

说明材料

1、参赛项目 Git 仓库链接

<https://github.com/SiyuWang15/NAIC>

2、必要的代码级样例展示

所提方案的整体流程如以下代码所示：

- ① 从接收数据 Y 中分离出接收导频信号 Y_p 以及接收数据信号 Y_d

```
# 接收导频与接收数据划分
Y = np.reshape(Y, [Ns, 2, 2, 2, 256], order='F')
Yp = Y[:, :, 0, :, :]
Yd = Y[:, :, 1, :, :]
```

- ② 基于接收导频信号 Y_p 对信道进行粗估

```
# 第一层网络粗估信道
H1 = FC(Yp.reshape(Ns, 2*2*256))
```

- ③ 基于接收导频信号 Y_p 和接收数据信号 Y_d 以及粗估信道 $H1$ 对信道进行精估

```
# 第二层网络精估信道
H2 = CNN(torch.cat([Yd, Yp, H1]), 2)
```

- ④ 将接收导频信号 Y_d 和接收数据信号 Y_d 以及精估信道 $H2$ 输入 turbo-like 模块进行信号检测和信道估计的联合优化

```
# 第三层网络和第四层网络进行信号检测和信道估计的联合优化
H_hat = H2
N = 2
for idx in range(N):
    # SD网络
    X_hat = SD(torch.cat([Yp, Yd, H_hat], 2))
    # CE网络
    H_hat = CE2(torch.cat([X_hat, Yd, H_hat]), 2)
```

- ⑤ 利用软比特最大似然检测基于接收数据信号 Y_d 和更新的估计信道 H_hat 得到最终的恢复数据 X_bits

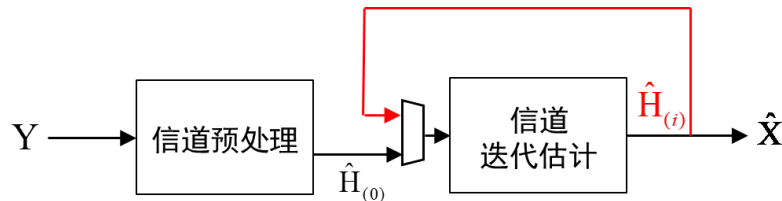
```
# 基于软比特最大似然检测得到恢复数据X
X_bits = SoftMLReceiver(Yd, H_hat)
```

（注意：以上代码可能与源码中符号不对应，仅为方便展示算法整体思路）

3、算法思路、亮点解读、建模算力与环境的说明

a) 算法思路:

本题旨在于设计接收机框架，根据接收到的信号 Y ，恢复出发射的比特 X 。传统的算法根据导频估计导频位置的信道，然后在同一帧传递的数据上进行解码判决，但是随着导频数量的降低，OFDM 截断限幅的影响，去 CP 后码间串扰的影响，对信道的估计会越来越不准确，也就导致最终的误码率性能变差。故，我们团队注意到使用神经网络的强大学习能力，使用有限数量的导频进行精确的信道估计是本题的一个关键点。围绕这一点，我们设计了两个模块（信道预处理模块和信道迭代估计模块）进行分别训练和级联验证，得到最终的对于发射比特的判决。



b) 亮点解读:

在本题目中，我们采用了两个模块（信道预处理模块和信道迭代估计模块），每个模块有多个神经网络，并对每个神经网络分别进行监督训练，最后进行级联验证的方法，在训练和级联验证的过程中，我们主要有以下几个亮点：

首先，在每个神经网络的分别训练的基础上，我们分别在两个模块（信道预处理模块和信道迭代估计模块）中联合 **retrain** 其中的神经网络参数，其中 **retrain** 的目的是为了让每个模块中的多个神经网络能够更好地互相匹配和拟合，并得到了更好的性能提升。

