

Application and Implementation of Deep Learning in Wireless Transmission Physical Layer

Junhui Gao, Jiake Li

Southeast University, Nanjing Jiangsu
Email: 543825106@qq.com

Received: Jan. 13th, 2020; accepted: Jan. 27th, 2020; published: Feb. 3rd, 2020

Abstract

The explosive emergence of new communication scenario and the rapid growth of terminal access equipment have made channel modeling difficult, and traditional communication algorithms have difficulty meeting the requirements for real-time and accurate signal processing. Deep learning has the characteristics of strong model learning ability, simple structure and high operation speed, so it has become the mainstream direction of wireless physical layer transmission research. This paper first introduces three classic neural networks based on deep learning, and then summarizes and explains the application results of deep learning in wireless transmission physical layer modules such as frame synchronization, encoder, detector, and end-to-end replacement of the entire receiver.

Keywords

Deep Learning, Frame Synchronization, Channel Coding, Signal Detection, Receiver

深度学习在无线传输物理层的应用与实现

高君慧, 李嘉珂

东南大学, 江苏 南京
Email: 543825106@qq.com

收稿日期: 2020年1月13日; 录用日期: 2020年1月27日; 发布日期: 2020年2月3日

摘要

新型通信业务大量涌现和终端接入设备的急剧增长, 导致信道建模复杂度显著提升, 传统通信算法难以满足实时且精确地进行信号处理的要求, 而深度学习(Deep Learning, DL)技术凭借其强大的模型学习能

力、结构简单且运算速度较高的特点, 成为无线物理层传输研究的主流方向。本文首先介绍了基于DL技术的三种经典神经网络, 随后对DL技术在无线传输物理层模块如帧同步、编码器、检测器以及对整个接收机端到端替代的应用成果进行了总结和说明。

关键词

深度学习, 帧同步, 信道编码, 信号检测, 接收机

Copyright © 2020 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

人类进入 21 世纪以来, 互联网产业和移动通信技术飞速发展。随着移动互联网和物联网时代终端设备和通信业务量的爆炸性增长, 新型通信场景的不断涌现, 传统通信算法中信道建模困难、高效实时信号处理难以实现以及模块化信号处理影响整体接收机性能的缺点逐渐显现。另一方面, 深度学习(Deep Learning, DL)技术在计算机视觉, 自动语音识别和自然语言处理等领域上取得巨大成功, 引起了业界的广泛关注。而深度网络被证明只要足够宽, 可以进行任何函数的模拟, 其显著的算法学习能力可以通过简便的训练方法获得复杂的信道场景, 而不需建立于定义明确的数学模型之上, 或者像传统算法那样严格基于信息理论等, 极大简化了建立算法的过程; 同时, DL 算法往往采用分布式并行架构, 可同时保证计算速度和处理容量, 满足信号实时处理的需要; 最后, 基于 DL 技术的无线传输系统可以训练优化端到端的特性来打破传统通信系统中人为定义的模块化结构, 获得全局性能的改善。基于以上优点, 将无线通信领域的问题尝试用 DL 技术进行解决成为了一个新的研究方向。

本章首先介绍了基于 DL 技术的三种经典神经网络以及神经网络优化的基本方法, 之后详细的介绍了 DL 技术对无线传输物理层模块如帧同步、编码器、检测器的应用与优化, 以及神经网络对于整个无线通信接收机端到端替代的运用成果。

本文中, 所用符号的相关说明见表 1。

Table 1. Symbol description

表 1. 符号说明

符号	符号说明
$(\cdot)^H$	矩阵或矢量的共轭转置
$(\cdot)^T$	矩阵或矢量的转置
$(\cdot)^{-1}$	矩阵的逆矩阵

2. 深度学习基础

1996 年, Langley 将机器学习(Machine Learning, ML)纳入人工智能领域, 其主要目的在于通过促使计算机进行自主学习, 不断进行经验的获取来改善自身性能, 使得计算机具备智能特性, 从而可以自主解决问题。20 世纪以后, ML 获得了很大的发展, 研究出了一系相关的算法, 主要包括决策树(Decision Tree,

DT)、向量机(Vector machine, VM)以及神经网络(Neural Network)等。而 DL 技术就发展自神经网络, 它是一种监督学习的方法, 其基本原理为在网络训练阶段通过后向传播(Backward propagation, BP)调整神经网络参数的权重值, 并不断地迭代来缩小代表预测值和真实值之间差距的损失函数, 在测试阶段将最优的神经网络参数应用于正向传播中得到最后的预测值。下面介绍 DL 技术中三种经典的神经网络结构, 全连接神经网络、卷积神经网络(Convolutional Neural Network, CNN)和循环神经网络(Recurrent Neural Network, RNN)。

2.1. 全连接网络

全连接神经网络的想法起源于神经元模型, 如图 1 所示, 在一个神经元中, 神经元的输出可以表示为:

$$y = \sigma \left(\sum_{k=1}^r w_k x_k + b \right) \quad (1)$$

其中, x_k 表示输入, w_k 表示对应的权重, b 为偏执, $\sigma(\cdot)$ 为激活函数, y 为输出。激活函数为神经元模型引入了非线性因素, 常见的激活函数为 Sigmoid 函数, 其表达式为:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2)$$

