

# 自动调制分类：一种支持深度学习的方法\*

Fan Meng<sup>1</sup>, Peng Chen<sup>1</sup>, Member, IEEE, Lenan Wu<sup>1</sup>, and Xiaobin Wang<sup>2</sup>, Fellow, IEEE

<sup>1</sup> Southeast University

<sup>2</sup> Western University

## 摘要

自动调制分类 (AMC) 在民用和军事应用中都起着至关重要的作用, 本文通过深度学习的方法对其进行了研究。传统的 AMC 可以分为基于最大似然 (ML) 的 AMC(ML-AMC) 和基于特征的 AMC。然而, ML-AMC 的计算复杂度较高, 其实际应用难度较大, 人工提取的特征需要专家知识。为此, 本文提出了一种基于端到端卷积神经网络 (CNN) 的自适应 AMC(CNN-AMC) 方法, 该方法自动从长符号率观测序列中提取特征和估计的信噪比 (SNR)。对于 CNN-AMC, 采用单元分类器来适应不同的输入维度。针对 CNN-AMC 直接训练模型复杂、任务复杂的问题, 提出了一种新的两步训练方法, 并引入了迁移学习来提高再训练的效率。在不同的场景下考虑了不同的数字调制方案, 仿真结果表明 CNN-AMC 的性能优于基于特征的方法, 并且更接近最优 ML-AMC。此外, CNN-AMC 对载波相位偏差和信噪比估计误差具有一定的鲁棒性。在并行计算的情况下, 基于深度学习的方法在推理速度方面比 ML-AMC 快 40 到 1700 倍。

**关键词——自动调制分类; 卷积神经网络; 深度学习; 两步训练**

## 1. 引言

为了实现最大可能的传输速率, 自适应调制方案被采用在民用无线通信系统中以达到充分利用时变信号。在这种情况下, 有关通信过程中使用的特

定调制方案的信息必须由发射机通过网络协议与接收机共享, 以协议开销为代价。当接收器具有调制方案识别能力时, 可以消除这种开销。另一方面, 许多军事应用也需要自动发现敌方信号使用的调制方案。这些应用包括信号拦截和干扰。因此, 自动调制分类 (AMC) [1] [2] 对于实现自动接收机配置、干扰抑制和频谱管理的民用和军用应用都具有重要意义。

传统的基于特征的 AMC 算法 [3]、[4] 可以通过数据预处理、特征提取和分类决策三个步骤来实现。由于其良好的统计特征能够以较低的复杂度提供稳健的性能 [5], 基于特征提取的 AMC 方法已经在许多已有的文献 [6]、[7] 中得到了研究, 如循环特征 [8]、高阶矩和累积量 [9]–[10]、小波变换 [11]、[12] 等。在分类决策中, 传统的分类器包括随机森林 [13]、多层感知器 [14]、[15] 和支持向量机 (SVM) [16]、[17]。这些 AMC 是不同特征提取算法和分类的组合, 需要专家知识和人工设计。

文献 [18] 也研究了基于最大似然 (ML) 的 AMC 方法, 其中 ML-AMC 已经被证明在使用数学信道模型的场景中达到了最优的性能。与基于特征的 AMC 不同, ML-AMC 计算所有候选调制的似然函数, 并选择似然值最大的调制方案, 作为端到端模块工作。然而, 较高的计算复杂度使得低成本的实时实现变得极其困难 [19]。此外, 在复杂的环境下, ML-AMC 的精确概率是很难实现的。

在基于特征的 AMC 中, 机器学习方法只是作为特征和多假设之间的映射函数。为了避免特征工程, 深度学习方法 [20] 近年来在算法设计和硬件实现方面发展迅速 [21], 它可以学习比浅层学习方法更复杂的函数。深度学习的快速发展已经在通信领

\*本文为 IEEE TVT 发表的对应该文献的中文翻译版。

域取得了一些成功的应用 [22]- [23], 包括调制分类。在 [24]中, 原始信号的眼图用作基于 Lenet-5 的分类器 [25]的输入, 并巧妙地重新结合图像识别领域的研究, 提出了 AMC 问题。文献 [26]、[27]提出了一种用于无线电调制识别的卷积网络, 并用 RML2016.10b 数据集进行了仿真, 结果表明该方法的分类精度高于专家特征工程方法。然后在 [28]中, 基于 LSTM 的 AMC 被证明在小尺度或中尺度上优于 CNN 模型的过采样接收信号。文献 [29]对长符号率信号进行了研究, 堆叠式自动编码器在增加仿真次数的情况下取得了很好的效果。

本文提出了一种基于深度神经网络 (DNN) 的自适应 AMC 算法, 它可以自动学习从低信噪比的长符号率信号中提取特征, 并采用卷积神经网络 (CNN) 来实现自动调制分类。CNN-AMC 可以在性能损失最小的情况下接近 ML-AMC, 同时速度有显著提高。ML-AMC 被用作我们提出的 CNN-AMC 的基准, 并且还可以生成不同场景下的训练数据。这项工作的贡献概括如下:

- 直接利用低信噪比的长符号率信号, 提出了一种基于深度神经网络的端到端可训练 AMC 算法。与基于特征的 AMC 不同, CNN-AMC 自动学习从原始信号中提取特征, 简化 AMC 设计。此外, CNN-AMC 对 ML-AMC 进行了近似, 相关处理可以并行完成, 加快了推理速度。
- 针对 CNN-AMC 的直接训练问题, 我们提出了一种包括预训练和微调的两步训练方法。对于不同调制方式和通信信道场景的不同组合, 还提出了迁移学习来进一步提高再训练效率。
- 通过将经过长预处理的观测序列分成单元大小的信号段, 引入单元输入维数以适应不同的输入维数, 同时并行实现分块计算。

本文的其余部分组织如下。第二节概述了经典决策理论框架下的调制分类问题。第三节给出了具有多输入特征的 CNN-AMC 的结构, 以及相应的两步训练、迁移学习和单元分类器。然后, 在第四节中, 对所提出的 CNN-AMC、ML-AMC 和基于特征的方法在不同环境下的信噪比进行了测试, 并对仿

真结果进行了分析。第五节给出了结论和讨论。

## 2. 系统模型

### 2.1. 问题表述

在不损失一般性的情况下, 接收基带信号的复包络可以表示为:

$$y(t) = x(t; \mathbf{u}_k) + v(t) \quad (1)$$

其中  $v(t)$  是具有零均值和双边功率谱密度为  $N_0/2$  的复加性高斯白噪声 (AWGN)。无噪声信号  $x(t; \mathbf{u}_k)$  通过综合形式被建模为:

$$x(t; \mathbf{u}_k) = ae^{j(2\pi\Delta f t + \theta_c)} \sum_{n=0}^{N-1} x_n^{k,i} g(t - nT_s - \varepsilon T_s) \quad (2)$$

其中,  $\mathbf{u}_k$  定义为多维参数集。其中, 在第  $k$  次调制方案下, 一串未知信号和信道变量是随机的或确定的, 并且它被给出为:

$$\mathbf{u}_k = \left\{ a, \theta_c, \Delta f, \{x_n^{k,i}\}_{i=1}^{M_k}, h(t), \varepsilon \right\} \quad (3)$$

在 (2) 和 (3) 中使用的符号如下所示:

- $a$ , 信号幅度。
- $\theta_c$ , 因传播延迟及初始载波相位引入的固定相位偏移。
- $N$ , 接收序列中的符号数, 即观测序列空间。
- $x_n^{k,i}$ , 第  $k$  调制方案下的第  $i$  星座点, 其中符号均匀分布,  $i \in \{1, \dots, M_k\}, k \in \{1, \dots, C\}$ , 标记  $M_k$  分别表示使用该调制的符号数目,  $C$  是候选等概率调制方案的数目。调制的码元通常用单位功率进行归一化。

$$\frac{1}{M_k} \sum_{i=1}^{M_k} |x_n^{k,i}|^2 = 1 \quad (4)$$

此外, 数据块中的符号被假定为独立同分布 (I.I.D.)。

- $\Delta f$ , 去除载波后的剩余载波频率。
- $T_s$ , 符号间隔。
- $g(t)$ , 剩余通道  $h(t)$  和脉冲整形函数  $p(t)$  的复合效应。它由方程  $g(t) = h(t)*p(t)$  表示, 其中

\* 是卷积算子。通常,  $p(t)$  是持续时间  $t_s$  的升余弦脉冲函数。

- $\varepsilon$ , 发送器和接收器之间的时序偏移的归一化历元,  $0 < 1$ 。

本文将接收信号表示为采样复数基带表示, 并考虑了  $\frac{E_s}{N_0}$  的信噪比。以符号间隔  $t_s$  采样的离散观测序列可以写为  $y = [y_0 \dots y_{N-1}]^T$ , 其中上标  $T$  是转置运算符。接收信号的每个采样时刻都是接收端脉冲整形匹配滤波器的输出, 可以表示为:

$$y_n = \frac{1}{T_s} \int_{nT_s}^{(n+1)T_s} y(t)p(t - nT_s) dt \quad (5)$$

在盲分类的情况下, 发送信号的脉冲形状  $p(t)$  对于接收器是未知的。同样, 瞬时  $y_n$  通常归一化为单位功率。

$$y_n = \frac{y_n}{\sqrt{\frac{1}{N} y^H y}} \quad (6)$$

其中上标  $H$  是共轭转置运算符。在分类器中, 需要估计符号信噪比, 这相当于既知道  $a$  又知道  $N_0$ 。

假设  $H_k$  下  $y_n$  的条件概率密度函数 (PDF) 可以表示为:

$$\begin{aligned} p(y_n | H_k, u_k) &= \sum_{i=1}^{M_k} \frac{1}{M_k \sqrt{\pi N_0}} \exp\left(-\frac{|y_n - x_n^{k,i}(u_k)|^2}{N_0}\right) \\ &\propto \frac{1}{M_k} \sum_{i=1}^{M_k} \exp\left(-\frac{|x_n^{k,i}|^2 - 2\Re\{y_n^* x_n^{k,i}(u_k)\}}{N_0}\right) \end{aligned} \quad (7)$$

