

空中传输推理在多跳 MIMO 网络中的应用

Chenghong Bian, *Student Member, IEEE*, Meng Hua, *Member, IEEE*, and Deniz Gündüz, *Fellow, IEEE*

Abstract——一种新的空中传输机器学习框架在多跳多输入多输出 (MIMO) 网络上被提出。核心思想是通过精心设计发射节点的预编码矩阵, 利用多个 MIMO 信道模仿全连接 (FC) 神经网络层。采用一个名为原型网络的神经网络, 该网络包含多个 FC 层, 并且每一层的神经元数量等于相应终端的天线数量。为了达到令人满意的性能, 我们基于自定义损失函数对 PrototypeNet 进行训练, 该损失函数由分类误差和潜在向量功率组成以满足发射功率约束条件, 在训练过程中加入了噪声注入。然后通过求解一个优化问题来获得每个跳转段的预编码矩阵。当天线数量有限时, 我们也提出了多块区的扩展方案。数值结果验证了所提出的空中传输方案在功率约束下可以实现令人满意的分类准确性。结果显示, 在适度的信噪比 (SNR) 条件下, 随着跳跃次数的增加, 可以获得更高的分类准确性。

Index Terms——空中传输、多跳、MIMO、图像分类。

I. 介绍

存在一种日益增长的趋势, 即将计算与无线通信集成起来以提升未来物联网 (IoT) 网络的性能 [1], [2]。与传统数字方案中计算和通信分离不同, 空中计算 (OAC) 通过使用模拟传输实现了集成通信和计算 [3]–[6]。OAC 方案的一个关键优势是利用无线信号的传播原理将计算负担转移到无线领域。例如, 可以通过探索无线介质的信号叠加特性来执行求和和多项式等线性操作, 这些已被广泛应用于联邦学习中 [3], [4]。

最近, 边缘学习 [1] 已经得到了广泛研究, 在这种情况下, 节点参与机器学习任务并通过无线链路相互通信。为了促进这一点, 提出了空中神经网络 (NN) 的概念, 其中将无线信道的输出视为 NN 的输出 [7]–[9]。例如, [7] 的作者通过利用可重构智能表面周围的无线传播环境构建了一个用于调制分类问题的卷积神经网络。受到多输入和多输出 (MIMO) 信道与预编码矩阵之间以及全连接层之间输入-输出关系相似性的启发, [8] 和 [9] 的作者开发了无线网络上的高效深度学习算法的 OAC

方案。具体来说, 在 [8] 中, 构建了一个 OAC 全连接层, 其中多个同步用户对其消息进行预编码并将其传递给目标节点, 形成了多输入单输出 (MISO) 系统。进行了系统级验证, 并且在 MNIST 数据集上评估时实现了 92% 的分类准确率。然而, 其性能受限于接收端的单一天线。一个更复杂的 OAC FC 层在 [9] 中基于 MIMO 信道考虑了分割学习, 其中 FC 层是通过收发端的预编码和合并矩阵构建的。梯度通过 MIMO 信道反向传播以更新预编码矩阵, 导致了大量的计算开销, 这不适合低复杂度的边缘设备。此外, 这些工作集中在单跳场景。

为了填补这一空白, 本文研究了多跳 MIMO 网络上的 OAC 机器学习。一个多跳 MIMO 网络类似于具有多个全连接层的复数神经网络。我们首先设计了一个新颖的 PrototypeNet, 它由多个全连接层、功率归一化和非线性激活函数组成, 并使用定制的损失函数 (分类误差与潜在向量功率的组合) 以端到端的方式进行训练。获得 PrototypeNet 权重以及信道状态信息 (CSI) 后, 通过求解优化问题来获取每跳的预编码矩阵, 目标是尽可能准确地空中近似 PrototypeNet 的操作。这与 [9] 不同, 在那里预编码和合并矩阵会随着神经网络权重一起训练。数值实验表明, 所提出的 OAC 方案可以在功率约束下实现令人满意的分类精度。结果还显示, 即使是在适度的信噪比值下, 更多的跳数也能带来更高的分类精度。虽然这种方法在空中部署 NNs 时取得了成功, 但 NN 的宽度受限于发射机上的天线数量。为了克服这一限制, 我们提出通过时间传输每个输入样本的多个信道符号。在所提出的方案中, 高维输入被划分为几个低维组件, 每个组件由不同的预编码矩阵进行预编码。本文表明, 采用多符号传输可以显著提高分类性能, 允许使用有限数量的天线部署更宽的 NN。

II. 系统模型

我们考虑一个多跳 MIMO 网络上的图像分类问题, 但结果可以应用于 NN 执行的其他任务。一个由 $\mathbf{S} \in \mathbb{R}^{C \times H \times W}$ 表示的图像, 其中 C 、 H 和 W 分别代表图像的通道数、高度和宽度, 在源节点 (S) 可用。每

The authors are with the Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, U.K. (e-mail: {c.bian22,m.hua,d.gunduz}@imperial.ac.uk).