将上述多个神经元连接起来, 变成层数为 1 的层次化结构, 即成为最基本的全连接神经网络, 该网络由输入层和输出层组成, 结构简单。网络训练时, 输入层进行样本输入后, 即可完成一个由输入到输出的非线性映射, 随后计算输出值和真实值(也称作标签)之间的代价函数, 通过梯度下降法(Gradient Descent, GD)不断优化网络的权重参数来降低代价函数的值, 使得该网络可以更好的模拟输入和输出之间的映射函数。常用的代价函数包括回归问题中的均方误差函数和分类问题中的交叉熵函数等。

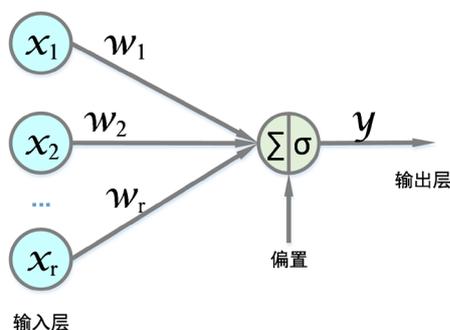


Figure 1. Neuron model
图 1. 神经元模型

由于激活函数给神经网络赋予了非线性特性。当将网络层数加深且每层神经元的个数足够时, 那么该神经网络理论上讲可以模拟任意函数特征, 我们称此神经网络为全连接深度神经网络(Fully connected deep neural network, FC-DNN), 该网络由输入层、隐藏层和输出层三部分组成, 具体结构如图 2 所示。在此种网络结构中, 每一层的神经元都与之相邻层的神经元相连, 与不相邻的层互不连接。当隐藏层的层数逐渐增多时, 采用梯度下降法进行网络参数的优化将会变得很困难, 而 BP 算法却可以很好的解决这个问题, 大大提高了 FC-DNN 网络的可用性。但是随着网络深度和神经元个数的增加, 网络训练过程中还是会出现梯度消失或梯度爆炸、网络收敛速度太慢以及网络过拟合的问题。

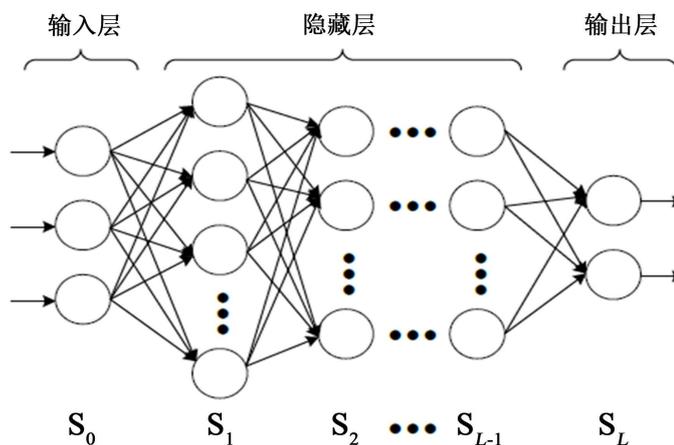


Figure 2. Fully connected deep neural network model
图 2. 全连接深度神经网络模型

由于目前神经网络的优化方法都是基于 BP 算法的思想, 即根据损失函数计算的误差通过梯度反向传播的方式, 指导深度神经网络参数权值的更新优化。而深度神经网络由许多非线性层堆叠而来, 每一层非线性层都可以视为是一个非线性函数, 因此整个深度网络可以视为是一个复合的非线性多元函数。所以说, 若对激活函数进行求导, 如果此部分大于 1, 那么层数增多的时候, 最终的求出的梯度更新将以指数形式增加, 即发生梯度爆炸, 如果此部分小于 1, 那么随着层数增多, 求出的梯度更新信息将会以指数形式衰减, 即发生了梯度消失。因此, 很多时候的梯度消失或者梯度爆炸可以通过选择更为合适的激活函数来解决。例如 Relu 激活函数, 其表达式为:

$$\text{Relu}(x) = \max(x, 0) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (3)$$

Relu 函数在大于 0 的区间内导数一直为 1, 每层网络都是相同的更新速度, 这样自然就不会出现梯度消失或梯度爆炸的问题了。

为了加快网络收敛速度, 首先可以采用动态学习速率, 即在初始时刻学习率设置较大之后随着训练轮数的增加逐渐降低, 如目前的 Adagrad, Momentum 和 Adam 算法就可以进行动态学习率的调整。还可以采用小批次梯度下降算法, 即每轮放弃对全部样本进行训练, 而是每次只选择小批次样本训练, 但是此方法有可能会陷入局部最优解。另外选择合适的网络初始化参数也可以有效提升收敛速度。当神经网络在训练集得到很好的性能后可以进入测试阶段, 此时可能会出现在测试集性能较差的情况, 这种现象称为过拟合。一般来说, 训练误差会随着训练次数的增多逐渐下降, 而测试误差则会先下降而后再次上升。为了避免过拟合训练集, 一个很好的解决方案是提前停止(Early stopping), 当它在验证集上的性能开始下降时就中断训练; 过拟合也可以理解为当一个模型过为复杂的时候, 它可以很好地记忆每一个训练数据中随机噪音的部分而忘记了要去学习训练数据中通用的趋势, 因此可以在损失函数中加入刻画模型复杂程度的指标, 这种方法称为正则化。其基本思想均为通过限制权重的大小, 使得模型不能任意拟合训练数据中的随机噪音。最后一种常用的防止过拟合的方法为随机失活(Dropout), 主要是指在神经网络的学习过程中, 随机将部分隐含层节点的权重归零, 由于每次迭代受归零影响的节点不同, 因此各节点的重要性会被平衡, 从而降低过拟合的风险。