参数集  $u_k$  中的元素是具有已知 PDFs 的确定性或随机变量。因此, 可以通过取 (7) 的统计平均值来获得观测序列的边际密度函数, 并且它被给出为:

$$\Gamma(y_n | H_k) = \mathbb{E}_{u_k} \{\Gamma(y_n | H_k, u_k)\} \quad (8)$$

其  $\mathbb{E}\{\cdot\}$  是期望运算符。

假设信道环境变化缓慢, 设置  $u_k$  中的参数在整个观察期内是静态的。此外, 归一化历元  $\varepsilon = 0$  且噪声为白色, 因此  $y$  中的元素服从独立同分布 (I.I.D), 函数  $p(y_n | h_k, u_k)$  表示时刻  $n$  处单次观测的概率密度, 整个观测序列的联合似然函数可描述为:

$$\Gamma(y | H_k) = \prod_{n=0}^{N-1} p(y_n | H_k) \quad (9)$$

数值计算中的浮点溢出可以由一个大的  $N$  引入, 因此通常采用 (9) 的等效对数形式, 它由下式给出:

$$L(y | H_k) = \ln \Gamma(y | H_k) \quad (10)$$

根据最大似然准则, 给定观测序列  $y$ , 最终选择与第  $k$  个调制方案对应的最可能的假设  $H_k$ , 其似然值最大:

$$\hat{k} = \arg \max_{k \in \{1, \dots, C\}} \Gamma(y | H_k) \quad (11)$$

其中, 顶端标记  $(\cdot)$  表示估计值。

## 2.2. 正确概率

条件正确概率  $P_{cc}(y|H_K)$  定义为:

$$P_{cc}(y | H_k) = \frac{\Gamma(y | H_k)}{\sum_{k'=1}^C \Gamma(y | H_{k'})} \quad (12)$$

因此, 总体正确概率  $P_c$  被表示为:

$$P_c = \frac{1}{C} \sum_{k=1}^C P_{cc}(y | H_k) \quad (13)$$

观测序列  $y$  的分布为  $f_Y(y)$ , 然后结合 (12) 和 (13),  $P_c$  的期望值由下式给出:

$$\begin{aligned} \mathbb{E}_Y \{P_c\} &= \int_y P_c f_Y(y) dy \\ &= \frac{1}{C} \sum_{k=1}^C \int_y \frac{\Gamma(y | H_k)}{\sum_{k'=1}^C \Gamma(y | H_{k'})} \Gamma(y | H_k) dy \end{aligned} \quad (14)$$

假设参数集  $u_k$  是已知的, 但实际上, 它通常是在建模或估计时会有一些偏差。然后将 (14) 表示为:

$$\mathbb{E}_Y \{P_c\} = \mathbb{E}_{u_k, \hat{u}_k, Y} \left\{ \frac{1}{C} \sum_{k=1}^C \int_y \frac{\Gamma(y | H_k, \hat{u}_k)}{\sum_{k'=1}^C \Gamma(y | H_{k'}, \hat{u}_{k'})} \right\} \quad (15)$$

其中  $\hat{u}_k$  是  $u_k$  的估计值。我们很难得到 (14) 和 (15) 的闭合解, 但可以用蒙特卡罗方法得到数值近似。

## 2.3. 信号段

将归一化似然向量定义为  $\xi^N = [\xi_1^N, \dots, \xi_C^N]^T$ , 其中对应的调制元素为  $\xi_k^N = P_{cc}(y|H_k)$ , 显然  $\sum_{k=1}^C \xi_k^N = 1$ 。因为观测序列中的时刻是独立同分布 (I.I.D), 所以整个向量可以分割成  $S$  个连续的子序

列，对应的长度为  $N_s$ ， $N = \sum_{s=1}^S N_s$ 。此外，结合 (9) 和 (12)， $\xi_k^N$  可以写为：

$$\begin{aligned}\xi_k^N &= \frac{\prod_{s=1}^S \Gamma(y^{N_s} | H_k)}{\sum_{k'=1}^C \prod_{s=1}^S \Gamma(y^{N_s} | H_{k'})} \\ &= \frac{\prod_{s=1}^S \xi_k^{N_s}}{\sum_{k'=1}^C \prod_{s=1}^S \xi_{k'}^{N_s}} \\ &= \gamma_k(\xi^{N_1}, \dots, \xi^{N_S})\end{aligned}\quad (16)$$

其中  $\xi^{N_s}$  表示分段  $s$  的归一化似然向量，当  $N$  较大时有意义，我们将其分成几个分段进行分块计算。更重要的是，(16) 表示分段等同于原来的分类任务，分块计算的设计可以灵活可实现。

### 3. 基于 CNN 的分类器

我们提出了一种新的端到端 DNN 使能 AMC [30]- [31]，它与 ML-AMC 非常接近，并显著提高了推理速度。

#### 3.1. 在 AMC 中使用 CNN 的原因

假设信道环境是时间不变和频率非选择性的，则观测序列  $y$  中的元素是独立同分布 (I.I.D)， $y$  的一些特性必须注意：

- 任意时间间隔内的分段  $y^{N_s}$  的统计特征是相同的，即：

$$\begin{aligned}\mathbb{E}\{L(y_i^{N_s} | H_k, u_k)\} &= \mathbb{E}\{L(y_j^{N_s} | H_k, u_k)\}, \\ 0 \leq i, j &< N - N_s,\end{aligned}$$

其中， $\mathbf{y}_i^{N_s} = [y_i, \dots, y_{i+N_s-1}]^T$

- 信号序列  $y_n$  可以被分成  $S$  个连续的子序列段，并且似然函数可以表示为  $L(\mathbf{y}_n | H_k) = \sum_{s=1}^{N_s} L(\mathbf{y}^{N_s} | H_k)$ 。
- 任意交换  $y$  中的元素任意次，得到  $\mathbf{y}$  的似然函数为  $L(\mathbf{y}' | H_k) = L(\mathbf{y} | H_k)$ 。

性质 1 表示信号在时域中均匀分布。具有卷积核的信号段在不同时刻的冲激响应是相同的，因此可以在任何时隙提取或检测特征。性质 2 证明似然函数是合成的，并且可以由堆叠的级联表示模型来近似。因此，具有局部感受野的 CNN 比全连接的

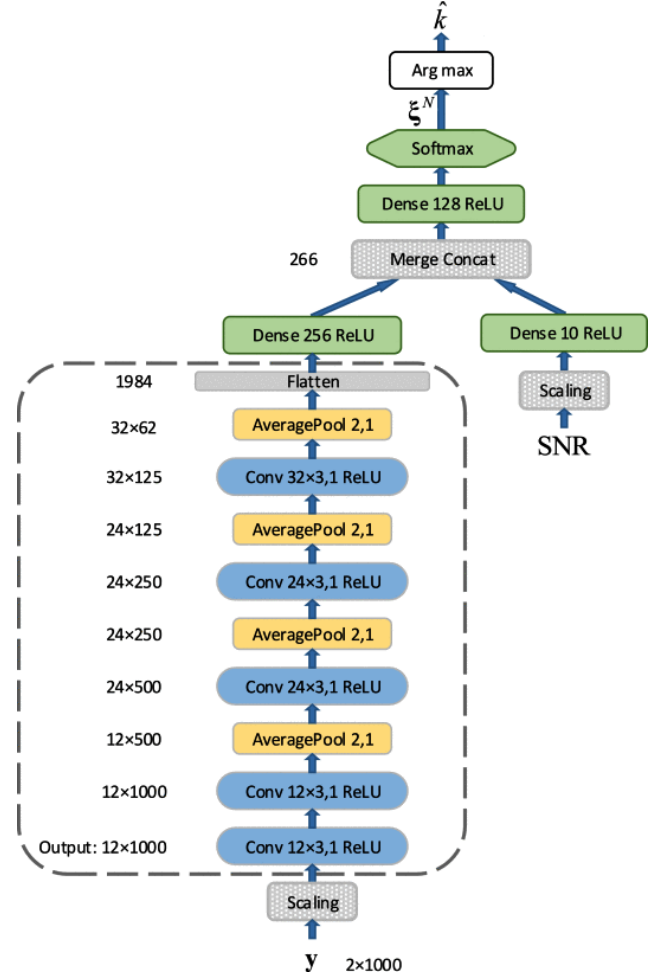


图 1. 以矢量和标量作为输入的基于 CNN 的调制分类器的图示。文本 ‘Conv 12×3, 1 ReLU’ 表示在该卷积层中，核数、核大小和步长分别为 12、3 和 1。类似地，文本 ‘AveragePool 2, 1’ 表示在该平均池化层中，窗口大小和步幅分别为 2 和 1。

NN 更适用。此外，在给定较大的  $N$  的情况下，对于全连接模型，训练过程中的过拟合问题变得更加严重。性质 3 表示信号是作为时间序列进行采样的，但它与时间无关，因此不建议使用递归神经网络 (RNN) 来解决此问题。

#### 3.2. 网络架构

我们设计的 CNN-AMC 的结构如图1所示，输入由两部分组成：维数  $N = 1000$  的预处理观测序列  $y$  和与 ML-AMC 相同的估计符号信噪比。然而， $y$  是一个原始信号向量，需要进行卷积才能生成更高