This work received funding from the UKRI for the projects AI-R (ERC Consolidator Grant, EP/X030806/1) and the SNS JU project 6G-GOALS under the EU' s Horizon program Grant Agreement No. 101139232.

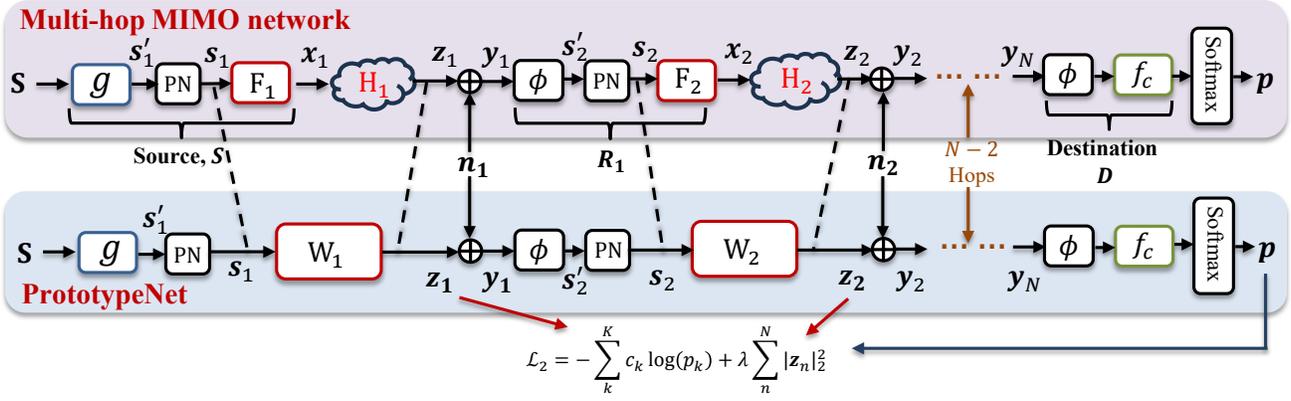


Figure 1: 多跳 MIMO 网络的流程图和 PrototypeNet。我们考虑一个 N -跳 MIMO 网络，这相当于具有 $(N + 1)$ 个全连接层（包括终点节点处的一个）的 PrototypeNet。在 (9) 中定义的第二个损失函数 \mathcal{L}_2 用于训练 PrototypeNet。

个图像 S 属于某个类别，可以通过一个 one-hot 向量 $c \in \mathbb{R}^K$ 来表示，其中 K 是类别的总数。目标节点 (D) 希望检测图像类别。然而，D 无法访问该图像，并且仅通过由 $\{R_1, \dots, R_{N-1}\}$ 表示的 $(N - 1)$ 个中继节点与 S 相连。源节点、目的节点和中继节点都配备了多根天线。我们进一步假设每跳的信道状态信息在发射机处均可获得。OAC 多跳 MIMO 网络的流程图如图 1 所示，具体细节如下。

源节点通过函数 $g(\cdot)$ 将输入图像 S 转换为一个复值向量，记作 $s'_1 \in \mathbb{C}^{M_1}$ ，此操作可通过二维平均池化实现，其中 M_1 表示源节点的天线数量。值得一提的是，我们分别用 $M_{i+1}, i \in [1, N - 1]$ 和 M_{N+1} 表示第 i 个中继节点和目的地节点的天线数量。然后，我们将幂归一化 (PN) 应用于复向量 s'_1 以获得 s_1 ，表示为：

$$s_1 = (s'_1 - \mu_1)/v_1, \quad (1)$$

其中 μ_1 和 v_1 分别表示 s'_1 的均值和方差，在训练过程中记录下来。PN 模块确保了 $\mathbb{E}[\|s_1\|_2^2] \leq M_1$ 。源节点 S 采用预编码矩阵 $F_1 \in \mathbb{C}^{M_1 \times M_1}$ ，传输信号记为 x_1 ，可以表示为：