2.2. 卷积神经网络

CNN 也是一种常用的神经网络, 主要是为了进行图像数据处理而提出, 此网络可以自主提取特征并

进行图像恢复, 与此同时相较于 DNN 网络拥有更少的训练参数, 由于 CNN 可以根据实际需要而自主设计网络结构, 这为其应用于通信物理层传输提供了可能性。该网络的基本模型结构如图 3 所示, 包含了卷积层、池化层和全连接层。

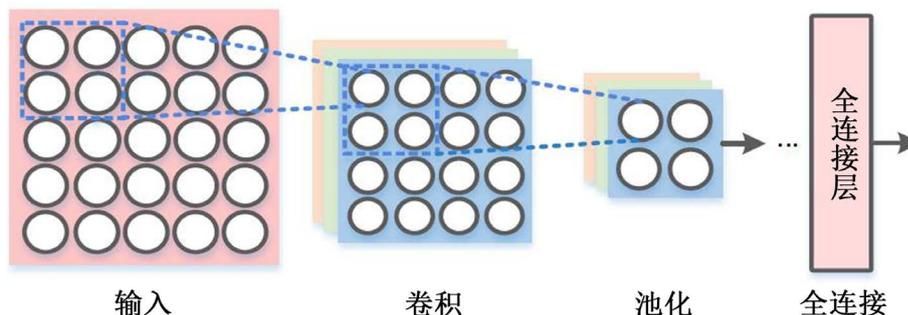


Figure 3. Convolutional neural network model

图 3. 卷积神经网络模型

由于 CNN 最初用于图像处理, 图像一般是由多通道的二维矩阵组成, 因此卷积层的输入一般为多通道的二维矩阵, 卷积层设有多个卷积核, 其中卷积核的通道数与输入层通道数相同, 每个卷积核根据步长数对输入进行扫描完成卷积运算, 通过减小代价函数和 BP 不断优化卷积核参数的值, 得到多张特征图, 特征图的通道数与卷积核个数相同。多层卷积层相连的神经网络称为深度 CNN, 在此结构中, 为了保持图像分辨率不变, 需要在每层的输入矩阵进行补零(Padding)操作。由于卷积层有一个明显的特征为权重共享, 因此与 FC-DNN 相比极大减少了训练参数。其实卷积的过程, 就是一个从细节到抽象的过程, 我们输入的图像就是一些纹理, 卷积核也可以看成纹理, 卷积的目的就是时卷积核的纹理和图像的纹理尽可能的相似, 在二维空间中, 纹理等价于向量, 卷积操作为向量的相乘, 相乘结果越大, 说明两个向量更为接近, 因此, 卷积后的新图像在具有卷积核纹理的区域信号更强, 其他区域较弱, 如此完成了从图像中抽取特征的操作, 卷积层的输出可通过激活函数来引入非线性因素。

卷积层后一般加入池化层, 减少特征图的维度, 降低数据运算量。池化运算, 一般有两种 MaxPooling 和 MeanPooling。选取一定大小池化窗口, 然后从左往右进行扫描, 选取池化窗口中最大值作为该位置的特征值的方法是 MaxPooling, 选取池化窗口中平均值作为该位置的特征值的方法是 MeanPooling, 而池化层的操作并不会损失重要的图像特征, 是因为图像数据在连续区域具有相关性, 一般局部区域的像素值差别不大。输出最后经过全连接层, 对特征进行重新拟合, 减少特征信息的丢失。由于通过卷积池化操作后得到的是多个特征矩阵, 而全连接层的输入为向量, 所以在进行全连接层之前, 要将多个特征矩阵压平为一个向量。需要注意的是, 无论是卷积层还是池化层, 均完整的保留了图像特征间的空间关系。比较经典的 CNN 网络有 LeNet, GoogLeNet, ResNet 等。

2.3. 循环神经网络

无论是 DNN 还是 CNN, 两者有一个共同的特点, 每一层神经元之间是相互独立的, 然而, 许多现实问题中许多元素是相关关联的, 而这种关联性较强的时序数据的处理和预测, 则需要 RNN 网络来完成。RNN 网络具有很好的记忆性, 即此时此刻网络的输出与上一时刻网络的输出也有关系, 具体实施为将上一时刻的部分输出反馈至这一时刻, 作为这一时刻的部分输入。深层 RNN 是在 RNN 模型中多添加了几个隐藏层, 这是因为考虑到当信息量过大时一次性无法保存所以重要信息, 通过多个隐藏层可以保存更多的重要信息, 深层 RNN 具体结构如图 4 所示。

