



Supplementary materials for

Xiang LI, Yibing LI, Chunrui TANG, Yingsong LI, 2023. Modulation recognition network of multi-scale analysis with deep threshold noise elimination. *Front Inform Technol Electron Eng*, 24(5):742-758.
<https://doi.org/10.1631/FITEE.2200253>

1 Modulation classification methods

Modulation classification methods are divided into manual and automatic modulation classification techniques. Manual modulation classification relies on the down-conversion of the received high-frequency signal to determine the type of signal modulation through devices such as an oscilloscope, spectrum analyzer, or demodulator. Manual classification recognizes limited types and has high complexity. Compared to the complex manual modulation recognition, machines automatically carry out the existing modulation recognition methods (Wang et al., 2019).

2 Architecture of the signal transceiver system

The read data source is sent through the transmit (TX) channel in the universal software radio peripheral (USRP) hardware driver (UHD) sink. USRP is a flexible and powerful general-purpose software radio peripheral. The low cost of wide broadband makes USRP cost-effective (Zitouni and George, 2016). During the transmission process, the signal processing inside the USRP is divided into two stages. In the first stage, the high-speed digital signal processing field programmable gate array (FPGA) in the mother board converts the digital baseband signal in the computer into a digital intermediate frequency signal. After sending control and digital up-conversion by FPGA, the signal is converted into data in the analog domain through the digital to analog converter (DAC) module. In the second stage, the child board filters the intermediate frequency (IF) signal in the analog domain to smooth the signal, and then multiplies it with the crystal oscillator signal to obtain the radio frequency (RF) signal. The signal radiated by the antenna is transmitted through the radio environment. Then the signal is received through the receive (RX) channel. The child board's low noise amplifier and crystal oscillator down convert the signal from RF to IF and perform filtering and smoothing to prevent aliasing. After that, the analog to digital converter (ADC) in the mother board performs the analog-to-digital conversion and sends signals to the FPGA for digital down-conversion and receiving control. Gnu's not unix (GNU) radio completes the establishment of communication with USRP by calling the application programming interface (API) provided by the UHD driver (Liu et al., 2017). The QT graphical user interface (QT GUI) time sink module in GNU radio is responsible for representing the signal. After giving a complex number to the input module, the module outputs both the real part and imaginary part of the signal, and it can be judged whether the transmission and reception are completed through the signal figure in GNU radio. The in-phase and quadrature components of the acquired signal is transferred to the computer through the file sink module and saved as a file of the corresponding modulation type. The highly compatible integration of software and hardware platforms facilitates complex signal processing.

3 Network in the signal recognition system framework

The designed network introduces batch normalization (BN) to solve the gradient explosion and disappearance caused by reverse gradient propagation. Besides, the data distribution after BN tends to be stable so that the subsequent network layers can learn features based on appropriate data distribution and accelerate the convergence of the loss function. We select the LeakyRelu activation function to enhance the nonlinearity of the network. In addition, when LeakyRelu takes a negative value, it has a slight slope to solve the problem that the input data neurons stop learning. The network introduces a dropout mechanism to increase the sparsity and randomness of the network design and avoid spending more time on learning unimportant features.

4 Autoencoder

As one of the mainstream architectures in deep learning, the purpose of the autoencoder is to minimize error so that the output reconstructs the input (Bengio et al., 2013). However, the basic autoencoder learns only with a single hidden layer, which makes it easy to obtain a linear mapping result. Researchers proposed a deep autoencoder, setting multiple hidden layers and training the network by backpropagation to solve the problem that a single hidden layer autoencoder simply copies the input as output (Hinton and Salakhutdinov, 2006). The deep autoencoder can efficiently learn the hidden layer representation of the input data to obtain more objective and complete features. The algorithm model contains the encoder and the decoder. The function of the encoder is to encode the input data \mathbf{x} as a latent variable \mathbf{h} by feature extraction and capture the most significant features of the neural network. The decoder converts the extracted features into \mathbf{x}^R through the decoding operation, restoring the features to the original dimension.