$$x_1 = F_1 s_1. \quad (2)$$

通过信道传输后，第一个中继节点 R_1 接收到的信号 $y_1 \in \mathbb{C}^{M_2}$ 可以表示为：

$$y_1 = H_1 x_1 + n_1, \quad (3)$$

其中 H_1 表示源与中继 1 之间的信道，其每个元素是独立同分布 (i.i.d.) 且遵循复高斯分布，即 $\mathcal{CN}(0, 1)$ ，而 n_1 表示在中继 1 接收到的噪声并满足 $n_1 \sim \mathcal{CN}(0, \sigma_1^2 I_{M_2})$ 。

在 R_1 处，接收到的信号 y_1 首先被馈送到非线性激活函数 $\phi(\cdot)$ 中，该函数实现为一个复数 ReLU 函数。具体

来说， y_1 的实部和虚部都通过了标准的 ReLU 函数。对非线性激活输出应用功率归一化以生成 s_2 。我们强调，PN 层对于满足中继节点 R_1 的功率约束是必不可少的。然后，第一个中继节点的传输信号可以表示为：

$$x_2 = F_2 \text{PN}(\phi(y_1)), \quad (4)$$

其中， F_2 是在 R_1 处的预编码矩阵，而 PN 表示功率归一化函数。我们注意到剩余的中继节点遵循相同的程序，且在第 i 跳中的传输信号、信道状态信息和信道噪声分别满足 $\mathbb{E}[\|s_i\|_2^2] \leq M_i$ 、 $H_i \in \mathbb{C}^{M_{i+1} \times M_i}$ 和 $n_i \sim \mathcal{CN}(0, \sigma_i^2 I_{M_{i+1}})$ 。

接收的信号 y_N 在目的地首先被输入到复数 ReLU 函数 $\phi(\cdot)$ 中，并通过连接复向量的实部和虚部，将其转换为一个 $2M_{N+1}$ 维的实值向量。经过 FC 层 $f_c(\cdot)$ 并跟随 softmax 激活后，获得一个概率向量 $p \in \mathbb{R}^K$ ，其中 p 中最大元素的索引 k^* 是最终分类输出。系统的分类准确率定义为：

$$\text{Acc}(\%) = \mathbb{E}(\mathbf{1}(k^* = c^*)), \quad (5)$$

其中， $\mathbf{1}(\cdot)$ 是指示函数，而 c^* 表示正确的类别索引。在下面的内容中，我们假设经过各个跳点的天线数量和噪声功率是相同的，即 $M_1 = \dots = M_{N+1} \triangleq M$ 和 $\sigma_1^2 = \dots = \sigma_{N+1}^2 \triangleq \sigma^2$ ，为了简化起见。

III. 方法论

在本节中，我们首先介绍所提方案的动机和训练方法。然后，将展示获得预编码矩阵 $\{F_1, \dots, F_N\}$ 的过程。

A. 原型网络

在提出的解决方案中，我们希望中继网络的第 i 跳作为目标神经网络的第 i 层。如图 1 所示，对于第 i 跳的

MIMO 信道实现, \mathbf{H}_i , 我们的目标是确定一个预编码矩阵 \mathbf{F}_i , 使得在第 i 个中继节点接收到的信号可以被视为一个 FC 层的输出 (不包括偏置项)。令 $\mathbf{W}_i \in \mathbb{C}^{M_{i+1} \times M_i}$ 表示称为 PrototypeNet 的目标神经网络的第 i 层的权重矩阵。第 i 跳接收到的信号可以表示为:

$$\mathbf{y}_i = \mathbf{H}_i \mathbf{F}_i \mathbf{s}_i + \mathbf{n}_i. \quad (6)$$

由于我们希望中继网络的行为类似于 PrototypeNet, 因此问题可以被公式化为寻找 \mathbf{F}_i 使得 $\mathbf{H}_i \mathbf{F}_i$ 等同于 \mathbf{W}_i 。首先, 我们将引入一种新的端到端优化算法来确定 PrototypeNet 的权重 \mathbf{W}_i , 具体细节如下。