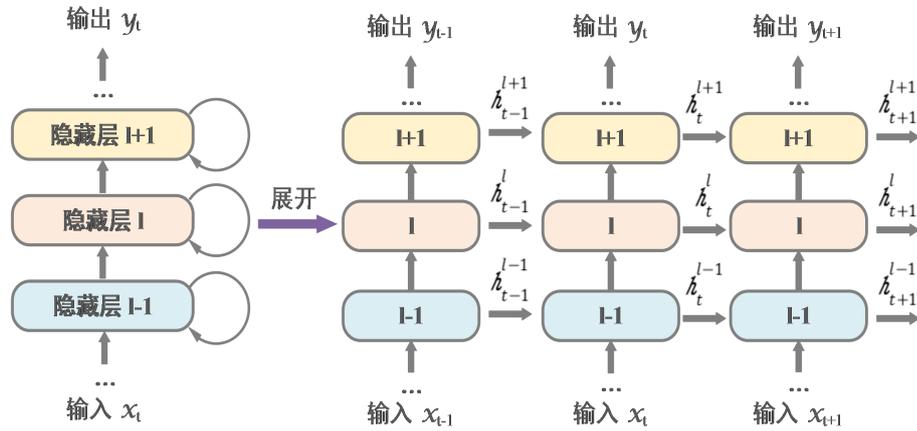


Figure 4. Recurrent neural network model
图 4. 循环神经网络模型

RNN 网络同样遵从权重共享的特性,但是在训练过程中会存在长期依赖的问题,简单来说,长期依赖的问题就是指在每一个时间间隔不断增大时,RNN 将会失去连接到远处信息的能力,这主要是由于梯度消失或者梯度爆炸引起的。而 LSTM 网络可以通过刻意的设计避免长期依赖的问题,使记住长期的信息成为 LSTM 的默认行为,具体的 LSTM 单元结构如图 5 所示,RNN 网络包含细胞状态、遗忘门、输入门和输出门四部分。细胞状态用来保存之前的重要信息;遗忘门控制遗忘上一层细胞状态的内容,通过 Sigmoid 激活函数,得到上一层细胞状态内容决定哪些去除和哪些留下,具体函数表达式为:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (4)$$

其中 $\sigma(\cdot)$ 表示 Sigmoid 激活函数, W_f 和 b_f 分别表示权重和偏置。输入门则处理当前序列位置的输入,确定需要更新的信息,去更新细胞状态。此过程分为两部分,一部分是通过 Sigmoid 激活函数决定哪些新的信息加入到细胞状态;之后要将新信息转换成可以加入到细胞状态的正确形式,即使用 tanh 函数产生新的候选向量,该控制门的计算表达式和新添加的细胞状态可以表示为:

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \end{aligned} \quad (5)$$

其中,激活函数 tanh 的表达式为:

$$\tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \quad (6)$$

经过遗忘门和输入门之后,就可以把细胞状态 C_{t-1} 更新为 C_t 了,具体表达式如下所示:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (7)$$

其中, $f_t * C_{t-1}$ 为希望遗忘的信息, $i_t * \tilde{C}_t$ 为新添加的信息。最后输出门要根据细胞状态所包含的内容决定输出。具体是指输出门需要使用 Sigmoid 激活函数决定哪些内容需要输出,之后使用 tanh 激活函数对细胞状态包含的内容进行处理,两部分相乘得到最后的输出,计算表达式为:

$$\begin{aligned} o_t &= \sigma(W_o [h_{t-1}, x_t] + b_o) \\ h_t &= o_t * \tanh(C_t) \end{aligned} \quad (8)$$

通过细胞状态的作用,序列信号在前面时刻获得的信息可以一直进行保存,并会对后续的 LSTM 单元产生影响,此网络结构解决了长期依赖的问题,成为目前解决时序数据处理问题的最常用网络。除此

以外, 有时候序列的预测值可能由前面时刻和后面时刻的若干输入共同决定, 因此也产生了双向 RNN、双向 LSTM 等网络结构。

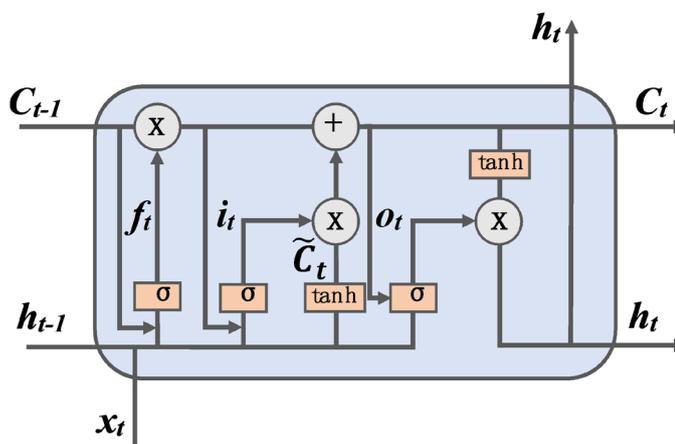


Figure 5. LSTM unit structure

图 5. LSTM 单元结构

3. 深度学习在无线传输的应用

随着 DL 技术在计算机视觉、自然语言处理等领域的成功应用, 将 DL 技术应用于无线传输的物理层以适应越来越复杂的通行场景提成为无线通信领域的热门研究方向。早期的研究人们尝试与将 DL 技术应用在物理层的单个模块中, 如帧同步模块、信道译码和信号检测等, 以期获得优于传统通信算法的性能。虽然此方法设计得到的模块是可控且通用的, 但是无法保证可以达到整个系统的全局最优, 所以近年来的研究更加倾向于将神经网络对整个通信系统进行端到端的整体替代, 因此本节也主要围绕以上两个方面进行详细的介绍。