其次，我们对训练好的网络进行迭代展开，如上图红线所示，信道迭代估计模块中的上一层的输出会输入到本层，进行多步的循环并最终得到对发射比特的判决结果。其基本思想如下：由于课题中导频数目非常少，仅依靠导频做信道估计，精度非常差。我们在第一个模块信道预处理结束后，已经能够得到信道的初步估计，经实验验证，对于 8 导频情况此时已有 88% 的 bit 准确率，这意味着，在 88% 的位置上的 bit 都是正确的，这些正确估计的 bit，可以作为额外的导频，对信道估计的结果进行修正。具体而言，我们在信道迭代估计模块首先使用一个信号检测网络输出每一个 bit 为 1 的概率，概率越接近于 0 或 1，这一位的检测置信度越高，对信道估计的修正作用越强。因此，我们将信号检测网络的输出置信度结果直接输入至第二层信道修正网络，输出修正后的信道。同理，修正后的信道依旧能通过信号检测网络输出新的检测 bit 置信度，我们又可以将新的置信度再次输入信道修正网络。以此往复形成迭代循环。这其中蕴含了 turbo 码迭代译码的思想，经实验验证，在迭代 2-3 轮时，准确率性能达到最优。最后拿到信道的最终修正值，与接收数据信号做逐比特译码，计算各个 bit 位的软信息进行最终的判决。

第三，我们在信道预处理模块中信道精估计网络使用了两种神经网络分别实现该信道精估计功能，分别为 Resnet18 和 Resnet34，同时在信道迭代估计模块中的信号检测神经网络中，我们使用了三种方法分别实现该信号检测功能，分别为 Resnet18、MRC-net 和软比特最大似然。那么在实际验证测试的过程中会存在不同的信道预处理模块中信道精估计网络和信道迭代估计模块中信号检测的组合，比如 Resnet18-Resnet18，Resnet18-MRC-net，Resnet18-软比特最大似然和 Resnet34-Resnet18，Resnet34-MRC-net，Resnet34-软比特最大似然六种组合方式，并使用这六种模型中对测试数据进行分别测试，最后我们选取区别度最大的五个模型使用集成学习进行投票得到最终对测试集 X 的估计。

c) 建模算力和环境

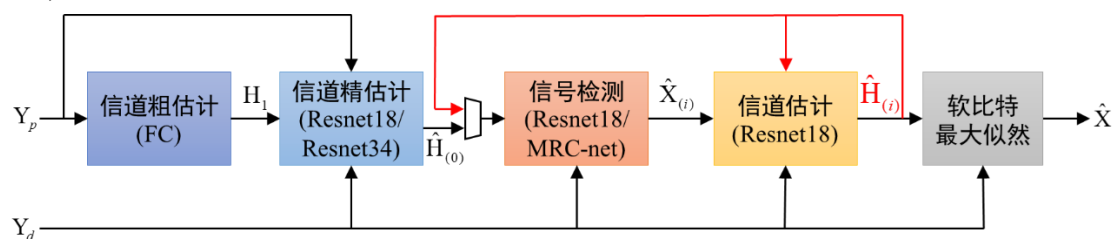
算力：8*Tesla P100*(24h*10days)

环境：

python	3.6
pytorch	1.4.0
CUDA	10.2

4、详细的解题思路说明

a) 算法框架图：



b) 符号解释：

符号	符号解释
Y	无线通信接收端收到的信号
Y_p	无线通信接收端收到的 导频 信号
Y_d	无线通信接收端收到的 数据 信号
H	真实信道
H_1	信道粗估计模块得到的 粗估计 信道
$\hat{H}_{(i)}$	第 i 步迭代中信道精估计模块得到的 精估计 信道
X	无线通信发射端 发送的比特
$\hat{X}_{(i)}$	第 i 步迭代中信号检测模块估计得到的 软比特（置信度）
\hat{X}	软最大似然估计得到的 最终比特判决

c) 解题思路说明：

本题旨在于设计接收机框架，根据接收到的信号 Y ，恢复出发射的比特 X 。传统的算法根据导频估计导频位置的信道，然后在同一帧传递的数据上进行解码判决，但是随着导频数量的降低，OFDM 截断限幅的影响，去 CP 后码间串扰的影响，对信道的估计会越来越不准确，也就导致最终的误码率性能变差。故，我们团队注意到使用神经网络的强大学习能力，使用有限数量的导频进行精确的信道估计是本题的一个关键点。围绕这一点，我们设计了两个模块（信道预处理模块和信道迭代估计模块）进行分别训练和级联验证，得到最终的对于发射比特的判决 \hat{X} 。