首先, 我们构建了 PrototypeNet, 如图 1 所示, 它类似于多跳 MIMO 网络。特别地, PrototypeNet 和多跳 MIMO 网络共享相同的变换函数 $g(\cdot)$ 、PN 模块以及最终的 FC 层 $f_c(\cdot)$ 。乍一看, PrototypeNet 与标准图像分类网络相同。然而, 通过训练标准图像分类网络获得的权重在部署到多跳 MIMO 网络时无法达到令人满意的性能。这是因为标准网络没有考虑 $\mathbf{W}_i, i \in [1, N]$ 的 Frobenius 范数。此外, 在训练标准网络时不考虑信道噪声, 这会降低其在噪声信道中实现时的性能。

提出的 PrototypeNet 考虑了功率约束以及信道噪声。对于第 i 个节点, 在生成复向量 \mathbf{s}_i 如同 (1) 中所示, 下一个节点的输出 \mathbf{y}_i 得到如下:

$$\mathbf{y}_i = \mathbf{z}_i + \mathbf{n}_i, \quad (7)$$

其中我们定义了 $\mathbf{z}_i \triangleq \mathbf{W}_i \mathbf{s}_i$ 。直观上, 存在一个功率消耗 \mathbf{z}_i 与分类准确性之间的权衡: 当 $\|\mathbf{z}_i\|_2^2$ 较大时, 噪声 \mathbf{n}_i 可以忽略不计, 并且对准确性的影响很小。另一方面, 当 $\|\mathbf{z}_i\|_2^2$ 较小时, 噪声占主导地位并且性能显著下降。

为了实现功率和分类准确率之间的权衡, 我们考虑使用两种损失函数来训练 PrototypeNet:

$$\mathcal{L}_1 = -\sum_{k=1}^K c_k \log(p_k) + \frac{\lambda}{N} \sum_{j=1}^N \|\mathbf{W}_j\|_F^2, \quad (8)$$

$$\text{and } \mathcal{L}_2 = -\sum_{k=1}^K c_k \log(p_k) + \frac{\lambda}{N} \sum_{j=1}^N \|\mathbf{z}_j\|_2^2, \quad (9)$$

其中 λ 平衡了两个损失函数的功率和分类准确率。请注意, \mathcal{L}_1 的第二项, 即 $\frac{\lambda}{N} \sum_{j=1}^N \|\mathbf{W}_j\|_F^2$, 最小化权重的 Frobenius 范数, 这类似于广泛使用的权重正则化项以避免过拟合。在这种情况下, 该项隐式地最小化了 \mathbf{z}_i 的功率。另一方面, 第二种损失函数 \mathcal{L}_2 直接最小化 $\|\mathbf{z}_i\|_2^2$ 。尽管通过模拟显示, 使用 \mathcal{L}_2 训练 PrototypeNet 会导致更满意的结果。

Algorithm 1 优化第 i 个预编码矩阵 \mathbf{F}_i 。

```

1: Input  $\{\mathbf{W}_i, \mathbf{H}_i, \tilde{\mathbf{S}}_i, \lambda_1, \eta, \epsilon\}$ ; Output  $\mathbf{F}_i$ 
2: Initialize  $\mathbf{F}_i = \mathbf{H}_i^{-1} \mathbf{W}_i$ ,
3: Objective  $G_p = \|\mathbf{W}_i - \mathbf{H}_i \mathbf{F}_i\|_F^2 + \lambda_1 \text{tr}(\tilde{\mathbf{S}}_i \mathbf{F}_i^\dagger \mathbf{F}_i)$ 
4: while True do:
5:   Calculate the gradient:  $\nabla \mathbf{F}_i = -2\mathbf{H}_i^\dagger \mathbf{W}_i + 2\mathbf{H}_i^\dagger \mathbf{H}_i \mathbf{F}_i + 2\lambda_1 \mathbf{F}_i \tilde{\mathbf{S}}_i$ ,
6:   Update:  $\mathbf{F}_i = \mathbf{F}_i - \eta \nabla \mathbf{F}_i$ ,
7:   New objective:  $G_c = \|\mathbf{W}_i - \mathbf{H}_i \mathbf{F}_i\|_F^2 + \lambda_1 \text{tr}(\tilde{\mathbf{S}}_i \mathbf{F}_i^\dagger \mathbf{F}_i)$ ,
8:   if  $|G_c - G_p| < \epsilon$  then
9:     Return  $\mathbf{F}_i$ .
10:  else
11:     $G_p = G_c$ .

