README

1) GitHub链接

https://github.com/BenedictYoung/naic2020_THU_CS/tree/main/Contest

2）样例展示

我们的模型可以分为三个模块，信道估计模块、信号检测模块和集成学习模块，各个模块的关键步骤如下:

1.信道估计模块

1.1 模型定义

```
# model define
if args.m == 34:
    model = resnet34()
elif args.m == 50:
    model = resnet50()
elif args.m == 121:
    model = densenet121()
elif args.m == 169:
    model = densenet169()
else:
    model = densenet201()

model = torch.nn.DataParallel(model).cuda()
```

1.2 模型训练

```
# training
model.train()
for step, (Y, X, H) in enumerate(train_data_loader):
    sample = torch.as_tensor(Y, dtype=torch.float, device='cuda')
    label = torch.as_tensor(H, dtype=torch.float, device='cuda')
    predict = model(sample)
```

```
    loss = loss_func(predict, label)
    optimizer.zero_grad()
    loss.backward()
    optimizer.step()
```

2.信号检测模块

2.1 模型定义

```
# model define
model = resnet18(1024)
model = torch.nn.DataParallel(model).cuda()
```

2.2 模型训练

```
# training
model.train()
for step, (Y, X, H) in enumerate(train_data_loader):
    H = H.detach().cpu().numpy()
    Hf = Ht2Hf(H)
    Hf = HfComplex2Float(Hf)

    Y = Y.detach().cpu().numpy()
    Y = np.reshape(Y, (-1, 2, 2, 2, 256), order='F')
    Yd = Y[:, :, 1, :, :]

    sample = np.concatenate((Yd, Hf), axis=2)
    sample = np.reshape(sample, [-1, 1, 12, 256])

    sample = torch.as_tensor(sample, dtype=torch.float, device='c
uda')
    label = torch.as_tensor(X, dtype=torch.float, device='cuda')
    predict = model(sample)

    loss = loss_func(predict, label)
    optimizer.zero_grad()
    loss.backward()
    optimizer.step()
```

3.集成学习模块

3.1 投票集成

```
#
data = []
for m in models:
    data_batch = read_data('./Temporary/X_pre_' + str(index) + '_
' + str(m) + '.bin')
    data_batch = torch.tensor(data_batch, dtype=torch.long)
    data.append(data_batch)
data = torch.stack(data, dim=0)
X_final, X_indices = torch.mode(data, dim=0)
X_final = X_final.numpy().astype(np.bool)
return X_final
```