级别的信息，而 SNR 只是一个标量，不能卷积。不同类型的信号不能简单地串联起来作为 CNN-AMC 的输入。取而代之的是使用具有多个输入的混合神经网络。

为了将复矢量  $y$  变换成实数数据，将实部和虚部分开，然后将其转换成形状  $2 \times N$ 。例如，对于第一卷积层  $l = 1$ ，输入矢量可以表示为  $\mathbf{X}^{l-1} = \mathbf{y} \in \mathbb{R}^{N_{l-1} \times V^{l-1}}$ ，其中矢量数量和维数分别为  $N_{l-1} = 2$  和  $V_{l-1} = 1000$ 。输出向量的大小列在图1中相应层框的左侧。对于具有  $N_{l-1}$  输入和  $N_l$  输出向量的一维卷积层  $l$ ，存在  $N_l \times N_{l-1}$  卷积核（也称为权重矩阵）和  $N_l$  偏置，核大小用  $k_l$  表示。给出上一层的输出向量  $\mathbf{x}_i^{l-1}, i \in \mathcal{N}_{l-1} = \{1, \dots, N_{l-1}\}$  当前层的第  $i$  个输出向量表示为：

$$\mathbf{x}_i^l = f \left( \sum_{j=1}^{N_{l-1}} \mathbf{x}_j^{l-1} * \mathbf{k}_{i,j}^l + \mathbf{b}_i^l \right) \quad (17)$$

其中，采用整流线性单元 (RELU) [32] 作为快速非线性激活函数：  $f(x) = \max(0, x)$ ，并且  $*$  是卷积操作，在卷积层  $l$  中有  $N_l \times N_{l-1} \times K_l + N - l$  可训练参数。池化层也称为子采样层，它产生输入向量的下采样版本。形式上，我们将第  $i$  个输出向量设为：

$$\mathbf{x}_i^{l+1} = \text{down}(\mathbf{x}_i^l) \quad (18)$$

其中， $\text{down}(\cdot)$  表示子采样函数，并且平均池化 [33] 用于通过计算加窗输入向量的平均值来降低维数。在堆叠卷积层和池化层之后，输出向量被 Flatten 层压缩成向量。然后使用 Dense 层将该向量编码成包含提取的高层信息的 256 维向量。对于具有输入矢量  $\mathbf{X}^{l-1}$  的 Dense 层  $l$ ，输出矢量可以描述为：

$$\mathbf{x}^l = f(\mathbf{W}^l \mathbf{x}^{l-1} + \mathbf{b}^l) \quad (19)$$

其中  $\mathbf{W}^l \in \mathbb{R}^{N_l \times N_{l-1}}$  和  $\mathbf{b}^l \in \mathbb{R}^{N_l}$  分别是权重矩阵和偏置。这是一个全连接层，可训练参数的数量为  $N_l \times N_{l-1} + N_l$ 。然后将该向量与标量信噪比的矢量化表示级联。在另一 Dense 层之后，然后，由激活函数为 Softmax 的输出层产生归一化似然向量，即  $\hat{\xi}^N$ 。输出  $\hat{\xi}^N = [\hat{\xi}_1^N, \dots, \hat{\xi}_C^N]^T$ ，元素  $\hat{\xi}_i^N$  定义为：

$$\hat{\xi}_i^N = \frac{e^{x_i}}{\sum_{j=1}^C e^{x_j}} \quad (20)$$

其中  $x_j$  是第  $j$  个预激活输出。最后，CNN-AMC 的表示可以描述如下：

$$\hat{\xi}^N = F(\mathbf{y}, \text{SNR}; \theta) \quad (21)$$

其中  $\theta$  是网络参数， $F(\cdot)$  是基于 CNN 的分类器的总体函数。在该深层模型中，特征提取由 CNN 部分自动实现，高层主要处理输入输出数据之间的映射关系。通常，CNN-AMC 是目标函数 (9) 的强大非线性逼近器  $F(\cdot)$ 。完整的训练/测试和仿真源代码可以从 [https://github.com/mengxiaomao/cnn\\_amc](https://github.com/mengxiaomao/cnn_amc) 下载。

### 3.3. 两步式训练

在一般的训练过程中，CNN-AMC 是逐时段训练的，并且在一个时段内，所有训练样本被用来训练 CNN-AMC 一次。在一个历元内，对整个训练集进行变换，分成大小为  $N_b$  的批次，逐批次地对模型进行训练。本文采用交叉熵损失函数，将训练批次的损失函数表示为：

$$L(\theta) = -\frac{1}{N_b} \sum_{i=1}^{N_b} \xi_i^N \log \hat{\xi}_i^N + (1 - \xi_i^N) \log (1 - \hat{\xi}_i^N) \quad (22)$$

其中  $\xi_i^N$  表示目标输出。该损失函数通过随机梯度下降 (SGD) 算法最小化，并且模型参数被迭代地更新为：

$$\theta_{t+1} = \theta_t - \eta \Delta L(\theta_t) \quad (23)$$

其中  $\eta$  表示学习率，操作  $\Delta$  表示梯度运算符。作为 SGD 算法的一种变体，本文采用了自适应矩估计 (ADAM) [34] 算法，并采用了缺省设置。ADAM 优化器通过对低阶动量的自适应估计，动态地自适应调整  $\eta$  以加快收敛速度，并被证明是有效的，特别是对于深部模式。记录每个训练时段的参数集  $\theta$  和验证损失，并定义最优的验证损失最小的 CNN-AMC。仿真中可以产生不同信噪比下不同长度的接收观测序列，ML-AMC 还可以产生大量的输出数据。

然而，在我们的实验中，由于模型复杂，目标函数复杂，基于 CNN 的分类器的直接训练具有挑战性。通过大量的训练数据和足够的迭代次数，我们发现损失函数  $L(\theta)$  不能下降，说明该模型从一开始

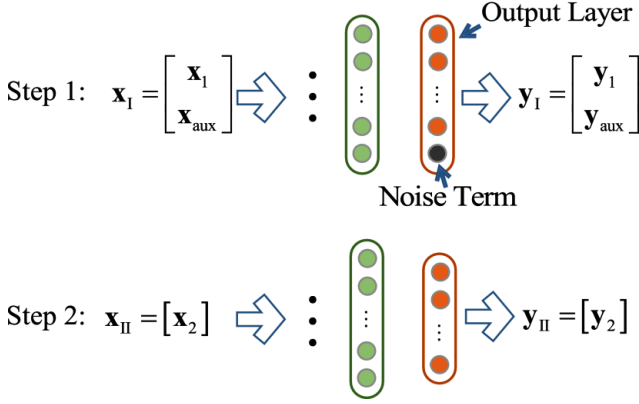


图 2. CNN-AMC 的两步训练说明。在步骤 1 中，所有初级输出元素用橙色圆圈表示，而附加噪声项用灰色圆圈表示。在步骤 2 中去除高斯噪声的输出元素。步骤 1 中的输入和输出用  $x_I$  和  $y_I$  表示，类似地，步骤 2 用  $x_{II}$  和  $y_{II}$  表示。

表 1. 两步训练的仿真参数

训练步骤	步骤 1	步骤 2
Epoch 数量	20	240
SNR 范围/dB	[0, 10]	[-6, 12]
训练数据量	79,200	228,060
验证数据量	8,800	25,340
使用 ML 分类器的输出	No	Yes

就停留在高维参数优化上。虽然最后一项任务很困难，但对于模型来说，学习更简单的函数会容易得多。有一个使用辅助数据训练基于 CNN 的分类器的简单但有用的技巧。如图 2 所示，整个培训过程分为两个步骤。

- 预训练：此步骤旨在帮助 CNN-AMC 在训练开始时收敛，因此训练数据的规模很小，如 1 所示。本步骤中增加了另一个带灰色圆圈的输出项，其对应的类别是零均值的高斯白噪声，而不是其他调制方案。噪声的辅助训练样本量与任何其他调制方案相同，其 I/O 数据用  $x_{aux}$  和  $y_{aux}$  表示。输出向量被设置为二进制。在仿真测试中，损失函数  $L(\theta)$  减小，梯度方差变小，训练一段时间后收敛。存储验证损失最小的参数  $\theta$ ，以便在下一次训练中进行模型初始化。
- 微调：首先，第一步，CNN-AMC 加载存储的参数  $\theta$ ，辅助输出项在最终分类中是冗余的，所以

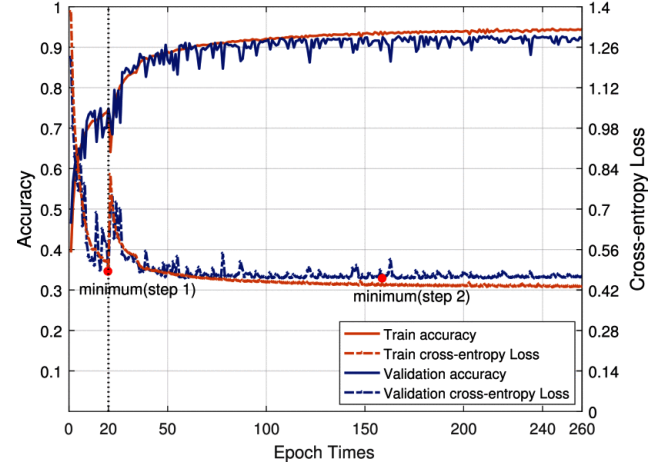


图 3. 准确性和损失的培训/验证曲线。X 轴表示训练时期。验证和训练曲线用深蓝色和橙色线条绘制，精确度和损耗曲线用实线和虚线表示。

用主输出层替换顶层，然后随机初始化。在该步骤中，仅生成调制方案的样本，并且 ML-AMC 以归一化似然向量  $\xi^N$  的形式产生训练数据的输出，这与第一步不同。精心设计的训练数据帮助 CNN-AMC 逼近 ML-AMC，仿真结果表明，使用 ML-AMC 生成的数据比使用普通二进制标记的  $P_c$  性能更接近 ML-AMC。类似地，验证损失最低的 CNN-AMC 在以下模拟中被存储、重新加载和测试，详细设置如表 1 所示。

