上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

课程论文

COURSE PAPER

论文题目：Project 14: Spectrum Resource Prediction

学生姓名：冯靖皓, 蒋铭阳, 郑文卓, 沈天乐, 严涵

（按姓名顺序）学生学号：519030910362, 519030910384, 519030910360, 519030910403, 519030910404

课程名称：智能物联网

指导教师：田晓华教授

学院(系)：电子信息与电气工程学院

Contents

0.1 Introduction

Imagine that you are working in a classroom where many devices are transmitting and receiving signals, and they all work near 2.4GHz in the ISM band. At this time, you want to use a device to send data, but unfortunately, there is a frequency band conflict and the information cannot be transmitted successfully. "It's so bad," you may think, "can there be a more clever signaling strategy?" And now there is it. Our group propose an effective method to predict the frequency band occupancy based on machine learning, so as to better plan the signal transmission and reception strategy and reduce the frequency band conflict.

We observe that the band occupancy information is very suitable for learning with temporal sequential networks, based on which, we propose a short-time spectrum occupancy prediction network. The main contributions of this paper are summarized as follows:

- We propose a short-time spectrum occupancy prediction network based on LSTM
- We propose a novel metric: Collision Rate
- We achieve real-time prediction on small PC and Linux virtual machine
- We cooperate with group 13 to form a real-time monitoring and prediction mechanism

Figure 0–1 shows the framework of our system, which mainly consists of data preprocessing module and deep network module.

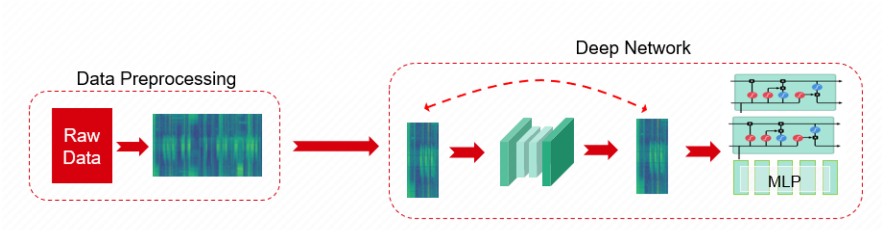

Figure 0–1 Framework of our system

0.2 Data Preprocessing

In the data preprocessing module, we transfer the raw data from time domain to frequency domain through STFT, and divide the frequency domain axis equally into a certain number of bins, which is shown in figure 0–2.We then take the energy of each bin as it's value. Here, we actually convert the raw data into a vector sequence.

To suppress noise, we turn vectors into dBm unit measurement, then we apply a threshold to binarize the vectors into 0-1 values, representing whether a bin is occupied. There are two reasons for this. On the one hand, we only need to know whether a bin is occupied at a certain time, rather than the specific bin value; On the other hand, after binarization, our LSTM network can be better trained and get more accurate prediction results.

0.3 Network Architecture

We regard frequency prediction as a temporal sequential prediction task, in which a lot of work has made great progress, such as RNN, LSTM, Transformer... Considering the real-time require-

