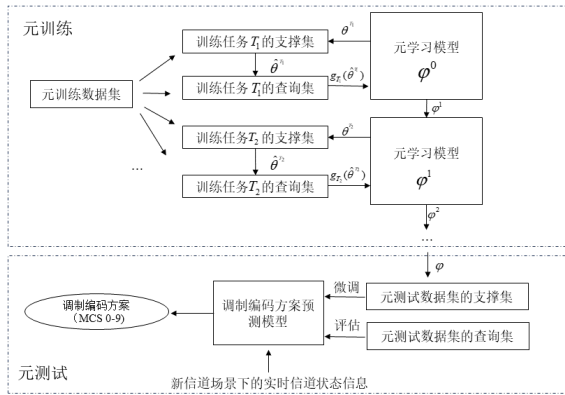


图4 基于元学习的自适应调制编码算法流程图

整个算法分为两个阶段:元训练阶段和元测试阶段。在元训练阶段,主要是为了训练获取神经网络的

初始化参数。通过从元训练数据集  $D_{train}$  中采样一些任务进行训练,学习到更具有迁移性的内部特征,并获得一个好的神经网络的初始化参数。在元测试阶段,在该神经网络初始化参数的基础上,使用少量来自元测试数据集  $D_{test}$  的样本对神经网络模型进行微调,并在新任务的未知样本中进行验证。以上两阶段的算法详细流程描述如下:

#### (1) 元训练阶段

元训练阶段的目标是确定神经网络初始化参数。在该阶段中,为了避免采用单层循环造成模型参数在训练任务上无法收敛或者过拟合,采用内外两层循环的训练方式,在不同任务上不断递归地优化模型参数。该阶段中,神经网络的初始化参数由两个学习器获得,分别是元学习器和基学习器。假设元学习器上的神经网络初始化参数为  $\varphi$ , 基学习器上的神经网络初始参数为  $\theta$ , 对第  $k$  个任务  $T_k$ , 将最新的元学习器模型参数  $\varphi$  赋值给基学习器模型参数  $\theta$ , 用于内层循环的神经网络参数更新,首先从任务  $T_k$  的支撑集  $\mathbb{R}_k^{su}$  中采样  $N$  条样本数据,并使用梯度下降算法更新内循环参数,则有:

$$\theta_k^{su} = \theta_{k-1}^{su} - \alpha \nabla_{\theta_{k-1}^{su}} L_{T_k}(f_{\theta_{k-1}^{su}}, \mathbb{R}_k^{su}) \quad (4)$$

其中  $\alpha$  为内循环学习率,  $\theta_{k-1}^{su}$  是基学习器中上一个任务训练完成的神经网络参数,  $f_{\theta_{k-1}^{su}}$  是神经网络参数  $\theta_{k-1}^{su}$  的对应基学习器,  $L_{T_k}$  代表任务  $T_k$  的损失函数,  $\nabla_{\theta_{k-1}^{su}}$  代表损失函数对  $\theta_{k-1}^{su}$  的梯度, 然后从查询集  $\mathbb{Q}_k^{qu}$  中采样  $N$  条样本数据来验证内层神经网络的性能,基于当前任务  $T_k$  在查询集上的损失函数,并使用 Adam 优化器来更新元学习模型参数。元目标如下:

$$\min_{\theta} \sum_{T_k} L_{T_k}(f_{\theta_k^{su}}) = \sum_{T_k} L_{T_k}(f_{\theta_{k-1}^{su}} - \alpha \nabla_{\theta_{k-1}^{su}} L_{T_k}(f_{\theta_{k-1}^{su}}, \mathbb{R}_k^{su})) \quad (5)$$

元学习模型参数按照下式进行更新:

$$\varphi_k^{qu} = \varphi_{k-1}^{qu} - \beta \sum_{T_k} \nabla_{\varphi_{k-1}^{qu}} L_{T_k}(f_{\varphi_{k-1}^{qu}}, \mathbb{Q}_k^{qu}) \quad (6)$$

其中  $\beta$  为外循环学习率,  $\varphi_{k-1}^{qu}$  是元学习器中上一个任务训练完成的神经网络的参数。然后,元学习器将  $\varphi_k^{qu}$  作为下一个训练任务的神经网络的初始值,并利用下一个任务的支撑集和查询集重复执行上述流程,不断循环这些步骤,直到元训练集中的所有训练任务均参与训练。其中损失函数定义为交叉熵损失函数,其表达式为:

$$L_{T_k}(f_{\phi}) = \sum_{x^{(j)} \in \mathbb{Q}_k^{qu}} y^{(j)} \log f_{\phi}(x^{(j)}) + (1 - y^{(j)}) \log (1 - f_{\phi}(x^{(j)})) \quad (7)$$