表 2 给出了两步训练方法的总结。在图 3 中，绘制了完整的两步训练过程中的损失和精度的训练/验证曲线。一般来说，损耗曲线在两个阶段都是下降的，而精度曲线是上升的。在第二步中，训练损失在开始时急剧下降，然后稳步下降。验证损失曲线与训练损失曲线具有相似的趋势。随着历元的增加，训练时间趋于平稳，抖动较小，很难低于 0.47，即训练处于饱和区。存储具有最小验证损失的相应  $\theta$ ，并用红点标记步骤 2 中的最小值。

复平面上的辅助高斯噪声的似然函数是准凸的，这是一种比较直观的逼近函数。从 (7) 中我们知道，一个观测时刻的条件概率密度函数是具有不同均值和相同方差的高斯分布的加权和。因此，对于已经学习了高斯噪声函数的 CNN-AMC 来说，对其他调制方案的更复杂的 PDF 的近似变得更加容易。在学

表 2. 两步训练综述

步骤 1:

- 1) 生成训练数据，并且包括附加高斯噪声的样本。
- 2) 随机初始化 CNN-AMC 参数。
- 3) 训练 CNN-AMC，以最小的验证损失存储模型参数。

步骤 2:

- 1) 产生训练数据，并从 ML-AMC 获得目标输出。
- 2) 更换 CNN-AMC 的顶层并随机初始化；从存储器加载其他模型参数。
- 3) 训练 CNN-AMC，以最小的验证损失存储模型参数。

习了符号的近似后，根据 (9)，目标函数变得更容易获得。在第一步中，初始收敛问题得到了解决。通过预训练，CNN-AMC 在第二步中被进一步训练为尽可能接近最优 ML-AMC。

### 3.4. 迁移学习

上述两步训练处理的问题是：给定具有随机参数初始化的 CNN-AMC，以及如何训练它以利用给定的训练和验证数据来分类调制方案。事实上，一个训练好的 CNN-AMC 可以应用于一个特定的任务，例如具有几种候选调制模式的相干环境。一旦场景或调制集改变，CNN-AMC 需要用新的训练数据重新训练。

事实上，没有必要在迁移学习的帮助下从一开始就重新启动新的培训过程 [35]。如前所述，CNN-AMC 的 CNN 部分从经过预处理的原始观测序列中产生更高层的信息。特征提取可以在不同的任务中普遍使用，并且可以利用一些已经学习到的先验知识或经验而不是纯粹的空白来更容易地进行训练。这里，我们以两个案例为例：

- 多种调制方式：在不同的情况下，根据分类任务的要求，增加或删除一些调制方式。除了新的训练数据外，CNN-AMC 的输出层需要更改为新的相应类别。因此，除了最上层的参数外，所有 CNN-AMC 参数都可以作为初始化加载，而不是随机设置。
- 不同的环境：在预处理期间，载波相位补偿包括在相干情况下，但不包括非相干情况。使用提出的两步法可以用新的非相干训练数据来训练

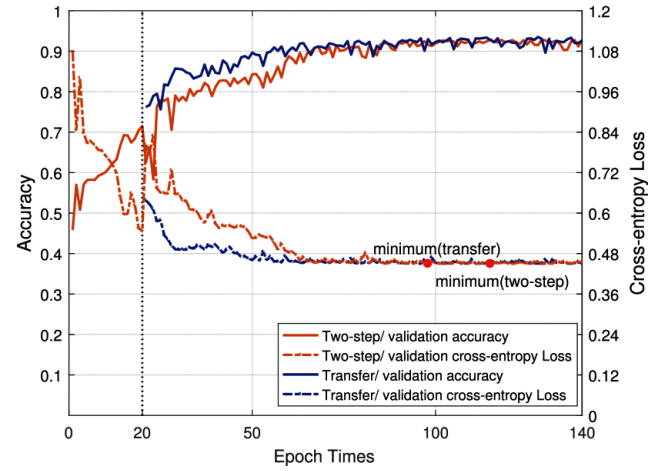


图 4. 验证数据的丢失和准确度随历元时间的变化曲线。与两步训练算法相比，采用转移学习的算法达到饱和区域所需的时间更短，并且减少了预训练步骤。

CNN-AMC。此外，在一致性情况下学习的基于 CNN 的分类器可以作为初始模型使用这些数据训练，所有的模型参数都是从一致性情况下训练的模型参数中加载的。我们给出了一个使用迁移学习的例子，与之对应的是使用两步训练的例子。如图4所示，损失曲线的下降速度比使用两步训练的速度快。此外，可以省去训练前的步骤，进一步降低了训练周期的时间成本。

虽然迁移学习有助于提高训练效率，但它是通过两步学习来发现的。

### 3.5. 单位分类器

为了对低信噪比的信号实现高分类精度，序列空间  $N$  可以高达几个甚至数万。随着  $N$  的增加，需

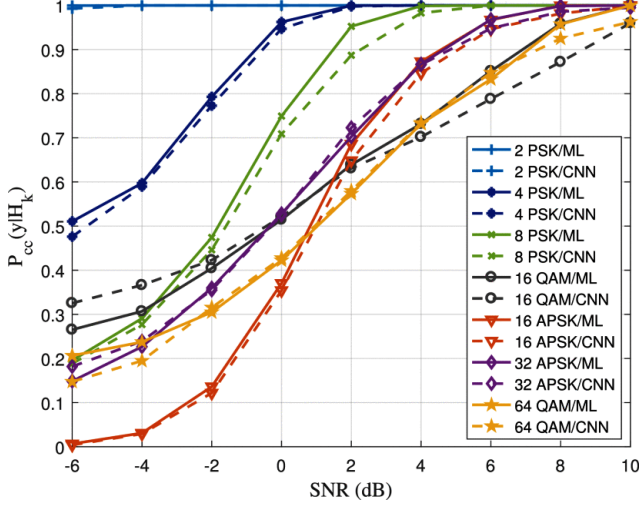


图 5. 7 种调制方式的  $P_{cc}(y|H_k)$  曲线, 信噪比范围为  $[-6, 10]$  dB. ML 分类器和基于 CNN 的分类器分别用实线和虚线描述。

要更多的训练样本来克服过拟合问题, 从而使训练变得更加困难。问题是在有限的存储空间和计算资源下, 分类及其对应的  $N$  的训练可能很难, 甚至不切实际。另一方面, 设计一种输入空间可变的分类器也是一项具有挑战性的工作。

根据前面在第 3.3 小节中的分析, 大  $N$  的输入  $y$  可以被分成具有单位长度的段。将输入信号  $y$  为单位输入维数的基于 CNN 的分类器定义为单元分类器。从理论上讲, 在相干环境下, 该分段与完全观测分段具有相同的分类性能。更重要的是, 单元分类器关于训练、缓存和计算的比例是可调整的。因此, AMC 的设计可以灵活地适应硬件, 与序列空间  $N$  无关, 也可以对所有的单元分类器并行进行推理。

## 4. 仿真结果

通过研究, AMC 的通用序列空间  $N$  从数百到数千不等, 并考虑到硬件条件, 对观测空间  $N = 500$  (命名为 CNN-500) 和  $N = 1000$  (命名为 CNN1000) 的 CNN-AMC 进行了测试。此外, 将 ML-AMC 与基于 9 个高阶累积量的基于特征的 AMC [36] (称为 Cu-AMC) 进行了比较。采用二阶、四阶和六阶累积量作为特征, 采用两层隐层神经网络作为分类器。考虑了几种调制方案。总体正确概率和条件正确概率对信噪比的影响是需要研究的两个主要问题。此外,

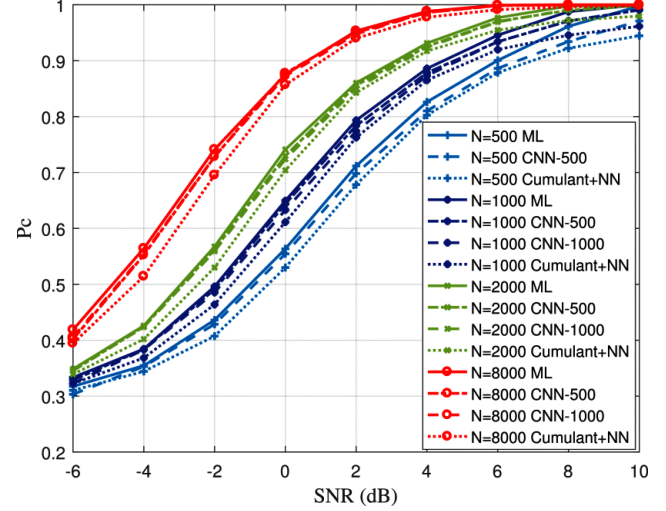


图 6. 不同序列空间  $N$  对信噪比的  $P_c$  曲线。ML-AMC、CNN-500 和 CNN-1000 分别用实线、点状线和虚线表示。

还必须考虑对估计误差的鲁棒性、推理时间开销和系统存储需求等多个方面。

### 4.1. 相干

仿真从相干环境开始, 考虑了先验概率相等的 7 种调制方式, 即 2PSK、4PSK、8PSK、16QAM、16APSK、32APSK 和 64QAM。假设接收器具有精确的参数集  $u_k$  的知识, 包括恒定相位偏移  $\theta_c$  和 SNR, 但是需要推断调制方案。除了 (6) 的功率归一化之外, 在预处理周期中用共轭相位偏移来补偿观测序列。

$$y = ye^{-j\theta_c} \quad (24)$$

结合 (7)、(8) 和 (9), 假设  $H_k$  下的似然函数由下式显式给出:

$$\Gamma(y | H_k) \propto \frac{1}{M_k} \prod_{n=0}^{N-1} \sum_{i=1}^{M_k} \exp \left( \frac{|x_n^{k,i}|^2 - 2\Re \{y_n^* x_n^{k,i}\}}{N_0} \right) \quad (25)$$

