# 无线通信\_无名王者\_模型自验报告

（吴铭晖、1461610587@qq.com）

# 模型简介

采用端到端训练方案，神经网络的输入为接受信号$Y$，输出为解调后的估计信号X，其中对于输入的导频部分我们做了一些处理，只保留了插入导频子载波位置的接收信号，对于32导频的就是每8个子载波插入一个导频，由于错开半个导频间隔的位置，因此在接收端每隔4个子载波提取一个导频，其他位置视作噪声丢弃可以减少无用噪声的影响。

## 网络模型结构简介：

将输入的接收信号Y拆分成两部分，第一部分为导频信号Yp,第二个为数据信号Yd,导频信号Yp首先展开经过一次dense层维度增大一倍并重构成维度为N\_c × 16的数据流，可以将其看作是初步隐形的信道估计，随后采用补零的维度不变的卷积残差网络进一步细化我们的隐式估计信道，为了之后的神经网络模拟信道均衡与解调的过程因此将这个数据流看作是非显示的估计信道并且为了能提供更多信息比真实频域信道扩大一倍。然后将此数据流与数据信号在第三个维度拼接并经过一次卷积将通道数扩大以获取更多的特征。随后并经过第二个卷积残差网络，最后再经过一次卷积层输出通道数为4将输出数据维度变换为1×Nc×4最后经过sigmoid激活函数并判决即为输出的预测比特流，将其按照天线与子载波与调制顺序的关系重新排列成长度为1024的比特流，网络中总共有两大块卷积残差网络，第一块是为了隐式估计信道，第二块是为了解调数据考虑子载波间的相关性，其中第二块对于通道采用attention机制。

## 数据集：

训练集与验证集包含真实的传输比比特和MIMO-OFDM信道和接受数据。测试集包含传输比特和接收数据。

## 代码提交地址：

U盘直接拷贝

# 代码目录结构说明

- data

- X\_pre\_1.bin

- X\_pre\_2.bin

- Y\_1.csv(请自行拷贝测试集)

- Y\_2.csv(请自行拷贝测试集)

- pb\_model

- model\_8.best.pb(请从百度网盘下载)

- model\_32.best.pb(请从百度网盘下载)

- 无线通信\_无名王者\_模型自验报告.docx

- eval.py

- outputMessage.txt

- README.md

- outputMessage.txt

- run.sh

# 自验结果

（包括所用MindSpore版本、自验环境、自验精度结果、自验训练速度结果、论文精度等，推荐多提供截图参考。提供自验结果截图或日志文件）

## 自验环境：

Ascend910+Tensorflow1.15

## 推理：

### 如何启动推理脚本：

推理如何启动：

eval.py如何启动

在终端行输入

export install\_path=/usr/local/Ascend # 软件包安装路径，请根据实际修改

# driver包依赖

export RANK\_ID=0

export LD\_LIBRARY\_PATH=/usr/local/Ascend/add-ons:$LD\_LIBRARY\_PATH

# fwkacllib包依赖

export LD\_LIBRARY\_PATH=${install\_path}/fwkacllib/lib64:$LD\_LIBRARY\_PATH

export PYTHONPATH=${install\_path}/fwkacllib/python/site-packages:${install\_path}/fwkacllib/python/site-packages/auto\_tune.egg/auto\_tune:${install\_path}/fwkacllib/python/site-packages/schedule\_search.egg:$PYTHONPATH

export PATH=${install\_path}/fwkacllib/ccec\_compiler/bin:${install\_path}/fwkacllib/bin:$PATH

# tfplugin包依赖

export PYTHONPATH=${install\_path}/tfplugin/python/site-packages:$PYTHONPATH

# opp包依赖

export ASCEND\_OPP\_PATH=${install\_path}/opp

export JOB\_ID=10087

export ASCEND\_DEVICE\_ID=0

python eval.py

run\_eval.sh如何启动：

终端输入 ./run.sh即可运行

如果权限不够在之前先chmod +x ./run.sh

### 推理精度结果：

1P推理结果：



可以看到32导频的测试集在39秒内推理完毕

8导频的测试集在32秒内推理完毕，npu性能得到充分利用。

# 参考资料

## 参考git项目：

华为云昇腾+Tensorflow2.1推理指南(https://support.huaweicloud.com/ug-tf-Inference-tensorflow/atlasoiug\_26\_0002.html)