5 Deep residual network

The neural network learns more sophisticated features with more network layers, and the training achieves better system recognition. However, when the layers are too deep, the network will face degradation, which means that the performance rapidly reaches saturation or even decreases. Assuming that the model at layer m has been trained to the optimal network at layer n (where $m > n$), the network at layers $m-n$ is redundant. Due to a nonlinear activation function, the redundant network of layers $m-n$ generates irreversible information loss and makes the network degenerate. He et al. (2016) proposed the concept of a deep residual network to reduce the effect of redundant layers on network degradation. It divides the network into two parallel parts. One part keeps the original network design. The other is designed as a bypass connection, which is an identity mapping corresponding to the starting layer of the original network, and the mapping is added across multiple hidden layers to the output layer of the original network. The two act together as the input to the next layer. In the deep residual network, \mathbf{x} is the input, $F(\mathbf{x})$ is the output after the original network, and let the final output of the network be $H(\mathbf{x})$. Then $H(\mathbf{x}) = F(\mathbf{x}) + \mathbf{x}$. If $F(\mathbf{x})$ acts on a redundant layer, which means it lacks a positive effect on the output. Then $H(\mathbf{x})$, in the presence of \mathbf{x} , can guarantee that the output results are at least consistent with the results of directly on \mathbf{x} , namely $H(\mathbf{x}) \approx \mathbf{x}$. Calling $F(\mathbf{x}) = H(\mathbf{x}) - \mathbf{x}$ the residual term, then $F(\mathbf{x}) \approx 0$. Since the initialized parameter is generally around zero, learning $F(\mathbf{x}) \approx 0$ is simpler than directly learning $H(\mathbf{x}) \approx \mathbf{x}$ when parameters are updated. Moreover, the residual model can avoid the gradient disappearance that occurs when the gradient is backpropagated. In backpropagation, ε represents the loss equation, which is obtained by the chain rule

$$\frac{\partial \varepsilon}{\partial \mathbf{x}} = \frac{\partial \varepsilon}{\partial H(\mathbf{x})} \cdot \frac{\partial H(\mathbf{x})}{\partial \mathbf{x}} = \frac{\partial \varepsilon}{\partial H(\mathbf{x})} \left[1 + \frac{\partial F(\mathbf{x})}{\partial \mathbf{x}} \right]. \quad (\text{S1})$$

Even if $\frac{\partial F(\mathbf{x})}{\partial \mathbf{x}} = 0$, the lossless propagation term still exists in the gradient, with no gradient disappearance.

References

- Bengio Y, Courville A, Vincent P, 2013. Representation learning: a review and new perspectives. *IEEE Trans Patt Anal Mach Intell*, 35(8):1798-1828. <https://doi.org/10.1109/TPAMI.2013.50>
- He KM, Zhang XY, Ren SQ, et al., 2016. Deep residual learning for image recognition. Proc IEEE Conf on Computer Vision Pattern Recognition, p.770-778. <https://doi.org/10.1109/CVPR.2016.90>
- Hinton GE, Salakhutdinov RR, 2006. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504-507. <https://doi.org/10.1126/science.1127647>
- Liu XY, Yang DY, Gamal AE, 2017. Deep neural network architectures for modulation classification. Proc Asilomar Conf on Signals Systems and Computers. <https://arXiv.org/abs/1712.00443v3>
- Wang Y, Liu M, Yang J, et al., 2019. Data-driven deep learning for automatic modulation recognition in cognitive radios. *IEEE Trans Veh Technol*, 68(4):4074-4077. <https://doi.org/10.1109/tvt.2019.2900460>
- Zitouni R, George L, 2016. Output power analysis of a software defined radio device. IEEE Radio and Antenna Days of the Indian Ocean, p.1-2. <https://doi.org/10.1109/RADIO.2016.7771996>