3.1. 基于 DL 的帧同步模块

在无线链路传输中, 传输数据包本应在预定的时间发送, 并且接收机应该根据发送端的数据发送时间得知每个分组数据包的到达时间, 但是, 由于无线电信号和时钟的传播延迟的偏移, 数据包到达的确切时间在接收端并不能确定。因此接收端通常使用传输数据包前面的前导序列判断一帧数据的开头, 此操作称为帧同步。传统的帧同步方法是发送端选择一个自相关特性良好的序列作为同步序列, 接收端将接收到的信号与本地同步序列进行滑动互相关, 当互相关的值最大时的接收信号起始点作为同步头。此方法在 SNR 值较大时性能良好, 但是当 SNR 的值很低时, 由于随机噪声的影响经过无线信道传输后的同步序列不再有很好的自相关特性, 进而降低了帧同步的准确性。文献[1]提出了一种基于 CNN 架构的帧同步算法, 具体结构图 6 所示, 该算法将一个由伪随机噪声作为同步序列的发送信号通过无线信道, 经过脉冲成形和过采样后的数值与本地同步序列进行互相关, 其输出的一维向量转换成一个二维矩阵, 将二维矩阵作为输入给一个三层的二维 CNN, 每层 CNN 通过设置多个卷积核学出该二维图像的多个图像特征, 增加激活函数来引入非线性因素, 引入 Dropout 层防止过拟合, 最后经过网络训练分类器的输出即为同步头的索引。仿真结果表明在低信噪比下, 此算法的性能相较于传统同步算法具有明显的提升。这是因为 SNR 的值较低时, 由于过采样效应的影响, 相关器的输出值在同步头附近多次采样的平均值接近正态分布, 传统基于相关性的帧同步方法忽略了其输出的分布形状而仅去寻找最大值, 但是, 基于 CNN 的帧同步方法根据图像处理的原则不仅考虑了最大值, 而且考虑了相关值的形状, 从而更准确的找到了同步头, 这就是 CNN 分类器的优点。



Figure 6. Frame synchronization module based on CNN architecture

图 6. 基于 CNN 架构的帧同步模块

3.2. 基于 DL 的信道译码模块

移动通信中存在干扰和衰落, 在信号传输过程中可能出现差错, 故对数字信号必须采用纠、检错技术, 以增强数据在信道中传输时抵御各种干扰的能力, 提高系统的可靠性。因此发送端要对发送信号进行纠、检错编码, 这就是信道编码, 相应地在接收端需要进行信道译码。将 ML 应用于信道译码技术的研究在 1990 年就开始进行, DL 技术在此领域的研究更是取得了明显的研究成果, 主要有以下几个原因。首先信道译码技术主要就是对经过 QAM 解调后的比特数据进行处理, 因此可以非常直接的当做神经网络的输入或输出信号。其次, 信道译码中所运用的码字是一种人为定义的码本, 进行样本的采集和标签的获得都非常容易, 这为神经网络的训练提供了足够多的训练数据, 与此同时, 训练样本由于信道噪声的影响增添了随机性, 防止了神经网络的过拟合。最后, 由于传统信道译码算法大多为迭代式算法, 复杂度高、运算时间较长, 难以满足系统实时传输的要求, 而基于 DL 技术的信道译码模块却可以直观的通过前向传播运算得到结果, 大大降低了运算时间。

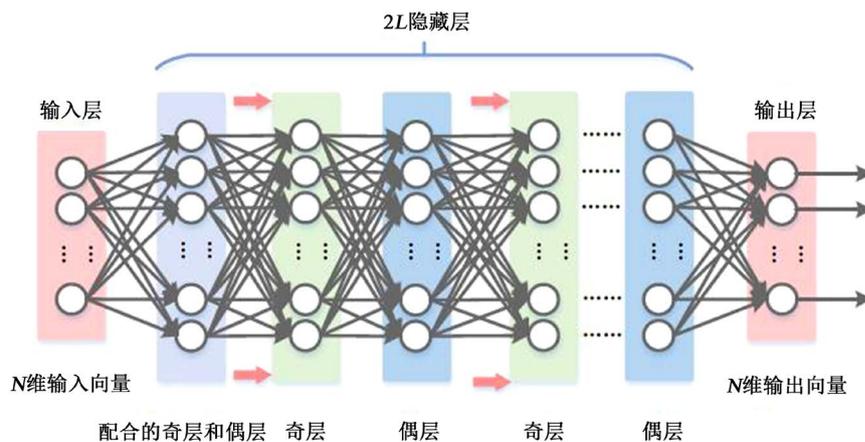


Figure 7. DNN-based BP decoder

图 7. 基于 DNN 的 BP 译码器

一种基于 DNN 架构的译码器在文献[2]中被提出, 具体结构如图 7 所示, 该译码器可以很好的改善高密度奇偶校验码的 BP 算法的译码性能。传统的 BP 译码器可以用 Tanner 图进行表示, Tanner 图内有校验节点和变量节点两种, 变量节点均和多个校验节点连接, 每进行一次迭代运算, 每个变量(校验)节点