3）算法亮点

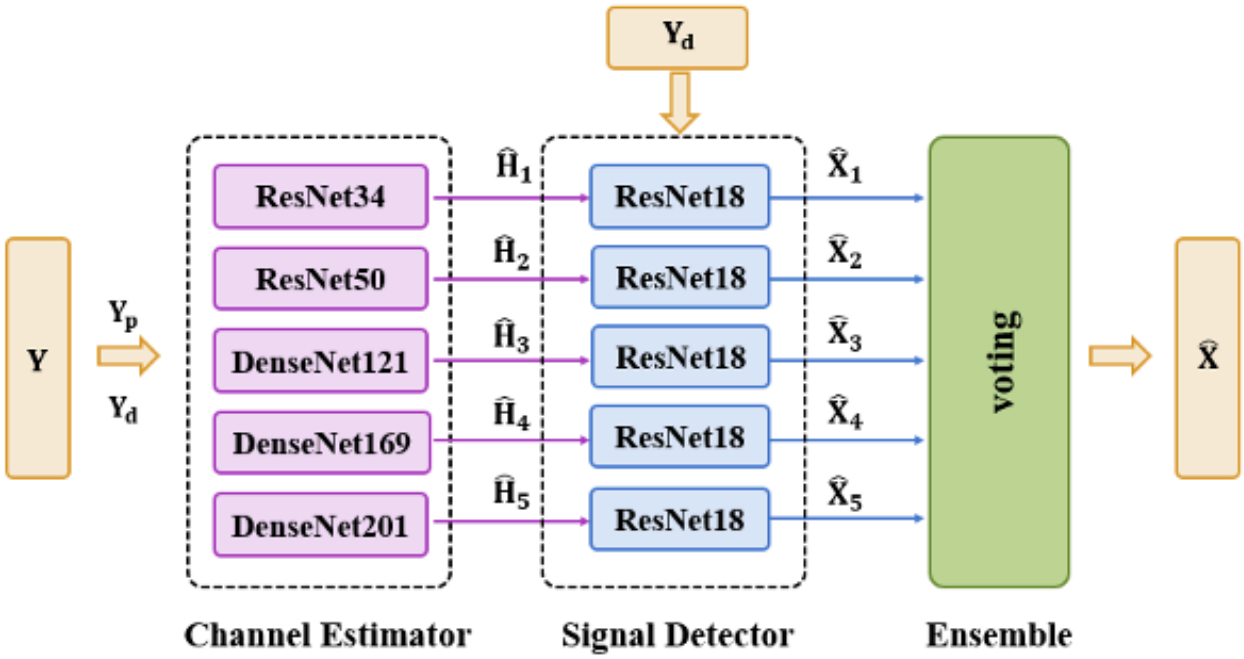

图1 整体算法设计

算法的总体思路如图1所示，针对OFDM接收机设计问题，根据传统通信系统设计方法，本队伍将该问题拆分为信道估计(Channel Estimator)和信号检测(Signal Detector)两个模块，分别建模为不同的神经网络模型进行单独训练。

信道估计模块目的是为精确恢复信道信息，以接收导频和接收数据为网络输入，以估计信道数据为网络输出；信号检测模块目的是为恢复发射数据，以接收数据和前一级估计信道为输入，以恢复的数据信息为输出。最终，由于信道估计模块采用不同深度、不同架构的神经网络模型，本项目以集成学习(Ensemble Learning)方法将不同网络恢复的发射数据进行集成，得到高精度的恢复数据。

本项目的亮点体现在以下两个方面：

1.数据和模型联合驱动：将整个接收机拆分为信道估计和信号检测两个部分，相比于联合训练方法，模块化设计降低单个网络的训练复杂度，并获得比联合训练更好的性能。初步分析可能是ResNet18的信号检测网络对信道估计误差有一定的容忍度，而联合训练可能会使得误差传播的过程中越来越严重。

2.集成学习：由于信道估计精度对于本题目效果影响显著，因此本队伍对信道估计采用多种网络训练，并且最后使用了集成学习投票的方法利用多个弱监督网络形成强监督网络，获得了稳定且高精度的预测准确率。

4）解题思路

利用无线通信中的相关知识，全数字接收机的结构如图2所示。根据赛题，我们不考虑同步之前的接收机内容，所以我们需要从信道估计开始设计基于AI方法的接收机。在进行信号检测解码前，需要进行信道估计。信道估计的性能将严重影响信号检测的效果。所以我们总体思路是将信道估计和信号检测分开处理，分别使用深度学习方法进行设计。