详细算法步骤见表1。

表1 基于Meta-FNN和Meta-CNN的网络初始化算法

算法 基于Meta-FNN和Meta-CNN的网络初始化算法

参数:任务的分布  $T \sim \rho(T)$

参数: $\alpha$ 和 $\beta$ ,它们分别为内层梯度更新学习率和外层梯度更新学习率。

1. 初始化元学习模型  $f_\theta$  的权重参数  $\theta$
2. for 迭代轮次  $t$  do
3. 采样 batch 个任务  $T_k$ , 其中  $T_k \sim \rho(T)$
4. for each  $T_k$  do
5. 从  $T_k$  的支撑集  $\mathbb{R}_k^m$  中随机采样  $N$  个样本
6. 根据式(4)梯度下降更新内层网络参数  $\theta_k^m$
7. end for
9. 根据式(6)梯度下降更新外层网络参数  $\varphi_k^m$
10. end for

## (2)元测试阶段

在元测试阶段,针对新的信道场景建立数据集并对元训练阶段得到的神经网络参数进行微调。首先,将元学习模型的权重参数加载到新场景下的神经网络中。针对新场景下的模型微调过程不再需要随机初始化参数,而是直接使用在元训练阶段获得的模型参数。采用元测试数据集的支撑集对神经网络参数微调,并用元测试数据集的查询集对模型进行评估,得到最终的调制编码方案预测模型。对令  $\varepsilon$  为微调过程的学习率,更新过程可以表示为:

$$\phi_T = \varphi - \varepsilon \nabla_{\theta} L_T(f_{\varphi}) \quad (8)$$

具体算法微调流程见表2。

表2 Meta-CNN和Meta-FNN的微调算法

算法 微调算法

1. 输入:离线训练完成的元学习神经网络,其参数为  $\varphi$
2. 输出:训练完成的元学习神经网络参数  $\phi$
3. for 微调次数  $t$  do
4. 根据式(8)更新在新场景下的神经网络参数  $\phi_T$
5. end for

## 4 实验和讨论

### 4.1 实验参数设置

实验采用的MCS取值见表3,共有10种MCS取值,实验数据集方面,使用在TGah信道模型B采集的19600个数据样本用来训练,在信道模型C和信道模型D下采集的5000个数据样本用于微调神经网络,8400个数据样本用于测试<sup>[8]</sup>。三个信道模型的对比参数见表4所示。

通信仿真实验在16GB内存、8核CPU的电脑上

运行,处理器频率为3.1GHz。同时在Google Goloab搭建了基于python3.8环境和Tensorflow2.6.2环境的实验平台,并采用Keras库进行模型训练。仿真实验及模型训练所采用的参数见表5所示。

表3 802.11ah调制编码方案

MCS	调制方案	编码码率	空间数据流	速率(kb/s)
0	BPSK	1/2	1	333.3
1	QPSK	1/2	1	666.7
2	QPSK	3/4	1	1000.0
3	16-QAM	1/2	1	1333.3
4	16-QAM	3/4	1	2000.0
5	64-QAM	2/3	1	2666.7
6	64-QAM	3/4	1	3000.0
7	64-QAM	5/6	1	3333.3
8	256-QAM	3/4	1	4000.0
9	256-QAM	5/6	1	4444.4

表4 不同信道场景的参数对比

参数	信道B	信道C	信道D
断点距离(m)	5	5	10
RMS时延扩展(ns)	15	30	50
最大时延(ns)	80	200	390
Rician K因子(dB)	0	0	3
抽头数	9	14	18
集群(clusters)数	2	2	3

表5 仿真实验及模型训练参数

参数	取值
数据子载波数	24
循环前缀长度( $\mu$ s)	4
导频子载波数	2
载波频率	2.4 GHz
载波间隔	31.25 KHz
信道带宽(MHz)	1
$N_s \times N_c$	$32 \times 32$
内循环学习率	0.4
外循环学习率	0.001
迭代次数	100
批次(batch)大小	128