因此, 可以得到归一化似然向量  $\xi^N$ , 并将其与观测序列  $y$  相结合, 可以为 CNN-AMC 生成大量的训练和验证数据。

- 正确概率: 支持深度学习的 AMC 比 ML-AMC 快得多。然而, 作为最优 AMC 的近似值, 它在本仿真研究的  $P_{CC}(y|H_K)$  和  $P_c$  方面可能会受到一定程度的性能下降。



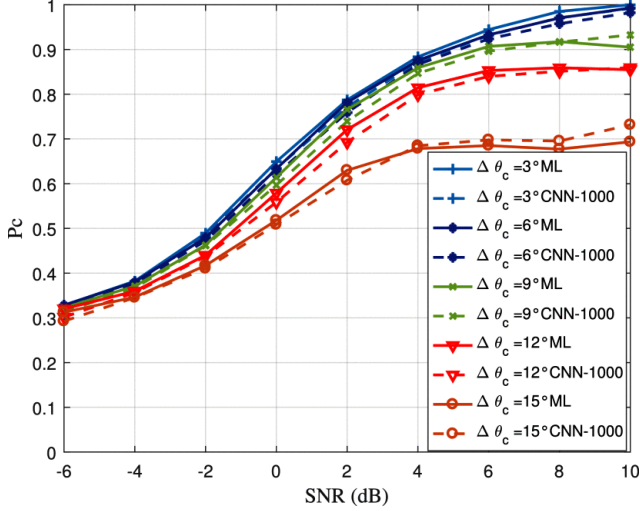


图 7. 集合  $\{3^\circ, 6^\circ, 9^\circ, 12^\circ, 15^\circ\}$  中不同相移  $|\Delta\theta_c|$  估计误差的  $P_c$  曲线。ML 和 CNN-1000 分类器分别用实线和虚线表示。

首先, 7 种调制的条件正确概率  $P_{CC}(y|H_K)$  曲线如图 5 ( $N = 1000$ ) 所示。ML-AMC 用实线表示, CNN-1000 用虚线绘制。显然, 2PSK 是我们集合中最容易识别的调制, 即使在低信噪比的情况下, 它的条件正确概率也是 100%。与其他调制方式相比, 4PSK 和 8PSK 更容易识别, 当  $SNR = 2dB$  和  $SNR = 4dB$  时, 其  $P_{CC}(y|H_K)$  分别达到 1.00。在高信噪比 ( $SNR = 10dB$ ) 下, ML-AMC 几乎可以毫无错误地识别出所有的调制方式。

通常, 将 CNN-1000 与 ML-AMC 进行比较时,  $P_{CC}(y|H_K)$  性能略有下降。在某些 SNR 区域, 当使用 CNN-1000 时, 对于某些调制,  $P_{CC}(y|H_K)$  高于 ML-AMC。例如, 在信噪比从  $-6$  到  $-2dB$  并且调制方案是 16QAM 的情况下, CNN-1000 似乎比 ML-AMC 更好。这是因为相应的条件概率被高估, 代价是某些其他调制方案 (例如 64QAM) 的较低的正确分类概率。CNN-1000 的  $P_c$  只能无限逼近 ML-AMC, 不能超过最优值。

ML-AMC、CNN-500、CNN-1000 和 Cu-AMC 的总体正确概率曲线如图 6 所示。当  $N = 500$  和  $N = 1000$  时, CNN-AMC 和 ML-AMC

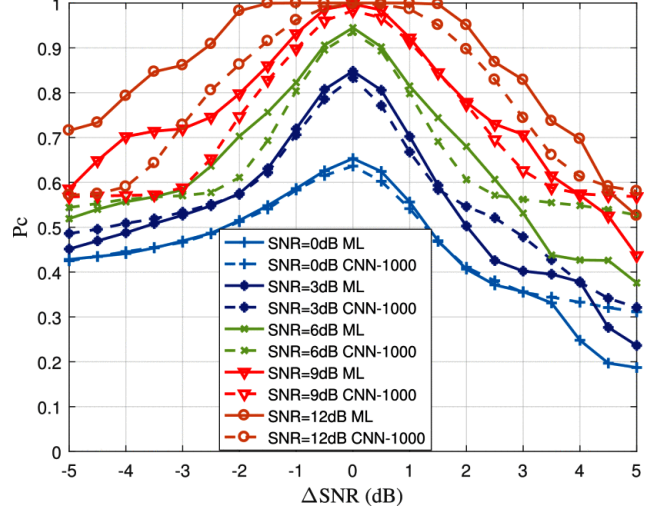


图 8. 在集合  $\{0, 3, 6, 9, 12\}$  中不同信噪比的  $P_c$  曲线, 估计误差  $\Delta SNR$  范围为  $[-5, 5]dB$ 。ML-AMC 和 CNN-1000 分别用实线和虚线表示。

之间大约有 1% 到 2% 的  $P_c$  性能下降。此外, 当序列空间  $N$  为单位输入长度的整数倍时, 使用 CNN-500 和 CNN-1000 作为单元分类器, 当  $N = 2000$  和  $N = 8000$  时, 逼近几乎非常接近 ML-AMC 曲线。仿真结果表明, 在精确知道参数集的情况下, 基于 CNN 的分类器能够很好地逼近 ML-AMC。另一方面, 在相干场景下, 它的计算速度比使用 7 种调制方式的 ML 使能方法快约 150 170 倍。此外, 基于 CNN 的分类器性能优于 Cu-AMC, 当  $P_c > 90\%$  时, 基于特征的分类效果不如 CNN-AMCS。更重要的是, 实验结果表明, 在这种情况下, 支持深度学习的方法比基于人工特征提取的方法具有更好的性能。

- 估计误差: 以前的仿真都是在完全知道固定相位偏移和信噪比这两个参数的前提下给出的。然而, 估计误差通常是不可避免的。在这一部分中, 对不同程度的估计误差进行了仿真, 并采用了变量控制的方法。在第一个仿真中, 研究了载波相位偏差。由于一般星座图是对称的先验知识, 正负偏差对载波相位的影响是等价的。我们使用  $|\Delta\theta_c|$  表示相位误差, 使用  $|\Delta\theta_c| \in \{3^\circ, 6^\circ, 9^\circ, 12^\circ, 15^\circ\}$

结果如图 7 所示。一般来说,  $P_c$  随  $|\Delta\theta_c|$  的

升高而减小。当  $|\Delta\theta_c| < 6^\circ$  时，分类性能下降很小。当  $|\Delta\theta_c| \in [6^\circ, 9^\circ]$  时，ML-AMC 明显恶化，当信噪比在  $[8, 10]dB$  ( $|\Delta\theta_c| = 9^\circ$ ) 范围内增大时，ML-AMC 更加恶化。当  $SNR > 6dB$ ， $|\Delta\theta_c| = 12^\circ$  时， $P_c$  保持在 0.86 左右。当  $|\Delta\theta_c| = 15^\circ$  时， $P_c$  甚至不能超过 0.7。作为比较，基于 CNN 的分类器与 ML-AMC 大致相同，在某些情况下可以执行得更好。在  $|\Delta\theta_c|$  高达  $15^\circ$  的极端情况下，基于 CNN 的分类器的  $P_c$  比 ML-AMC 的  $P_c$  要高，这意味着 NN 使能方法更健壮。

在第二次仿真中，研究了信噪比估计误差对系统性能的影响。该错误可能会导致错误地假设星座点周围的数据集中，并且这些字母表的中心也用有偏差的系数进行归一化。

如图8所示，x 轴为  $-5$  到  $5dB$  的 SNR 误差，定义为  $\Delta SNR = \widehat{SNR} - SNR$ ，y 轴为  $P_c$ 。实 SNR  $\in \{0, 3, 6, 9, 12\}dB$  的曲线如下贴上不同颜色和记号的标签。由此可见，分类器对信噪比误差非常敏感，即使是  $0.5dB$  的估计偏差也会导致推理性能下降。此外，高估的信噪比比低估的信噪比对分类器的危害更大。我们还注意到，CNN-AMC 的曲线比 ML-AMC 更平滑，在很多情况下，ML-AMC 和基于 CNN 的分类器的  $P_c$  相差可以超过 10%。在  $SNR = 12 \pm 1.5dB$  范围内，ML-AMC 的  $P_c$  仍为 1.0，表明性能饱和可以在一定程度上抵抗 SNR 估计误差。同时，在所有 5 个比较中，相应的基于 CNN 的分类器的性能是最差的。我们认为，这种退化可以归因于训练数据不足。在表1中，训练数据在  $[-6, 12]dB$  范围内采样，这意味着包括该范围在内的 CNN-AMC 没有被训练。

为了使分类器对信噪比误差具有较强的鲁棒性，一种可行的方法是在信噪比上增加噪声的情况下训练 CNN-AMC。该方法可以降低分类器对信噪比偏差的敏感度，但以牺牲整体分类性能为代价。