```

最后, 我们定义在 N 跳上的平均接收功率 P_r 为:

$$P_r = \frac{1}{N} \sum_{j=1}^N \|\mathbf{z}_j\|_2^2 = \frac{1}{N} \sum_{j=1}^N \|\mathbf{W}_j \mathbf{s}_j\|_2^2. \quad (10)$$

请注意, 我们也对整个网络的平均发射功率感兴趣, 定义如下:

$$P_t = \frac{1}{N} \sum_{i=1}^N \|\mathbf{F}_i \mathbf{s}_i\|_2^2. \quad (11)$$

然而, 由于我们在训练 PrototypeNet 时无法访问 $\mathbf{F}_i \mathbf{s}_i$, 我们使用 P_r 来训练它, 如 (9) 中所示。

B. 计算 \mathbf{F}_i

一旦获得 PrototypeNet 的权重, 在每次网络状态实现时, 我们的目标将是确定预编码矩阵以近似 PrototypeNet 的空中介入行为。以第一跳为例: 由于 \mathbf{H}_1 在源节点可用, 将使用 \mathbf{W}_1 和 \mathbf{H}_1 来确定预编码矩阵 \mathbf{F}_1 。

请注意, \mathbf{H}_1 的每个元素都服从独立同分布的复高斯分布。可以证明, \mathbf{H}_1 概率为一的情况下是满秩的。因此, \mathbf{H}_1 的逆矩阵总是存在的。可以通过简单最小二乘准则来获得

$$\mathbf{F}_1 = \mathbf{H}_1^{-1} \mathbf{W}_1. \quad (12)$$

在这种情况下, 由于我们有 $\mathbf{H}_1 \mathbf{F}_1 = \mathbf{W}_1$, 相应的分类准确率与 PrototypeNet 相同。然而, 这种简单的做法未能考虑到定义在 (11) 中的发射功率消耗 P_t 。特别地, \mathbf{F}_1 在 (12) 中的轻微偏差可以在不牺牲分类准确性的情况下显著降低功耗。为此, 我们提出了一种替代方法来确定 \mathbf{F}_1 , 具体如下。

我们制定了一个优化问题，目标是在消耗低传输功率的同时实现高分类准确率。对于前者目标，由于 \mathbf{W}_1 已经被优化，我们希望项 $\|\mathbf{H}_1\mathbf{F}_1 - \mathbf{W}_1\|_F^2$ 尽可能小。对于第二个目标，我们感兴趣的是最小化所有 \mathbf{s}_1 的传输功率的期望值，即 $\mathbb{E}(\|\mathbf{F}_1\mathbf{s}_1\|_2^2)$ 。请注意， $\mathbb{E}(\|\mathbf{F}_1\mathbf{s}_1\|_2^2)$ 可以表示为：

$$\mathbb{E}(\|\mathbf{F}_1\mathbf{s}_1\|_2^2) = \mathbb{E}(\text{tr}(\mathbf{s}_1\mathbf{s}_1^\dagger\mathbf{F}_1^\dagger\mathbf{F}_1)) = \text{tr}(\mathbf{S}_1\mathbf{F}_1^\dagger\mathbf{F}_1), \quad (13)$$

其中 $\mathbf{S}_1 \triangleq \mathbb{E}(\mathbf{s}_1\mathbf{s}_1^\dagger)$ 和 $\text{tr}(\cdot)$ 计算矩阵的迹。鉴于底层图像分布 P_S 未知，无法计算确切的 \mathbf{S}_1 项。因此，我们通过训练集的平均值来估计 \mathbf{S}_1 ：

$$\tilde{\mathbf{S}}_1 = \frac{1}{N_{tr}} \sum_{i=1}^{N_{tr}} \mathbf{s}_{1,i}\mathbf{s}_{1,i}^\dagger, \quad (14)$$

其中 N_{tr} 表示训练集中图像的数量，而 $\mathbf{s}_{1,i}$ 是与第 i 个图像对应的在 (1) 中的变换输入。

最后，优化问题可以表示为：

$$\min_{\mathbf{F}_1} \underbrace{\|\mathbf{W}_1 - \mathbf{H}_1\mathbf{F}_1\|_F^2 + \lambda_1 \text{tr}(\tilde{\mathbf{S}}_1\mathbf{F}_1^\dagger\mathbf{F}_1)}_G, \quad (15)$$