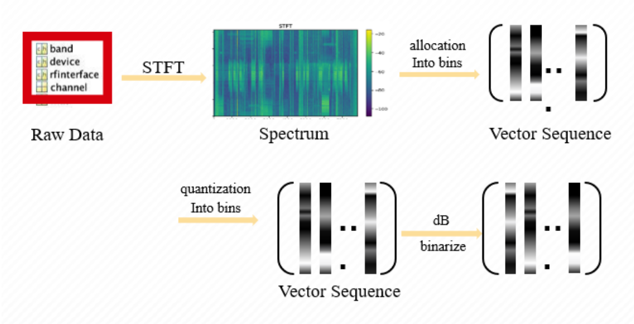

Figure 0–2 Data Preprocessing

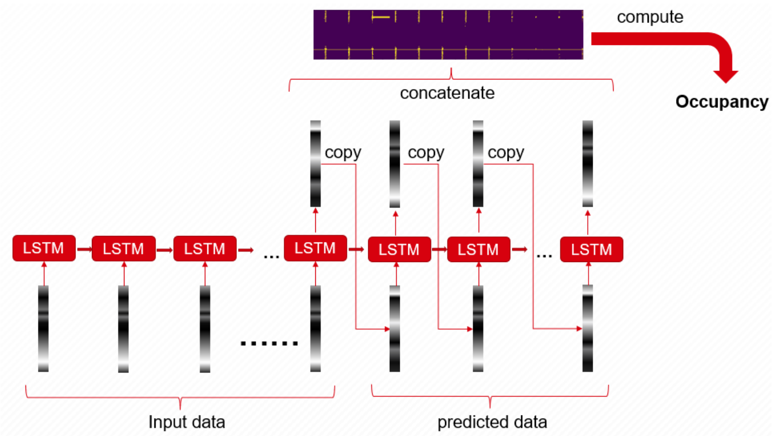

Figure 0–3 Vision1

ments of our work, we prefer a lightweight and efficient network. Long short-term memory(LSTM), a popular network that can remember historical data selectively, becomes the core of our network architecture. On this basis, it takes four iterations of improvement to produce the final version.

0.3.1 Ver1

Inspired by Qiu's "asynchronous sequence to sequence pattern" network, we design the first version of our network. The architecture is shown in Figure0–3. Given input sequence $\{x_1, \ldots, x_I\}$ of length I, the network recursively outputs sequence $\{y_1, \ldots y_O\}$ of arbitrary length (here we set the output length is O). Then we concatenate all outputs to generate a matrix with the size of B-by-O, where B is the length of a feature vector. Finally, we fit this matrix to the true STFT diagrams using L2 loss function.

The visual result Figure0–4 is promising, which seems to have learned the part of the cycle information and proves the feasibility of the network in return. However, it is time-consuming to use STFT diagram for supervision. For one thing, the size of the target matrix is proportional to the length of the predicted time. For another, the STFT diagram contains many redundant information as well as some undesirable noise.

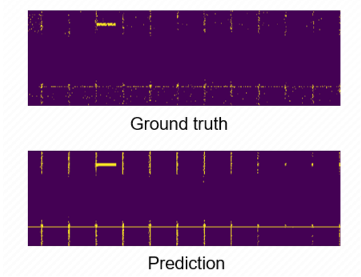

Figure 0–4 Vision1_result

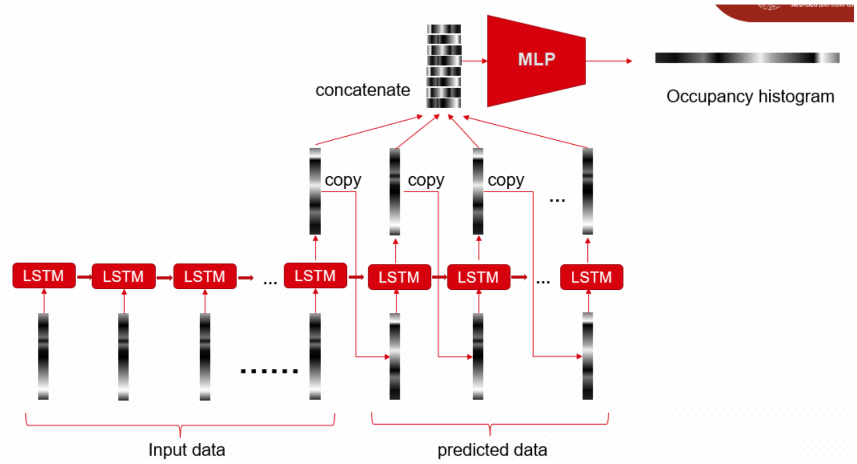

Figure 0–5 Vision2

0.3.2 Ver2

As discussed in Ver1, the generated B-by-O matrix can potentially capture the desired information of the signal. Therefore, we can confidently map it to a histogram which is expected to represent the occupancy in the following period of time by using a MLP. The architecture is shown in Figure0–5.