都会将其从其他校验(变量)节点处汇总得到的信息传输给任何一个与之相连的校验(变量)节点。对于 DNN 架构来说, 可以将进行 K 次迭代的运算扩展成一个具有 $2K$ 层隐藏层的神经网络, 每层隐藏层的神经元代表 Tanner 的一条边, 由于每次迭代的 Tanner 结构不变, 因此每层隐藏层神经元个数相同, 神经元的输出即为变量(校验)节点传输至其他校验(变量)节点的信号, 与传统的 BP 算法不同的是, 由于每个神经元均对应多个与之连接神经元的权重, 因此, 此码器给 Tanner 图的每条边赋予了权重值。该译码器的输入为码字的 N 维对数似然比(Log-Likelihood ratios, LLRs), 输出为 N 个经过译码的比特。文中首先采用线性码中的全零码进行训练, 在 AWGN 信道中进行样本采样, 在含有 10 层隐藏层的架构中译码 BCH(15,11) 后获得了接近最大似然比的结果; 而在译码大密度的 BCH 码时性能虽然有所下降, 但由于经过带权的 Tanner 图的边进行信号传输可以对小环效应进行补偿, 该译码器 BER 性能仍明显优于传统的 BP 算法。文献[3]将上述的基于 DNN 架构的 BP 译码器优化成 RNN 结构, 每次迭代校验节点的输入反馈称对应的变量节点的输入, 网络的时间层与迭代次数相同, RNN 架构与 DNN 架构相比大大减少了训练参数, 但却可以获得和 DNN 架构相近的性能。

文献[4]提出了另外一种 DNN 架构的译码器, 名为神经网络译码器(Neural Network Decoder, NND), 具体架构如图 8 所示。首先发送端将含 K 个信息比特的信号经过编码器得到长度为 N 的码字, 该数据通过含 AWGN 的无线信道后由发送端接受, 发送端经过解调后将码字通过 LLR 生成器, 产生 N 维 LLR 作为译码器的输入, 通过含 3 个隐藏层的 DNN 恢复成 K 个比特。文献仿真中将 16 比特的极化码通过该译码器, 可以得到近似最大后验概率的性能。该结构网络结构简单, 不含有迭代运算, 计算延迟短。但是随着信息比特的增加, 该译码器的性能明显降低, 且由于训练过程中的 SNR 不同, 实际测试阶段的性能也会变化。对于 NND 译码器无法适用于较长码字译码的缺点, 文献[5]提供了一种新的解决方案, 考虑到上述网络可以并行的特点, 文中将极化码拆分为多个子码, 在译码器模块中使用多个上述的 NND 译码器, 每个 NND 译码一组子码, 每个 NND 模块单独训练使每个 NND 均得到近似最大后验概率的性能, 此方法为解决长码字译码提供了新的思路。

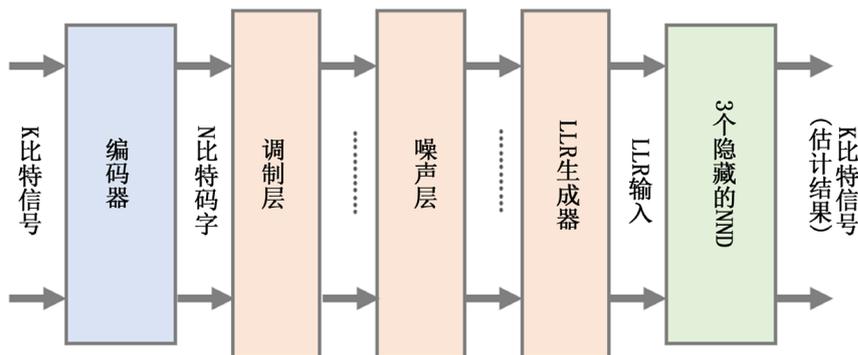


Figure 8. NND decoder structure diagram
图 8. NND 译码器结构示意图

3.3. 基于 DL 的信号检测模块

随着通信终端多样化、通信场景复杂化以及通信速率要求越来越高, 大规模 MIMO 和毫米波通信系统在性能提升和频谱资源利用等方面受到了极大的关注, 由此也带来了信道建模和信号检测复杂度的提升。而基于 DL 技术的信号检测可将传统迭代式检测算法展开, 以层次化架构高效精确的得到检测结果。文献[6]提出的一种基于 DL 技术的信号检测结构 DetNet, 该检测器将接收信号 y 和信道矩阵 H 作为输入, DetNet 将一种用于最大似然优化的梯度下降算法展开成神经网络的结构, 此算法的迭代公式为:

$$\hat{x}^{(k+1)} = \Pi \left(\hat{x}^{(k)} - \delta_k \frac{\partial \|y - Hx\|^2}{\partial x} \right) \tag{9}$$

$$= \Pi \left(\hat{x}^{(k+1)} - \delta_k H^T y + \delta_k H^T H x^{(k)} \right)$$

其中 $\hat{x}^{(k)}$ 表示第 K 次迭代的结果, δ_k 表示步长, $\Pi(\cdot)$ 函数表示非线性投影运算。将上述迭代算法展开到神经网络, 具体表达式为:

$$z^{(k)} = R \left(W_{1k} \begin{bmatrix} H^T y \\ \hat{x}^{(k)} \\ H^T H \hat{x}^{(k)} \\ v^{(k)} \end{bmatrix} + b_{1k} \right) \tag{10}$$