其中 λ_1 平衡了分类准确率和传输功率消耗。我们注意到简单的解决方案 $\mathbf{F}_1 = \mathbf{H}_1^{-1}\mathbf{W}_1$ 可以被视为 (15) 在 $\lambda_1 = 0$ 情况下的一个特例。容易证明优化问题是凸的，可以通过梯度下降求解。具体来说，对于给定的 \mathbf{H}_1 和 \mathbf{W}_1 ， \mathbf{F}_1 按如下方式更新：

$$\begin{aligned} \nabla\mathbf{F}_1 &= -2\mathbf{H}_1^\dagger\mathbf{W}_1 + 2\mathbf{H}_1^\dagger\mathbf{H}_1\mathbf{F}_1 + 2\lambda_1\mathbf{F}_1\tilde{\mathbf{S}}_1, \\ \mathbf{F}_1 &= \mathbf{F}_1 - \eta\nabla\mathbf{F}_1, \end{aligned} \quad (16)$$

其中， $\nabla\mathbf{F}_1$ 表示 \mathbf{F}_1 的梯度，而 η 是步长。当目标函数 G 的减小量，在 (15) 中定义的，连续两步都小于 ϵ 时，优化过程可以终止。整体优化过程总结在算法 1 中。我们根据不同的统计信息 $\tilde{\mathbf{S}}_i$ 和信道实现 \mathbf{H}_i 计算每跳的预编码矩阵 \mathbf{F}_i 。优化算法的性能根据分类准确率和在 (11) 中定义的平均传输功率 P_t 进行评估。

C. 多块传输策略

在一个实际场景中，中继节点将具有有限数量的天线。这意味着 \mathbf{s}_i 的维度可能非常小，导致单次传输时分类性能下降。在本小节中，我们将考虑每输入图像传输 J 个信道输入，使得带有 M 个天线的系统能够支持 L 维度的输入， \mathbf{s}_i 。不失一般性，我们假设 $L = JM$ 。

为了描述系统操作，我们再次以源节点为例。生成 \mathbf{s}_1 后，源节点将其划分成 J 个块，表示为 $\mathbf{s}_1 = [\mathbf{s}_1^1, \dots, \mathbf{s}_1^J]$ 。

对于第 j 个块，信号 \mathbf{s}_1^j 由 \mathbf{F}_1^j 进行预编码。我们假设信道矩阵 \mathbf{H}_1 在 J 次传输过程中保持不变，第一个中继接收到的信号可以表示为 $\mathbf{y}_1^j = \mathbf{H}_1\mathbf{s}_1^j + \mathbf{n}_1^j$ 。通过堆叠接收到的信号 J 个时隙，我们得到：

$$\mathbf{y}_1 = (\mathbf{I}_J \otimes \mathbf{H}_1) \text{diag}(\mathbf{F}_1^1, \dots, \mathbf{F}_1^J) \mathbf{s}_1, \quad (17)$$

其中 \otimes 表示 Kronecker 积，而 $\text{diag}(\cdot)$ 表示一个对角矩阵，其对角元素为矩阵 $\{\mathbf{F}_1^1, \dots, \mathbf{F}_1^J\}$ 。

为了获得预编码矩阵，我们遵循在第 III-A 和 III-B 节中讨论的相同程序。在训练阶段，对于每个跳段，神经网络权重 $J, \mathbf{W}_1^j \in \mathbb{C}^{M \times M}$ 通过端到端训练获得。然后，给定信道实现 \mathbf{H}_1 ，我们类似地计算 \mathbf{F}_1^j 如同 (15)：

$$\min_{\mathbf{F}_1^j} \|\mathbf{W}_1^j - \mathbf{H}_1\mathbf{F}_1^j\|_F^2 + \lambda_1 \text{tr}(\tilde{\mathbf{S}}_1^j(\mathbf{F}_1^j)^\dagger\mathbf{F}_1^j), \quad (18)$$

其中 $\tilde{\mathbf{S}}_1^j$ 表示 $\mathbf{s}_1^j(\mathbf{s}_1^j)^\dagger$ 的期望，如同在 (14) 中一样。

我们注意到，在考虑的场景中，带有 M 个天线的情况下，每跳有 JM^2 个可学习参数，这比带有 JM 个天线的情况要少，后者将使用 $(JM)^2$ 个参数。

IV. 数值实验

在本节中，我们评估了所提出的多跳 MIMO 网络的分类准确率和功耗 P_t 及 P_r 。图像选自 Fashion MNIST 数据集，其中每个 28×28 灰度图像属于 $K = 10$ 类。除非另有说明，我们考虑跳数为 $N = 2$ ，天线数量为 $M = 49$ 以及噪声方差为 $\sigma^2 = 0.1$ 。为了训练 PrototypeNet，我们采用 Adam 优化器，学习率为 5×10^{-4} 。epoch 数和 batch 大小分别设置为 200 和 32。