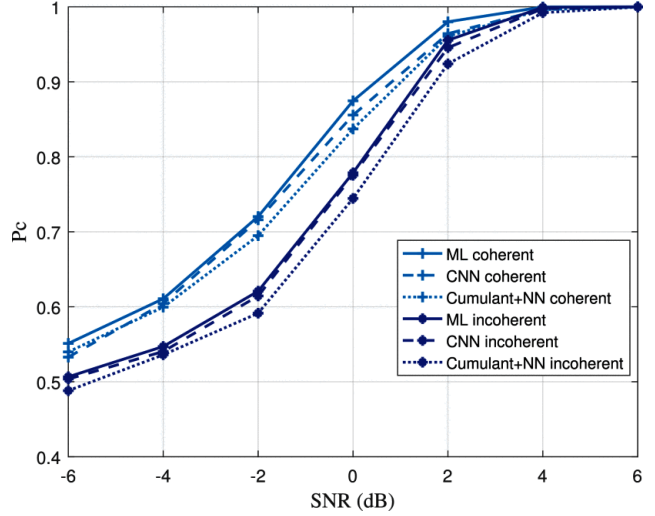


图 9. ML-AMC 和 CNN-500 的  $P_c$  曲线在  $[-6, 6]dB$  范围内随信噪比变化。一致性情况下的分类器用浅蓝色线条标绘，非一致性情况下的分类器用海军蓝线条表示。

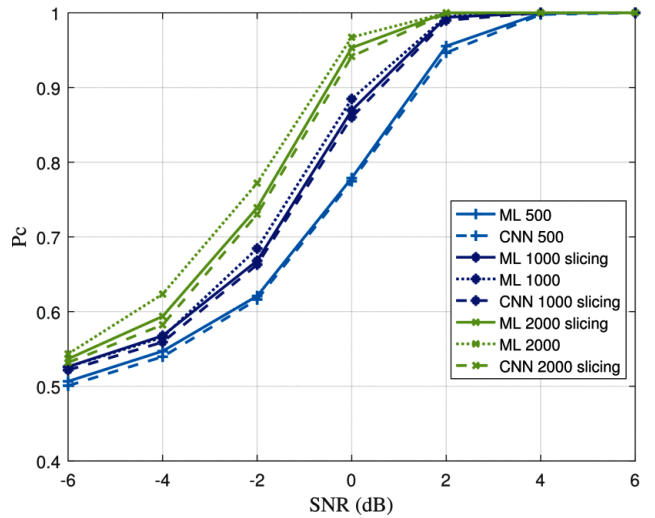


图 10. ML-AMC 和 CNN-500 在不同观测序列空间  $N$  下的  $P_c$  曲线与信噪比的关系。ML-AMC 用点状线绘制，使用信号段时用实线表示。CNN-AMC 用虚线描绘，并使用单元分类器。

## 4.2. 不相干

在非合作的情况下，相干环境通常是不可访问的，需要在接收端使用盲相位估计算法。在以往的仿真中，当相位估计误差  $|\Delta\theta_c|$  大于  $6^\circ$  时，CNN-AMC 的分类性能会受到很大的影响，因此最终的分类精度很大程度上取决于所采用的估计算法。在本小节中，假设  $\theta_c$  是在  $[0, 2\pi]$  范围内均匀分布的随机变量。

类似地，可以如下获得似然函数：

$$\begin{aligned} \Gamma(y | H_k) &\propto \mathbb{E}_{\theta_c} \left\{ \frac{1}{M_k} \prod_{n=0}^{N-1} \sum_{i=1}^{M_k} \exp \left( -\frac{|x_n^{k,i}|^2}{N_0} \right) \right. \\ &\quad \times \exp \left( \frac{2\Re \{ y_n^* x_n^{k,i} e^{j\theta_c} \}}{N_0} \right) \left. \right\} \\ &= \frac{1}{M_k} \int_0^{2\pi} \prod_{n=1}^N \sum_{i=1}^{M_k} \left[ \exp \left( -\frac{|x_n^{k,i}|^2}{N_0} \right) \right. \\ &\quad \times \exp \left( \frac{2\Re \{ y_n^* x_n^{k,i} e^{j\theta_c} \}}{N_0} \right) \left. \right] d\theta_c \end{aligned} \quad (26)$$

事实上，零阶第一贝塞尔函数可以用来简化 (26)，但由于其复杂性，在数值计算中不建议使用它。在替换中，通过将积分区间  $[0, 2\pi]$  分成相同长度的子区间，可以借助 Simpson 规则 [19] 充分逼近  $\theta_c$  上的期望值。利用复平面上星座点的对称性，可以将距离缩短到  $[0, \pi]$ 。在  $\theta_c$  上的附加积分使得计算总量是相干环境下的几倍，并且计算时间开销与子区间数成正比。

为了减少总计算量，调制集中仅包括 2PSK、4PSK、8PSK 和 16QAM。此外，只有 CNN-500 使用观测空间  $N = 500$  的非相干数据进行再训练。在预处理期间，不需要 (24) 中的相位补偿步骤。

- 相干 VS 非相干：ML-AMC、CNN-AMC 和 Cu-AMC 正确分类的总体概率如图 9 所示。可以看出，在非相干场景下，CNN-AMC 的近似非常接近最优分类，而且它的性能也优于 Cu-AMC。作为比较，三种分类器在相干场景下的总体正确概率用浅蓝色线条表示。在相干情况下，CNN-AMC 的  $P_c$  性能略好于 Cu-AMC，但在非相干情况下，CNN-AMC 的优势更加明显。要达到  $P_c = 90\%$ ，非相干分类器所需的信噪比可达 1.6dB，比相干分类器高约 1.0dB。一般来说，相干分类器和非相干分类器之间的性能差距随着信噪比的增加而逐渐缩小。当  $SNR=4dB$  时，两种分类器的正确判决概率均为 100%。非相干 CNN-500 也在观测空间  $N \in \{500, 1000, 2000\}$  上进行了测试，图 10 用虚线将 ML-AMC 视为最佳对比度。文本“切片”表示使用单位分类器

Predict Class	2PSK	9797 24.5%	48 0.1%	170 0.4%	187 0.5%	94.9%
	4PSK	175 0.4%	2324 5.8%	2108 5.3%	1929 4.8%	36.3%
	8PSK	70 0.2%	3385 8.5%	3565 8.9%	3094 7.7%	35.3%
	16QAM	85 0.2%	4116 10.3%	4157 10.4%	4790 12.0%	36.4%
	Precision	98.0%	23.2%	35.7%	47.9%	51.2%
ML-AMC						
Target Class						
ML-AMC						
Predict Class	2PSK	9633 24.1%	244 0.6%	232 0.6%	235 0.6%	93.1%
	4PSK	183 0.5%	3049 7.6%	2964 7.4%	2672 6.7%	34.4%
	8PSK	29 0.1%	2208 5.5%	2258 5.6%	1890 4.7%	35.4%
	16QAM	155 0.4%	4499 11.2%	4546 11.4%	5203 13.0%	36.1%
	Precision	96.3%	30.5%	22.6%	52.0%	50.4%
CNN-AMC						
Target Class						
CNN-AMC						

图 11. 在  $SNR = -6dB$  范围内，左侧 ML-AMC 和右侧 CNN-AMC 的可视化混淆矩阵。每个目标类用 10000 个样本进行测试，确切的数量和比例用蓝色和黄色的方块给出。在灰度列块、灰度行块和白块中，分别给出了查全率、查准率和准确率。

计算似然函数，使用分段的 ML-AMC 用实线表示。从理论上讲，在非相干环境下，观测信号的分块计算并不是最优的。从图中可以看出，当分段数  $N_s$  为 2 和 4 时，分别约有 0.2dB 和 0.4dB 的  $P_c$  损耗。另一方面，单元 CNN-500 的近似也非常接近采用不同输入间隔分段的 ML-AMC。当  $N \in \{1000, 2000\}$  时，在  $SNR = 2dB$  范围内，分类器的总体正确概率达到 100%。虽然单元分类器在性能上有一定损失，但在非相干情况下作为次优方法是可行的。此外，具有特定输入空间的新的 CNN-AMC 可以被训练成最优方法的近似值。

- 条件正确概率：通过对 ML-AMC 和 CNN-AMC 四种条件正确概率结果的初步分析，发现基于 CNN 的分类器在高信噪比区域与 ML-AMC 的近似程度相当接近，但在  $SNR < 2dB$  时不能很好地匹配。

为了进一步研究条件正确概率性能，在  $SNR = -6dB$  的范围内，图 11 显示了两个 AMC 的混淆矩阵。每行表示 AMC 的预测类中的实例，每列表示目标调制类中的实例。比较两个 AMC 的召回率和查准率，主要区别在于 4PSK 和 8PSK。4PSK 和 8PSK 都是相移键控，在复平面上邻接点之间的距离很小。因此，这两种调制方式很容易相互误分，特别是在低信噪

比的情况下。损失函数被设置为最大化总体正确概率，并且即使  $P_c$  在图9中接近近似，在这种情况下也不能保证条件正确概率的近似。

#### 4.3. 平坦衰落信道

研究了非频率选择性慢衰落信道。因此，假设幅度  $a$  和载波相位  $\theta_c$  在整个观测周期内为常数。载波相位的  $PDF$  与4.2中的相同。此外，假设振幅  $a$  的  $PDF$  服从下式给出的瑞利分布：

$$p(a) = \frac{2a}{\sigma_a^2} \exp\left(-\frac{a^2}{\sigma_a^2}\right), \quad a \geq 0 \quad (27)$$