$$\hat{x}^{(k+1)} = \psi_{t_k} \left(W_{2k} z^{(k)} + b_{2k} \right)$$

$$\hat{v}^{(k+1)} = W_{3k} z^{(k)} + b_{3k}$$

$$\hat{x}^{(1)} = 0$$

其中 $R(\cdot)$ 表示 Relu 激活函数, $\psi(\cdot)$ 表示自主定义的分段线性软符号算子。DetNet 检测器的单层架构如图 9 所示, 若该算法迭代 K 次, 则有 K 个上图所示架构首尾连接。DetNet 检测器将所有层的输出与标签之间的差距作为代价函数, 采用 Adam 优化器进行训练。文献中在固定信道(Fixed Channel, FC)模型和变化信道(Varying Channel, VC)模型两种信道条件下进行仿真, 与传统的近似消息传递(Approximate Message Passing, AMP)算法和半定松弛(Semidefinite Relaxation, SDR)算法进行 BER 性能对比, 其中 SDR 算法可以看成是一个参考的下界。仿真结果表明, 在两种信道模型下训练, DetNet 检测器的性能均优于 AMP 算法, 且接近 SDR 算法, 但运算时间比 SDR 算法缩短了 30 倍, 这说明 DetNet 检测器在不同信道场景下具有很好的鲁棒性, 而为了进一步降低处理时间, DetNet 检测器也可以适当的减少迭代层数, 用可以接受的性能损失来均衡模块的实时处理的要求。

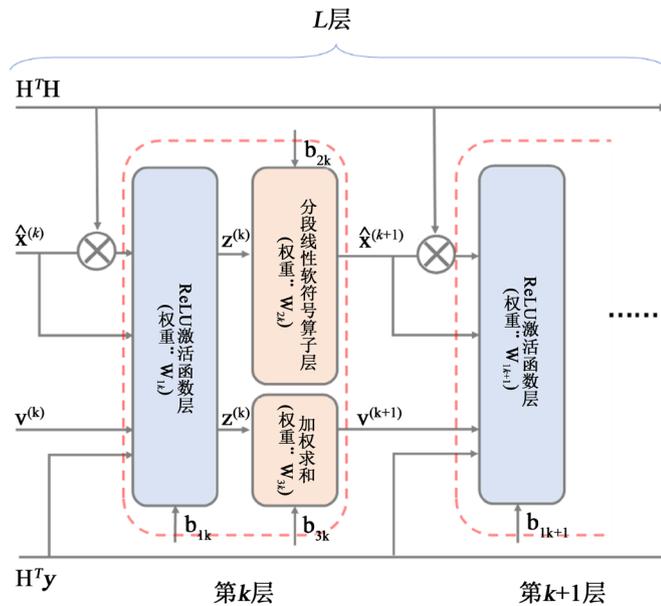


Figure 9. DetNet detector single layer structure diagram
图 9. DetNet 检测器单层结构示意图

文献[7]指出在真实的无线传输系统中, 往往存在着符号间干扰(Inter Symbol Interference, ISI)的情况, 即发射端的信号由于多径传播在接收时相互重叠而产生干扰, 此时进行序列检测而非符号检测可以得到更好的性能。因此, 文中提出一种基于 LSTM 网络架构的检测器, LSTM 属于 RNN 网络的一种。该网络含有 K 个 LSTM 层, 后接一层激活函数为 Softmax 的全连接层。Softmax 激活函数的表达式为:

$$\text{Softmax}(x) = \frac{e^x}{\sum_j e^{x_j}} \quad (11)$$

每一层 LSTM 的输出均由当前的接收信号和之前 $K-1$ 个时刻的接收信号共同决定, 充分利用 LSTM 具有记忆性的特点, 很好的消除了符号间干扰的影响。该检测器没有经过复杂的迭代运算, 网络结构简单, 运算时间较短, 与此同时, RNN 网络权重共享的特性也可以更好的减少训练参数, 加快收敛速度。仿真结构表明, 该检测器相对于传统检测算法在含有 ISI 的系统下 BER 性能提升明显。

3.4. 基于 DL 的 OFDM 接收机

DL 技术除去应用在单个物理层模块来提高性能以外, 也可作为一个整体嵌入接收机中。例如, 文献[8]中提出的基于神经网络的 FC-DNN 接收机。文中将 1 个数据符号和 1 个导频符号组成一帧, 经过 OFDM 调制(文中 OFDM 符号子载波个数为 64)和加循环前缀(Cyclic Prefix, CP)的操作化通过上行链路传输, 在接收端将经过 OFDM 解调后的复值数据分成实值和虚值两部分, 后直接输入一个 5 层全连接的 DNN 网络进行线下训练, 完成信道估计、信号检测和 QAM 解调, 引入 Sigmoid 激活函数使得最后的输出区间为 $[0,1]$, 随后引入门限判决, 即若输出的值大于 0.5, 则判定为 1; 若小于 0.5, 则判定为 0, 从而得到最后的发送比特。FC-DNN 采用 MSE 函数作为损失函数, 具体表达式为:

$$L = \frac{1}{N} \sum_{i=1}^N (\hat{b}_s(i) - b(i))^2 \quad (12)$$

其中 $\hat{b}_s(i)$ 为未经过门限判决之前的输出, $b(i)$ 为标签值即原始发送比特, N 为估计的比特数。并利用 Adam 优化器进行线下训练, 训练完成后嵌入 OFDM 接收机进行线上测试。具体架构如图 10 所示。该算