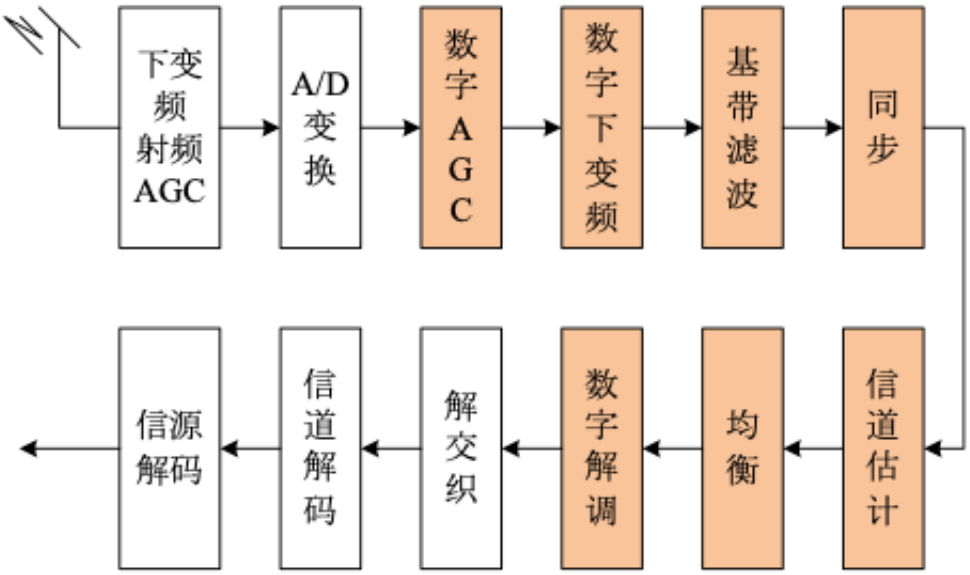

图2 全数字接收机结构

首先针对信道估计问题，我们使用了CNN的结构来进行信道估计。由于无线信道中存在复杂的衰落，多径，干扰和噪声等，存在着大量的非线性关系。所以我们分别训练了多个不同的基于CNN架构的神经网络，从中选出5个最优的结构——分别为：**ResNet34**、**ResNet50**、**DenseNet 121**、**DenseNet 169**、**DenseNet 201**。网络在训练阶段，根据大赛提供的代码，我们随机生成发送数据“X”，通过“H.bin”获得标签“Y”，由此来训练神经网络。用同样的方式通过“H_val.bin”文件生成验证集，评估训练准确度。针对题目给定的接收数据，训练后网络的输入即赛题中的接收信号数据，同时输入导频和传输数据，输出即为估计的时域信道信息。

针对信号检测问题，我们采用相同的**ResNet18**网络架构分别训练。经过信道估计模块，我们获得了5个估计的时域信道，将估计信道“$\hat{H}$”和接收信号“Y_d”作为网络的输入，分别送入5个**ResNet18**估计发送的信号“X”。由于信道估计网络存在差异，可以得到5个不同的参数的**ResNet18**网络。

由于信道估计问题的复杂性，尤其是对于8导频的问题，信道估计难度很大。所以我们尝试采用了结构、层数存在差异的网络去训练得到不同的模型。然而信号检测解码的难度相对较小，因此采用相同结构的基于CNN结构网络完成不同信道估计后的信号检测，模型结构类似但参数不同。

通过上述步骤，可以获得5个信号检测结果。为进一步提升预测性能及将这5个模型的优势相互结合，我们采用了集成学习(Ensemble Learning)的方式去获得一个更好的模型，相互纠正预测结果中的错误，使得多个弱监督的模型变成一个全面的强监督的模型，稳定输出的准确率。具体来说，我们使用了绝对多数投票法预测标记，获得了比单个模型优越的性能。

5）运行环境与方法

1.运行环境

表5.1 运行环境说明表

环境	版本
Python	3.8.5
Numpy	1.19.2
Pytorch	1.7.0
CUDA	11.0

2.文件说明

表5.2 文件说明表

文件名	描述
res_net.py	模型定义
dense_net.py	模型定义
data_generator.py	数据生成器
train_estimator.py	信道估计器训练
train_detector.py	信号检测器训练
channel_estimator.py	信道估计器
signal_detector.py	信号检测器
voter.py	集成投票器
funcitons.py	必要函数集合
utils.py	官方文档
Pilot_16	官方文档
Pilot_64	官方文档

表5.3 文件夹说明表

文件夹	描述
DataSet/	存放官方提供的数据集
Networks/	存放训练好的模型参数
Temporary/	存放预测过程中间数据
Result/	存放最终预测结果数据
Other/	存放官方要求提交材料

3.运行方法

```
bash run.sh
```

4.结果位置

```
Result/submit.zip
```

6) 额外数据集

无

7）其他说明内容

无