Compared with Ver1, Ver2 reduces a lot of time. Due to the recurrent structure of prediction, however, The time scale of what can be predicted is limited.

0.3.3 Ver3

Ver3 is an update of Ver2, which drops out the recurrent structure. Intuitively, the LSTM can capture the temporal feature of the input data, including the potential period or beacon. Therefore, the recurrent structure is redundant. The architecture is shown in Figure0–6.

0.3.4 Final Ver

During the cooperation with Group13, whose work is to realize the signal compression and decompression with less loss on the signal, we found that their encoder-decoder component can effectively suppress noise. Inspired by their work, we design a simple encoder-decoder network. The difference is that we use the encoded signal to pass through the LSTM and simultaneously

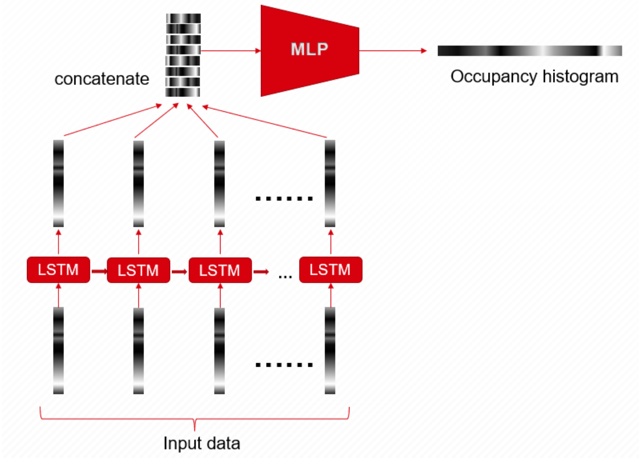

Figure 0–6 Vision3

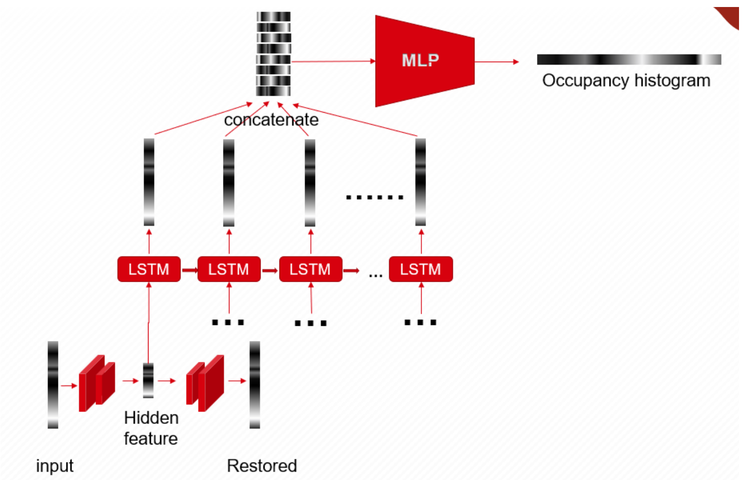

Figure 0–7 Vision4

expect the decoded signal more similar to the original by MSE loss. The architecture is shown in Figure0–7.

0.4 Results

Here, we propose a brand new evaluation index——Simulated Collision Rate(SCR). Our task is to predict the occupation of spectrum resources, and the purpose of doing so is to improve the efficiency of information transmission. Thus, based on our prediction, we apply some transmission strategy and quantify the collision rate by SCR. Formally, the SCR we use is:

$$SCR = \frac{N_{bincollision}^{(strategy)}}{N_T} \tag{0–1}$$

where N_T denotes number of time points, and$N_{bincollision}^{(strategy)}$ denotes number of bin collision(transmission bin chosen by the strategy is occupied by ground truth)

We apply 3 different strategies for SCR.