### 4.2 实验结果及讨论

为了验证本文所提出的算法性能,本节给出仿真实验结果及讨论。我们将采用基于FNN元学习模型的自适应调制编码算法称为Meta-FNN算法,将基于CNN元学习模型的自适应调制编码算法称为Meta-CNN算法,并对所提出的算法与传统CNN和FNN算法进行性能比较。为防止无关变量的干扰,传统CNN选择与CNN元学习模型相同的模型架构。

图5仿真了在信道C上,使用基于MAML和基于原始神经网络的自适应调制编码算法的系统吞吐量性能。由图5中可以看出,基于元学习参数初始化的Meta-CNN和Meta-FNN算法,相较于传统的CNN和FNN算法,展现出显著的性能优势。这是因为传统的CNN和FNN算法没有进行元学习的参数初始化。对于未在训练集里出现的信道C,传统的CNN和FNN模型的泛化性能较差,这体现了元学习算法的优越性。

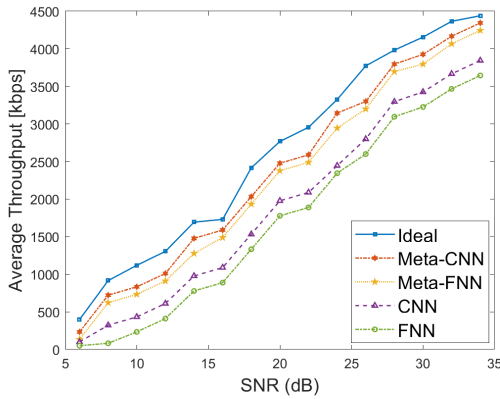


图5 信道C上Meta-CNN、Meta-FNN及基线算法的吞吐量

图6仿真了基于MAML和基于传统神经网络的自适应调制编码算法在信道D上的系统吞吐量性能。

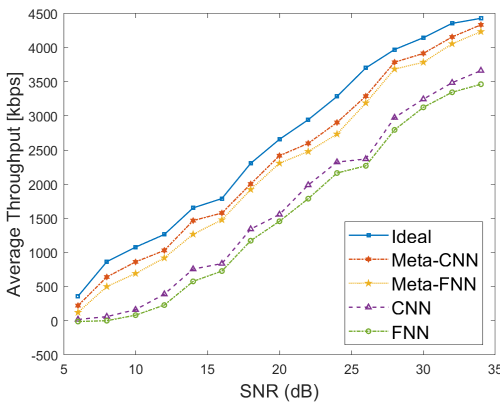


图6 信道D上Meta-CNN、Meta-FNN及基线算法的吞吐量

由图6中可以看出,采用了元学习算法的Meta-CNN和Meta-FNN明显好于采用传统CNN和FNN的吞吐量性能,同时,值得注意的是,由于信道D相对于信道C和信道B的差异性更大,采用原始神经网络与采用Meta-CNN和Meta-FNN的吞吐量性能差距更明显,这进一步验证了元学习算法的优势。同时,在信道C和信道D场景下,我们发现基于CNN的MAML算法

的自适应性能比基于FNN的MAML算法效果更好,说明相对于所采用的FNN而言,所采用的CNN的特征提取能力更强。以上结果表明基于元学习训练而来的初始化参数在不同信道场景下具有较好的泛化能力。