A. 性能评估

在图 2(a) 中，我们研究了 P_r 与分类准确率的关系，针对在 (8) 和 (9) 中提出的两种不同的损失函数。对应不同损失函数的 PrototypeNet 是用不同的 λ 值进行训练的。对于每个场景，不同的 λ 值对应于准确性-平均功率 (P_r) 折衷曲线上的不同点。可以看出，使用第二个损失函数训练的网络，即 \mathcal{L}_2 ，严格优于使用 \mathcal{L}_1 的情况，这表明直接最小化目标功耗更为有效。因此，在论文的其余部分，所有结果均基于 \mathcal{L}_2 作为损失函数。我们还评估了一种替代方案，其中 PrototypeNet 在无噪声的情况下进行训练，但在存在噪声的情况下进行测试。相应的性能如图中虚线所示。观察到显著的性能退化，这表明在训练期间考虑噪声是必要的。第 III 节介绍的多块传输策略也进行了评估。特别地，我们考虑每个节点有

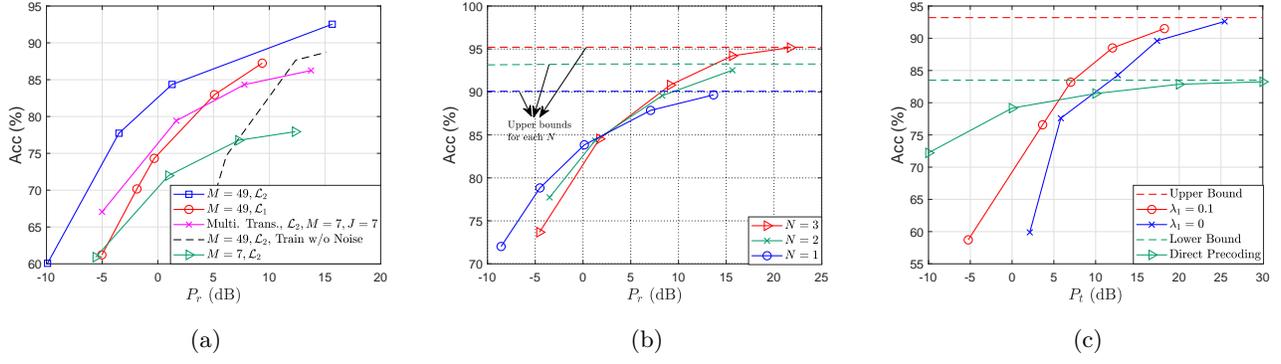


Figure 2: 所提方案在平均功耗方面的分类性能。(a) 使用不同损失函数、天线数量 (M) 和传输次数 (J) 的 PrototypeNet 获得的性能。(b) 具有不同跳数 (N) 的 PrototypeNet 的性能。(c) 多跳 MIMO 网络与直接预编码基线相比的性能。

$M = 7$ 个天线的中继网络。自然, 与 $M = 49$ 天线相比, 这个网络会导致显著的性能下降, 因为这将导致每层神经元数量大大减少的神经网络。另一方面, 当考虑在具有 $M = 7$ 天线的网络上每个输入图像有 $J = 7$ 次传输时, 分类准确性会显著提高。然而, 它仍然劣于 $M = 49$ 的情况。这是因为 $M = J = 7$ 设置可学习参数较少, 并且等效的神经网络架构更为受限。

在图 2(b) 中, 我们研究了跳跃次数 N 对系统性能的影响。与 $N = 1, 2, 3$ 对应的不同的 PrototypeNet 架构使用不同的 λ 值进行训练。虚线代表没有噪声的 PrototypeNet 架构所能达到的准确率上限。可以看出, 当 N 较小时, 准确率上限较低。这与直觉相符: 对于较小的 N , 全连接层的数量较少, 由于神经网络较浅, 性能受限。随着跳跃次数增加, 准确率上限提升。我们还观察到, 在较低的 P_r 下, 较大的 N 的 PrototypeNet 比较小的 N 的效果较差。这是因为当 N 变大时, 噪声在跳跃中累积导致结果变差。然而, 可以看到当 P_r 增加时, PrototypeNet 的准确性接近其上限。