其中  $\frac{2}{\sigma_a^2} = \mathbb{E}[a^2]$ 。除了预处理的观察序列  $y$  之外，CNN-AMC 仍然需要符号  $SNR$  的知识。似然函数表示为：

$$\begin{aligned} \Gamma(y | H_k, a) &\propto \mathbb{E}_{\theta_c} \left\{ \frac{1}{M_k} \prod_{n=1}^N \sum_{i=1}^{M_k} \exp\left(-\frac{|ax^{k,i}|^2}{N_0}\right) \right. \\ &\quad \times \exp\left(\frac{2\Re\{y_n^* ax^{k,i} e^{j\theta_c}\}}{N_0}\right) \Big\} \\ &= \frac{1}{M_k} \int_0^\pi \prod_{n=1}^N \sum_{i=1}^{M_k} \left[ \exp\left(-\frac{|ax^{k,i}|^2}{N_0}\right) \right. \\ &\quad \times \exp\left(\frac{2\Re\{y_n^* ax^{k,i} e^{j\theta_c}\}}{N_0}\right) \Big] d\theta_c \end{aligned} \quad (28)$$

仿真中考虑了单输入多输出 (SIMO) 的情况，其中一台发射机和一台接收机具有  $N_a$  个天线。假设这些通道彼此独立。具有多个观测值的似然函数给出如下：

$$\Gamma(\{y_1, \dots, y_{N_a}\} | H_k, \{a_1, \dots, a_{N_a}\}) = \prod_{u=1}^{N_a} \Gamma(y_u | H_k, a_u) \quad (29)$$

这忽略了发送的码元序列对于所有天线都相同的事实。形式上，一个信号源的多个观测过程类似于 CNN-AMC(16) 中的信号段。

在我们的仿真中，我们研究了使用 1, 2, 4 天线的接收机，从而测试了整体分类性能与平均符号信噪比 (用信噪比表示) 的性能。如图12所示，CNN-500 的曲线如图所示，ML-AMC 被视为它的对应物。可以看出，在平坦 Rayleigh 衰落信道中，当  $SNR = 4dB$

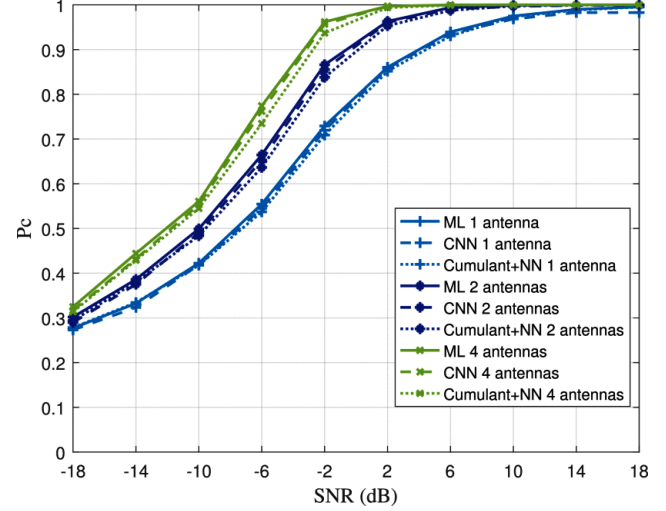


图 12.  $P_c$  与  $SNR$  在范围  $[-18, 18]$  dB 内的关系。每个天线的观测空间  $N = 500$ ，1, 2, 4 个天线的曲线分别用浅蓝色，深蓝色和绿色线表示

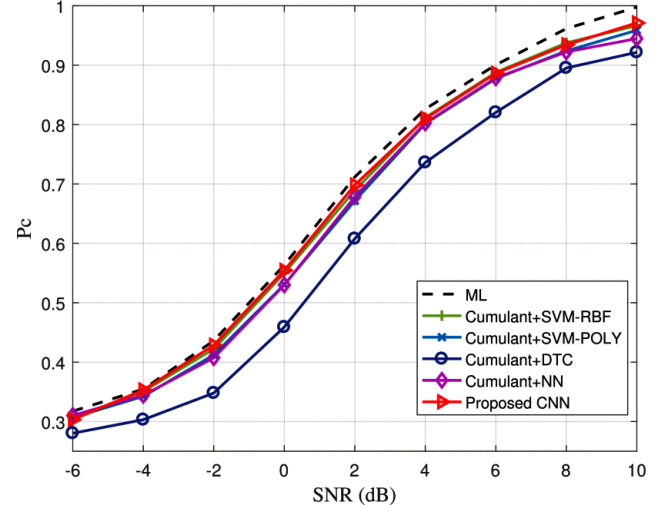


图 13. 在相干环境中 ( $N = 500$ ) 对各种 AMC 进行基线测试。  $P_c$  对  $SNR$  的曲线图。

时，单天线接收机的总体正确概率  $P_c = 90\%$ 。相比之下，具有 2 个和 4 个天线的接收机所需的信噪比分别约为  $-1dB$  和  $-3dB$ 。当  $SNR = 2dB$  时，在 4 个天线的情况下几乎没有分类错误。结合图6和图9的性能，CNN-AMC 的性能总体上可以达到更接近于最优的结果，相较 Cu-AMC 而言。

#### 4.4. 基于特征的方法

除了前面提到的基于神经网络的方法之外，本小节还考虑了其他一些分类器，包括决策树分类器



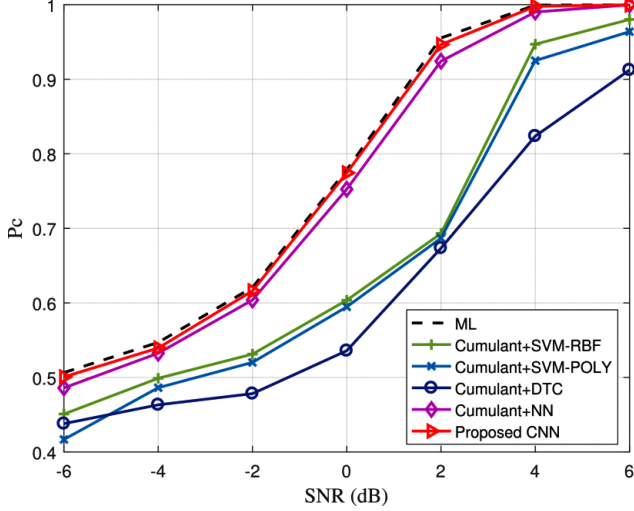


图 14. 在非相干环境中 ( $N = 500$ ) 对各种 AMC 进行基线测试。  $P_c$  对 SNR 的曲线图。

(DTC)、多项式支持向量机 (SVM) 和径向基函数 (RBF) 核的支持向量机 (SVM-Poly) 和支持向量机-RBF (SVM-RBF)。将 4.1 中的数据集 ( $N = 500$ ) 作为相干环境中的基线测试。如图 13 所示, CNN-AMC 和 SVM-RBF 的总体  $P_c$  非常相似, 而基于神经网络的 AMC 与这两者相比略有  $P_c$  性能损失。DTC 分类器的分类正确率最低。

然后用 4.2 中的非相干基线数据集 ( $N = 500$ ) 测试这些 AMC。结合图 13, DTC 分类器的正确分类性能仍然是最差, 而基于支持向量机的方法在图 14 中并不稳定。在基于特征的方法中, 只有基于神经网络的 AMC 在相干和非相干环境下都能获得与 CNN-AMC 最接近的  $P_c$  性能, 这就是为什么在以往的仿真中它被用作 Cu-AMC 的原因。此外, CNN-AMC 的  $P_c$  性能略好于基于神经网络的 AMC。

表 3. 相干环境下分类器的平均推理时间开销

$N$	500	1000	2000	8000
ML-AMC/ms	10.8	21.3	42.6	164.5
CNN-500/ms	0.0727	0.146	0.288	1.15
CNN-1000/ms	—	0.127	0.246	1.02

#### 4.5. 复杂性 & 实施

在这一节中, 我们讨论了 ML-AMC 和 CNN-AMC 的复杂性分析和实现。MLAMC 不需要内存,

而且 CNN500 和 CNN-1000 的规模很小, 在  $N=500$  和  $N=1000$  的情况下分别只占用 3.4MB 和 6.3MB 的容量。仿真平台为: CPU intel i7-6700k 和 GPU Nvidia GTX-960。ML-AMC 是用 Matlab 实现的, 主要利用了 CPU。而 CNN-AMC 则运行在搭载 GPU 的 Python 平台上。

- 相干: 表 3 列出了 4.1 中三个分类器在不同输入空间下的每次推理的时间成本。单位时间成本定义为每个推理的时间成本除以  $N$ , ML-AMC 的单位时间成本为  $20.8\mu s$ , CNN-500 的单位时间成本为  $0.144\mu s$ , CNN-1000 的单位时间成本为  $0.127\mu s$ 。在这种情况下, 基于神经网络的方法使计算速度提高了两个数量级以上。更具体地说, CNN-500 和 CNN-1000 分别比 ML-AMC 快 150 倍和 170 倍。
- 非相干: 在非相干情况下, ML-AMC 需要额外的载波相位积分。理论上, 单位时间开销约为相干场景下的  $N_c$  倍, 其中  $N_c$  为子区间数。在 4.2 中进行仿真测试后, 在相干和非相干场景下, 表 5 中列出的单位时间成本分别为  $6.01\mu s$  和  $268\mu s$ 。后者约为相干变量的 44.7 倍, 近似等于所采用的变量值  $N_c = 45$ 。

另一方面, CNN 使能方法的单位时间成本只与模型的体系结构有关。除最上层外, 7 种调制方式和 4 种调制方式的 CNN-500 的结构相同, 因此表 4 中的 CNN-500 的平均时间开销与表 3 中的相应方案非常相似, 而且基于 CNN 的方法的平均时间开销与通信环境无关, 其单位时间开销为  $0.155\mu s$ , 因此在此仿真情况下, 在相干和非相干环境下, CNN-500 分别比 ML-AMC 快约 40 倍和 1,700 倍。