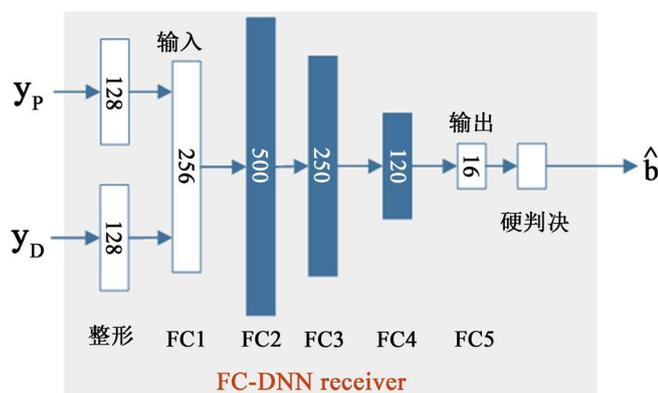


Figure 10. FC-DNN receiver structure diagram

图 10. FC-DNN 接收机结构示意图

法与传统 MMSE 算法相比, BER 性能接近, 但是当发送端在去掉 CP 或者是抑制峰均功率比等非线性条件下, FC-DNN 的 BER 性能明显优于传统的 MMSE 算法。由于 FC-DNN 采用纯数据驱动的方式搭建网络, 将整个接收机看成一个黑盒子, 使得其内部机制缺乏可解释性, 因此需要大量的样本数据进行网络训练且收敛速度较慢, 更重要的是此算法无法得到信道估计模块中的信道矩阵, 这对下行链路传输预编

码模块的功能实现产生了挑战。针对上述问题, 文献[9]提出了一种基于模型驱动的 ComNet 接收机, 该方法仍是基于 DNN 的网络架构, 没有将接收器视为一个整体, 而是将 OFDM 接收器分为信道估计模块和信号检测模块, 信道估计模块由一个仅包含输入层和输出层的神经网络组成, 信号检测模块由一个包含输入层、隐藏层和输出层的 3 层神经网络组成, 该接收机使用现有的传统通信算法的结果作为输入, 充分结合了无线通信知识, 与传统接收器和 FC-DNN 接收器相比, 此模型驱动的 DL 框架显示出更好的性能, 同时包含的训练参数更少, 并且收敛速度比 FC-DNN 接收机快。尽管 ComNet 接收机实际上是一个单输入单输出(Single Input Single Output, SISO)的接收机, 其模型驱动的思想仍然给 MIMO 接收机的设计提供了良好的思路。

4. 结束语

本文首先表明基于 DL 技术面向无线通信传输的特点与优势, 随后介绍了三种基本的神经网络结构: DNN、CNN 和 RNN 网络以及神经网络优化的基本方案, 之后总结了 DL 技术在无线通信物理层模块, 如帧同步、信道编码和信号检测中的应用成果, 以及在将神经网络替代完整的 OFDM 接收机的初步尝试。结果表明, 基于 DL 技术的算法在更为复杂的通信场景和更多通用设备接入的趋势下, 呈现出愈发明显的优势, 其强大的学习能力可以在较短时间内完成模块内部物理机制的学习, 同时具有更快处理速度, 同时拥有不逊于传统通信算法的性能, 在未来无线通信领域的研究中十分具有竞争力。

参考文献

- [1] Yadav, P., McCann, J.A. and Pereira, T. (2017) Self-Synchronization in Duty-Cycled Internet of Things (IoT) Applications. *IEEE Internet of Things Journal*, **4**, 2058-2069.
- [2] Nachmani, E., Be'ery, Y. and Burshtein, D. (2016) Learning to Decode Linear Codes Using Deep Learning. 2016 *IEEE 54th Annual Allerton Conference on Communication, Control, and Computing*, Monticello, 27-30 September 2016, 341-346. <https://doi.org/10.1109/ALLERTON.2016.7852251>
- [3] Nachmani, E., Marciano, E. and Burshtein, D. RNN Decoding of Linear Block Codes. arXiv:1702.07560.
- [4] Gruber, T., Cammerer, S. and Hoydis, J. (2017) On Deep Learning-Based Channel Decoding. 2017 *IEEE 51st Annual Conference on Information Sciences and Systems (CISS)*, Baltimore, 22-24 March 2017, 1-6. <https://doi.org/10.1109/CISS.2017.7926071>
- [5] Cammerer, S., Gruber, T. and Hoydis, J. (2017) Scaling Deep Learning-Based Decoding of Polar Codes via Partitioning. 2017 *IEEE Global Communications Conference (GLOBECOM)*, Singapore, 4-8 December 2017, 1-6. <https://doi.org/10.1109/GLOCOM.2017.8254811>
- [6] Samuel, N., Diskin, T. and Wiesel, A. (2017) Deep MIMO Detection. 2017 *IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Sapporo, 3-6 July 2017, 1-5. <https://doi.org/10.1109/SPAWC.2017.8227772>
- [7] Farsad, N. and Goldsmith, A. Detection Algorithms for Communication Systems Using Deep Learning. arXiv:1705.08044.
- [8] Ye, H., Li, G.Y. and Juang, B.H. (2018) Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems. *IEEE Wireless Communications Letters*, **7**, 114-117. <https://doi.org/10.1109/LWC.2017.2757490>
- [9] Gao, X., Jin, S., Wen, C. and Li, G.Y. (2018) ComNet: Combination of Deep Learning and Expert Knowledge in OFDM Receivers. *IEEE Communications Letters*, **22**, 2627-2630. <https://doi.org/10.1109/LCOMM.2018.2877965>