其中，信道预处理模块实现对信道 \mathbf{H} 的初步精估计 $\hat{\mathbf{H}}_{(0)}$ 。该模块包含两个网络，第一个网络是实现信道粗估计的全连接（FC）网络，其输入是 \mathbf{Y}_p ，输出是 \mathbf{H}_1 ，以完美信道 \mathbf{H} 作为标签，并进行监督学习；第二个网络是实现信道精估计的卷积神经网络（CNN），注意到接收到的数据信号 \mathbf{Y}_d 对应的发射符号是 QAM 调制，而 QAM 调制的符号模为 1，故 \mathbf{Y}_d 中至少包含着信道的幅度信息，所以第二个神经网络的输入包含 \mathbf{H}_1 ， \mathbf{Y}_p 和 \mathbf{Y}_d ，输出是 $\hat{\mathbf{H}}_{(0)}$ ，以完美信道 \mathbf{H} 作为标签，并进行监督学习。此信道预处理模块通过两个神经网络的使用和接收信号 \mathbf{Y} 的充分利用，得到了对于信道 \mathbf{H} 的初步精估计 $\hat{\mathbf{H}}_{(0)}$ 。

进一步，我们使用信道迭代估计模块对信道预处理模块估计的信道 $\hat{\mathbf{H}}_{(0)}$ 进行进一步估计。该模块包含两个网络，第一个网络是使用 Resnet18 或 MRC-net 实现对信号的检测功能，其输入是上一步估计的信道 $\hat{\mathbf{H}}_{(i-1)}$ 和 \mathbf{Y}_d ，在输入之前通过选择门实现对上一步信道的选择功能，输出是 $\hat{\mathbf{X}}_{(i)}$ ，以发射比特 \mathbf{X} 作为标签，并进行监督学习；第二个网络是使用 Resnet18 实现对信道的进一步估计，其中输入是上一个神经网络的输出 $\hat{\mathbf{X}}_{(i)}$ ，以及上一步估计得到的信道 $\hat{\mathbf{H}}_{(i-1)}$ 和 \mathbf{Y}_d ，输出是 $\hat{\mathbf{H}}_{(i)}$ ，以完美信道 \mathbf{H} 作为标签，并进行监督学习。此信号迭代检测模块通过使用训练好的两个神经网络来进行多次迭代，得到对于信道 \mathbf{H} 的进一步估计 $\hat{\mathbf{H}}_{(i)}$ 。

最后，我们使用软比特最大似然检测器输出最终对于发射符号的判决 $\hat{\mathbf{X}}$ ，其输入是接收到的数据信号 \mathbf{Y}_d 以及对信道的精估计结果 $\hat{\mathbf{H}}_{(i)}$ ，输出是最终判决结果 $\hat{\mathbf{X}}$ 。

值得一提的是，在信道预处理模块中信道精估计网络使用了两种神经网络分别实现该信道精估计功能，分别为 Resnet18 和 Resnet34，同时在信道迭代估计模块中的信号检测神经网络中，我们使用了三种方法分别实现该信号检测功能，分别为 Resnet18、MRC-net 和软比特最大似然。那么在实际验证测试的过程中会存在不同的信道预处理模块中信道精估计网络和信道迭代估计模块中信号检测的组合，比如 Resnet18-Resnet18，Resnet18-MRC-net，Resnet18-软比特最大似然和 Resnet34-Resnet18，Resnet34-MRC-net，Resnet34-软比特最大似然六种组合方式，并使用这六种模型中对测试数据进行分别测试，最后我们选取区别度最大的五个模型使用集成学习进行投票得到最终对测试集 \mathbf{X} 的估计。

5、项目运行环境和运行办法等信息，根据该文档操作指引，能复现选手结果；（一般为：运行 run.sh）

Codes for AI wireless communication contest Stage2.

Our codes are built mainly on pytorch1.6.0. To quickly start, run

`pip install -r requirements.txt`

All commands can be found in `run.sh`.

To generate final submitted files, please uncomment line 23-31 in `run.sh`.

To train our model, please read comments and use corresponding commands, if you have any questions, feel free to contact us.

6、如有额外数据集，需同时提供数据集的下载链接和数据说明，并阐述详细的训练过程
无

7、其他需要说明的内容
无