对于 ML-AMC 和 CNN-AMC, 计算量都随着观测空间  $N$  的增加而线性增长。本文提出的 CNN-AMC 算法的计算复杂度主要由网络结构决定, 调制方式和信道环境对总计算量的影响可以忽略不计。相比之下, ML-AMC 受到这些问题的影响很大。表 5 列出了在我们的三个模拟环境中两个 AMC 的计算复杂度。符号  $k_{m1}$  和  $k_{cnn}$  是相应 AMC 的系数。根据表 5, 直观地

表 4. 四种调制方案分类的平均时间开销

	$N$	500	1000	2000
Coherent				
	ML – AMC/ms	3.08	6.07	11.8
	CNN – 500/ms	0.0775	0.155	0.309
Incoherent	$N$	500	1000	2000
	ML – AMC/ms	138	271	528
	CNN – 500/ms	0.0775	0.155	0.309

表 5. 四种调制方案分类的平均时间开销

	ML-AMC	CNN-AMC
Coherent	$k_{\text{ml}}O\left(N\sum_{k=1}^C M_k\right)$	$k_{\text{cnn}}O(N)$
Incoherent	$k_{\text{ml}}O\left(NN_{\theta_c}\sum_{k=1}^C M_k\right)$	$k_{\text{cnn}}O(N)$
Flat fading channel	$k_{\text{ml}}O\left(NN_{\theta_c}N_a\sum_{k=1}^C M_k\right)$	$k_{\text{cnn}}O(NN_a)$

看出 CNN-AMC 算法的计算复杂度低于 ML-AMC 算法。实际上, CNN-AMC 的  $k_{\text{cnn}}$  的系数比 ML-AMC 的系数大得多, ML-AMC 的总计算量在相干场景下可以更小。但是, 由于并行处理, CNN-AMC 的每个推理在实现时的时间开销可以大大降低。可以并行实现一层不同核或神经元的输出, 也可以同时处理不同接收序列的分类。另一方面, ML-AMC 的计算复杂度容易受到信道环境和调制集的影响。相反, 所提出的 CNN-AMC 的每次推理的时间开销是不变的, 作为一种具有强大和冗余逼近能力的深度学习模型。这表明 CNN-AMC 的性能优于 ML-AMC, 特别是在更复杂的环境中。

## 5. 结论

本文提出了一种基于深度学习方法的自动调制分类方法。作为一种并行计算模型, 提出了一种基于深度学习的分类器 CNN-AMC, 作为 ML-AMC 的近似。CNN-AMC 自动从长符号率序列中提取特征, 还需要 SNR 作为输入。引入了相应的两步训练过程和转移学习, 分别解决了具有挑战性的训练任务, 提高了再培训效率。此外, 为了灵活地适应不同的观测序列维数, 还提出了一种单元分类器。仿真结果表明, 基于 CNN 的分类器性能优于基于特征的分类器, 且最接近最优 ML-AMC。此外, CNN-AMC 对

载波相位偏差和信噪比估计误差具有一定的鲁棒性。通过利用并行计算, 基于深度学习的方法在平均分类速度方面比 ML-AMC 快约 40 到 1700 倍, 具体取决于给定的场景。在未来的工作中, 我们将重点研究基于 CNN 的分类器对不同偏差的健壮性, 以及在复杂的非合作场景中的应用。

**说明.** 本文为发表在 IEEE TVT 的 Automatic Modulation Classification: A Deep Learning Enabled Approach 的中文翻译版, 原作者为 Fan Meng, Peng Chen, Lenan Wu, Xiaobin Wang。合肥工业大学计算机与信息学院惠英哲为本项工作完成者, 作为其学士毕业设计的外文翻译工作。

## 参考文献

- [1] A. Nandi and E. Azzouz, “Algorithms for automatic modulation recognition of communication signals,” IEEE Transactions on Communications, vol. 46, no. 4, pp. 431–436, 1998. 1
- [2] O. A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, “Survey of automatic modulation classification techniques: classical approaches and new trends,” IET communications, vol. 1, no. 2, pp. 137–156, 2007. 1
- [3] C. M. Bishop and N. M. Nasrabadi, Pattern recognition and machine learning, vol. 4. Springer, 2006. 1

- [4] M. Bkassiny, Y. Li, and S. K. Jayaweera, "A survey on machine-learning techniques in cognitive radios," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1136–1159, 2012. 1
- [5] A. Hazza, M. Shoaib, S. A. Alshebeili, and A. Fakhad, "An overview of feature-based methods for digital modulation classification," in *2013 1st international conference on communications, signal processing, and their applications (ICCSPA)*, pp. 1–6, IEEE, 2013. 1
- [6] S. U. Pawar and J. F. Doherty, "Modulation recognition in continuous phase modulation using approximate entropy," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 843–852, 2011. 1
- [7] Y.-c. Deng and Z.-g. Wang, "Modulation recognition of mapsk signals using template matching," *Electronics Letters*, vol. 50, no. 25, pp. 1986–1988, 2014. 1
- [8] L. Xie and Q. Wan, "Cyclic feature-based modulation recognition using compressive sensing," *IEEE Wireless Communications Letters*, vol. 6, no. 3, pp. 402–405, 2017. 1
- [9] A. Swami and B. M. Sadler, "Hierarchical digital modulation classification using cumulants," *IEEE Transactions on communications*, vol. 48, no. 3, pp. 416–429, 2000. 1
- [10] A. Smith, M. Evans, and J. Downey, "Modulation classification of satellite communication signals using cumulants and neural networks," in *2017 Cognitive Communications for Aerospace Applications Workshop (CCAA)*, pp. 1–8, IEEE, 2017. 1
- [11] C.-S. Park, J.-H. Choi, S.-P. Nah, W. Jang, and D. Y. Kim, "Automatic modulation recognition of digital signals using wavelet features and svm," in *2008 10th International Conference on Advanced Communication Technology*, vol. 1, pp. 387–390, IEEE, 2008. 1
- [12] D. Boutte and B. Santhanam, "A hybrid ica-svm approach to continuous phase modulation recognition," *IEEE Signal Processing Letters*, vol. 16, no. 5, pp. 402–405, 2009. 1
- [13] Z. Zhang, Y. Li, X. Zhu, and Y. Lin, "A method for modulation recognition based on entropy features and random forest," in *2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pp. 243–246, IEEE, 2017. 1
- [14] M. D. Wong and A. K. Nandi, "Automatic digital modulation recognition using artificial neural network and genetic algorithm," *Signal Processing*, vol. 84, no. 2, pp. 351–365, 2004. 1
- [15] E. E. Azzouz and A. K. Nandi, "Modulation recognition using artificial neural networks," in *Automatic Modulation Recognition of Communication Signals*, pp. 132–176, Springer, 1996. 1
- [16] D. Wang, M. Zhang, Z. Cai, Y. Cui, Z. Li, H. Han, M. Fu, and B. Luo, "Combating nonlinear phase noise in coherent optical systems with an optimized decision processor based on machine learning," *Optics Communications*, vol. 369, pp. 199–208, 2016. 1
- [17] X. Zhang, J. Chen, and Z. Sun, "Modulation recognition of communication signals based on schks-svm," *Journal of Systems Engineering and Electronics*, vol. 28, no. 4, pp. 627–633, 2017. 1
- [18] J. L. Xu, W. Su, and M. Zhou, "Likelihood-ratio approaches to automatic modulation classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 4, pp. 455–469, 2010. 1
- [19] J. L. Xu, W. Su, and M. Zhou, "Software-defined radio equipped with rapid modulation recognition," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 4, pp. 1659–1667, 2010. 1, 11
- [20] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015. 1
- [21] Y.-H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyerriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE journal of solid-state circuits*, vol. 52, no. 1, pp. 127–138, 2016. 1
- [22] Y. He, F. R. Yu, N. Zhao, V. C. Leung, and H. Yin, "Software-defined networks with mobile edge computing and caching for smart cities: A big data deep reinforcement learning approach," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 31–37, 2017. 2
- [23] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Machine learning for wireless networks with artificial intelligence: A tutorial on neural networks," *arXiv preprint arXiv:1710.02913*, vol. 9, 2017. 2

- [24] D. Wang, M. Zhang, Z. Li, J. Li, M. Fu, Y. Cui, and X. Chen, "Modulation format recognition and osnr estimation using cnn-based deep learning," *IEEE Photonics Technology Letters*, vol. 29, no. 19, pp. 1667–1670, 2017. 2
- [25] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. 2
- [26] T. J. O' Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *International conference on engineering applications of neural networks*, pp. 213–226, Springer, 2016. 2
- [27] T. O' shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017. 2
- [28] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, and S. Pollin, "Deep learning models for wireless signal classification with distributed low-cost spectrum sensors," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 3, pp. 433–445, 2018. 2
- [29] A. Ali, F. Yangyu, and S. Liu, "Automatic modulation classification of digital modulation signals with stacked autoencoders," *Digital Signal Processing*, vol. 71, pp. 108–116, 2017. 2
- [30] K. L. Priddy and P. E. Keller, *Artificial neural networks: an introduction*, vol. 68. SPIE press, 2005. 4
- [31] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016. 4
- [32] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 315–323, JMLR Workshop and Conference Proceedings, 2011. 5
- [33] P. Koniusz, F. Yan, and K. Mikolajczyk, "Comparison of mid-level feature coding approaches and pooling strategies in visual concept detection," *Computer vision and image understanding*, vol. 117, no. 5, pp. 479–492, 2013. 5
- [34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017. 5
- [35] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009. 7
- [36] M. W. Aslam, Z. Zhu, and A. K. Nandi, "Automatic modulation classification using combination of genetic programming and knn," *IEEE Transactions on wireless communications*, vol. 11, no. 8, pp. 2742–2750, 2012. 8