在图 2(c) 中, 我们通过将其与直接预编码基线进行比较来概述所提出的方案在多跳 MIMO 网络中的有效性, 具体如下。对于第 i 跳, 信道实现 \mathbf{H}_i 可以使用奇异值分解 (SVD) 表示: $\mathbf{H}_i = \mathbf{U}_i \boldsymbol{\Sigma}_i \mathbf{V}_i^\dagger$ 。发送器 (R_{i-1}) 使用 \mathbf{V}_i 预编码 \mathbf{s}_i 并通过 MIMO 信道传输它。第 i 次接力 R_i 结合接收到的信号 \mathbf{y}_i 使用 \mathbf{U}_i^\dagger 并应用最小均方误差均衡生成 $\hat{\mathbf{s}}_i$ 的估计值:

$$\hat{\mathbf{s}}_i = (\boldsymbol{\Sigma}_i^\dagger \boldsymbol{\Sigma}_i + \sigma^2 \mathbf{I}_M)^{-1} \boldsymbol{\Sigma}_i^\dagger \mathbf{U}_i^\dagger \mathbf{y}_i. \quad (19)$$

然后对 $\hat{\mathbf{s}}_i$ 进行功率归一化和预编码, 以便通过第 $(i + 1)$ 跳传输。最后, 目标节点估计 $\hat{\mathbf{s}}_N$, 将其输入到 $f_c(\cdot)$ 中, 然后通过 softmax 函数生成分类输出。我们可以观察图

2(c) 发现当 P_t 增加时, 所提出的方案优于直接预编码基线。还观察到使用 $\lambda_1 = 0.1$ 可以获得更优性能。这是因为非零的 λ_1 值试图最小化传输功率, 如第 III 节所述。图中的两条虚线分别表示通过无噪声的 PrototypeNet 使用 $N = 0$ 和 $N = 2$ 全连接层获得的结果, 它们分别作为直接预编码和所提出方案的上限。

请注意, 当 $P_t > 5\text{dB}$ 时, 采用 $\lambda_1 = 0.1$ 的所提出方案优于基线, 这验证了 PrototypeNet 和算法 1 中的优化方法的有效性。

V. 结论

在这篇论文中, 我们研究了一个在多跳 MIMO 网络上用于图像分类的新型 OAC 框架。对于每个跳跃, 中继处理其接收到的信号并计算预编码矩阵以模仿从端到端训练的 PrototypeNet 获得的 FC 层。然后, 通过求解一个优化问题来获取每跳的预编码矩阵。我们进一步考虑了一个具有有限数量天线的实际场景, 并提出了一种多块传输策略以显著提高分类性能。数值实验验证了所提出的 OAC 方案在多跳 MIMO 网络上的分类准确性效果, 并展示了多块传输策略的优势。

REFERENCES

- [1] J. Chen and X. Ran, "Deep learning with edge computing: A review," *Proc. IEEE*, vol. 107, no. 8, pp. 1655–1674, Aug. 2019.
- [2] M. Jankowski, D. Gündüz, and K. Mikolajczyk, "Wireless image retrieval at the edge," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 89–100, Jan. 2021.
- [3] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2022–2035, Mar. 2020.
- [4] M. Mohammadi Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," *IEEE Trans. Signal Process.*, vol. 68, pp. 2155–2169, Mar. 2020.

- [5] G. Zhu, J. Xu, K. Huang, and S. Cui, "Over-the-air computing for wireless data aggregation in massive IoT," *IEEE Wireless Commun.*, vol. 28, no. 4, pp. 57–65, 2021.
- [6] G. Zhu and K. Huang, "MIMO over-the-air computation for high-mobility multimodal sensing," *IEEE IoT J*, vol. 6, no. 4, pp. 6089–6103, 2019.
- [7] G. Sanchez *et al.*, "AIRNN: Over-the-air computation for neural networks via reconfigurable intelligent surfaces," *IEEE/ACM Trans. Netw.*, vol. 31, no. 6, pp. 2470–2482, Dec. 2023.
- [8] G. Reus-Muns, K. Alemdar, S. G. Sanchez, D. Roy, and K. R. Chowdhury, "AIRFC: Designing fully connected layers for neural networks with wireless signals," in *MobiHoc '23*, New York, NY, USA, Oct. 2023, p. 71 – 80.
- [9] Y. Yang, Z. Zhang, Y. Tian, Z. Yang, C. Huang, C. Zhong, and K.-K. Wong, "Over-the-air split machine learning in wireless MIMO networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 4, pp. 1007–1022, Apr. 2023.