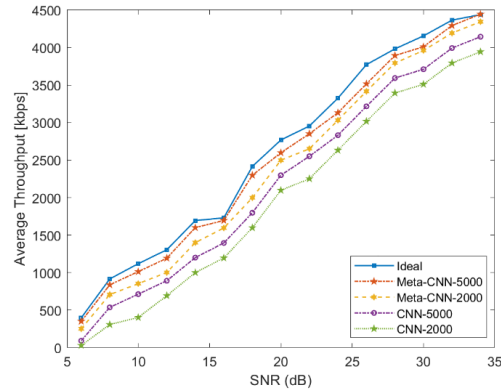


图7 信道C上Meta-CNN与CNN在微调时的吞吐量

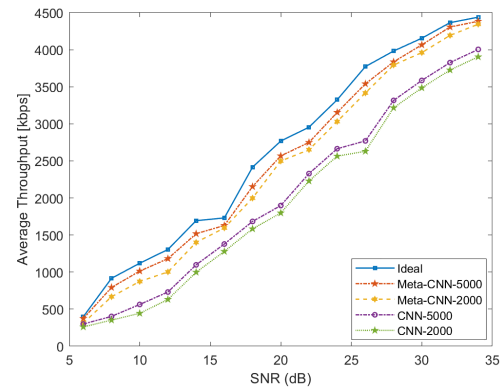


图8 信道D上Meta-CNN与CNN在微调时的吞吐量

为了更深入地探究所提方法相比传统机器学习方法在迁移至新信道环境时的优势,我们设计了进一步的实验。图7仿真了Meta-CNN和CNN均采用5000和2000样本数时在信道C上微调后的吞吐量性能。由图7中可以发现,当训练所使用样本数量达到5000时,Meta-CNN和传统CNN的吞吐量曲线较为接近,然而,当训练样本数减少为2000时,传统CNN在少量标签样本的情况下,难以有效学习并适应新信道场景下的数据分布,导致其性能较差。相比之下,Meta-CNN在标签样本较少时,仍能保持不错的自适应性能。与CNN算法相比,Meta-CNN针对差异性信道环境的微调过程需要的标签样本更少,传统的CNN不能在少量样本下适用于新环境。这体现出了所提的基于元学习的自适应调制编码方案在小样本学习上的优势。

图8仿真了Meta-CNN和CNN均采用5000和

2000样本时在信道D上微调后的吞吐量性能。由图8中可以看出,采用了MAML算法以后的Meta-CNN模型的吞吐量性能好于传统CNN模型的吞吐量性能。同时,我们发现,用基于元学习算法的Meta-CNN时,尽管信道D与信道B差异性更大,仍然取得了不错的性能,同时,CNN模型尽管经过了微调,但是当信道场景差异更大时,在少量样本上无法达到较好的吞吐量性能,以上结果均证明了采用所提算法比采用传统神经网络在场景迁移时能获得更好的吞吐量性能。

## 5 结论

本文针对基于传统深度学习的自适应调制编码方法在信道场景迁移时模型泛化能力下降的问题,提出了一种基于元学习的自适应调制编码算法。该算法由神经网络初始化参数的训练和在新的信道场景下的微调两部分组成,通过在元训练阶段充分学习可迁移知识,神经网络模型在新的信道场景只需要少量样本进行微调即可获得较高的预测性能。仿真结果和讨论表明,本文所提算法在信道场景迁移时仍然能够保持较好的吞吐量性能,并且相比现有深度学习方法,所提算法所需要的微调样本数更少,模型泛化能力更好。

## 参考文献(References):

- [1] Zhang C, Lei X, Yuan Y, et al. A learning approach to link adaptation based on multi-entities Bayesian network [J]. *Cluster Computing*, 2019, 22(Suppl 4): 8463-8473.
- [2] Daniels R C, Caramanis C M, Heath R W. A supervised

- learning approach to adaptation in practical MIMO-OFDM wireless systems [C]//*IEEE Global Telecommunications Conference*, 2008.
- [3] Daniels R C, Caramanis C M, Heath R W. Adaptation in convolutionally coded MIMO-OFDM wireless systems through supervised learning and SNR ordering [J]. *IEEE Transactions on Vehicular Technology*, 2010, 59(1): 114-126.
- [4] Yigit H, Kavak A. Adaptation using neural network in frequency selective MIMO-OFDM systems [C]// *5th International Symposium on Wireless Pervasive Computing*, 2010: 390-394.
- [5] Dong Z, Shi J, Wang W, et al. Machine learning based link adaptation method for MIMO system [C]//*29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2018: 1226-1231.
- [6] Tato A, Mosquera C, Henarejos P, et al. Neural network aided computation of mutual information for adaptation of spatial modulation [J]. *IEEE Transactions on Communications*, 2020, 68(5): 2809-2822.
- [7] Hussien M, Ahmed M F A, Dahman G, et al. Towards more reliable deep learning-based link adaptation for WiFi6 [C]// *International Conference on Communications*, 2021: 1-6.
- [8] Elwekeil M, Jiang S, Wang T, et al. Deep convolutional neural networks for link adaptations in MIMO-OFDM wireless systems [J]. *IEEE Wireless Communications Letters*, 2018, 8(3): 665-668.
- [9] Finn C, Abbeel P, Levine S. Model-agnostic meta-learning for fast adaptation of deep networks [C]//*34th International Conference on Machine Learning*, 2017, 70: 1126-1135.

编辑:王谦