Network Vesion	Random'SCR	General Random'SCR	Ideal Optimal'SCR	Collision Avoidance	Realtime : Pred Time
v2	19.32%	18.54%	8.83%	54.33%	0.197
v3	19.32%	16.88%	7.46%	61.41%	15.710
v4(100:800)	19.32%	**16.75%**	**6.78%**	**64.92%**	13.160
v4(100:1600)	19.32%	16.93%	6.95%	64.04%	26.278
v4(200:1600)	19.32%	16.96%	7.03%	63.60%	**29.782**

Table 0–1 Results based on Simulated Collision Rate

- Random strategy: choose transmission bin uniformly at random.
- General Random strategy: choose transmission bin based on a distribution D, where D = softmax(1 −occupancy rate), and transmit the data.
- Ideal Optimal strategy: choose the bin with minimum occupancy rate, and transmit the data.

In table 1, the Realtime : Pred Time means the ratio of the time we predict to the time it takes to predict, and this ratio > 1 indicates that the real-time requirements can be met. Also, the Collision Avoidance in table 1 represents the degree of decrease in the collision rate, when we take the random strategy as the baseline. We can see that our network can access the real-time requirements and greatly reduce the collision rate in signal transmission.

0.5 Cooperation with Group 13

In order to explore the feasibility of coupling our method with upstream tasks like spectrum resource monitoring, and verify the effect of our network in practical application scenarios, we co-operated with team 13. The task of team 13 is spectrum resource monitoring. By combining our system with theirs, we get a Full-chain ‘Monitor-Predict-Transmit’ system——Group 13 collect spectrum information from the environment and design an algorithm to compress the collected information, we then use the compressed data to predict the frequency band occupancy. In addition, according to our prediction, we designed a smarter signal transmission strategy to achieve less signal collision and higher signal transmission efficiency.

0.6 Extra Analysis

In response to Prof. Tian's queries that 'If some devices in the working environment change their states, our network might lose its performance', we give the following analysis and solutions.

First, the time period that devices change their states, such as switching from sleeping to activation, is relatively long, usually in minutes or even hours. Therefore, during training, we should sample datasets of the same level length. However, to sample data from ISM frequency band requires high sample frequency, the data density of dataset is so high. According to rough estimation, to sample dataset of 30 minutes, we need about 1.2 TB storage space, and 40 minutes of training time. It's easy to see that it's impossible to verify our network's accuracy based on long datasets.

To solve this contradiction, we propose to sample 2 datasets on different time and places, one of which is used as train set, the other as test set. Moreover, we let our network dynamically adjust its parameters through periodic back-propagations. For example, we can do BP every 5 epochs, so long as it's under real-time constraint's tolerance.

BP or not	Mean Loss	Random'SCR	Ideal Optimal'SCR	Collision Avoidance	Realtime : Pred Time
no BP	6.90%	5.61%	5.36%	4.46%	23.84
BP	2.29%	5.61%	4.30%	23.28%	20.33

Table 0–2 Results based on Simulated Collision Rate

If this method is proved to reduce collision rate effectively, that means our work is not in vain. Here we must clarify that our network can support online learning. In our network, training accounts for most of the computation time, while testing is adequately fast. Therefore, doing BP every few epochs will not affect real-time performance.

We did some experiments with our dataset and group 13's. Specifically, we use our dataset as train set and group 13's as test set, and check if our network can reduce collision compared to random strategy. Besides, we compare performance with and without back propagation every 10 epochs. Here we list some results in table 0-2.

As we can see from table 0-2, with back propagation once in a while, we get about 1/4 collision avoidance. That means although train and test sets have totally different distributions, our network can still acquire apparent effects.

Besides, the modification doesn't affect real-time performance greatly, our network can still guarantee real-time working.