END-TO-END LEARNING OF COMMUNICATIONS SYSTEMS

A. traditional schemes

B. The autoencoder concept

C. Autoencoder for multiple users

D. RTN

E. several common assessment

C. 和D.任选一个写，感觉D比较好实现

END-TO-END LEARNING OF COMMUNICATIONS SYSTEMS

In this section, we simulate a communication system using only deep neural networks. The main idea is to design a communication system by interpreting it as an autoencoder trained by SGD. This section would firstly reproduce some results of [1] ([1] Timothy J. O’Shea and Jakob Hoydis. An introduction to deep learning for the physical layer… progress report上有错. CoRR, abs/1702.00832, 2017.), then explain those results carefully and present several novel simulations about that idea.

1. Autoencoders for the Physical Layer

In a communications system, the simplest form consists three part: a transmitter, a receiver, and a channel. The messages sent by the transmitter are reconstructed by the receiver over a physical channel. The whole process of communication which reconstructs the messages is shown in Fig.1.

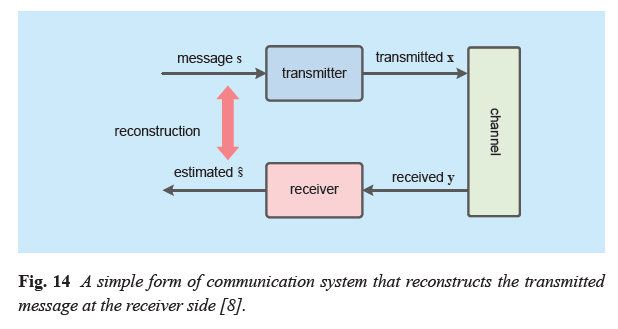


Fig. 1

The transmitted message is one out of possible messages, and the transmitter makes discrete uses of the channel. At the transmitter side, it applies the transformation . The transformation maps the message to the transmitted signal . Generally, the transmitted signal should be robust enough to be against the channel with noise. And should be imposed certain constraints due to the transmitter’s hardware. Usually, there are three common constraints: an energy constraint that demands , an amplitude constraint demands , and an average power constraint demands .

We apply the notation of [1]: the communication rate of the system is (bit/channel use), . And means that the transmitter sends one out of messages via channel uses. In addition, is the received -demensional signal noised by a channel. To describe the channel, we use a conditional probability density function to represent the channel. At the receiver side, the receiver applies the transformation to reconstruct the transmitted message as .

Therefore, we could consider the communication as a process of end-to-end reconstruction problem. The transmitter sends messages and the receiver reconstructs the messages over a physical channel. Based on DL theory, this reconstruction process could be considered as a particular type of *autoencoder*. This autoencoder could achieve global optimization for the transmitter and receiver over a physical channel. In this way, we apply fully connected dense layers to represent the transmitter and the receiver. To represent the AWGN channel between the transmitter and the receiver, we would apply a noise layer with a certain variance.

In this case, the purpose of applying the concept of autoencoder for communication system is to jointly optimize the transmitter and the receiver over a physical channel. At the transmitter side, the autoencoder seeks to learn how to represents the messages by applying the transformation . The representations should be robust with respect to the impairments of the physical channel. In this way, the transmitted messages could be reconstructed with small probability error due to the noise, fading, and distortion of the channel. At the receiver side, the optimization of estimation of the original messages can be done by training neural networks using SGD.

There is an example of autoencoder for an end-to-end communication system in Fig. 2. The transmitter firstly uses an -dimensional one-hot vector to represent . Then the -dimensional one-hot vector would be fed into multiple dense layers followed by a normalization layer. The normalization layer ensures that physical constraints on the generated -dimensional vector are met. We use an additive noise layer to represent the physical channel. The variance of the noise layer is a fixed value . denotes the energy per bit and denotes the noise power spectral density. We also apply several dense layers to the receiver. As same as the most part of deep learning for classification, at the receiver side, its last layer applies a softmax activation to generate the decoded messages . The output of the dense layer with softmax activation is an -dimensional probability vector. The sum of all elements of is equal to . Then the value of the element of with highest probability corresponds to the estimation . The autoencoder-based communication system could learn a joint optimization of coding and modulation scheme.

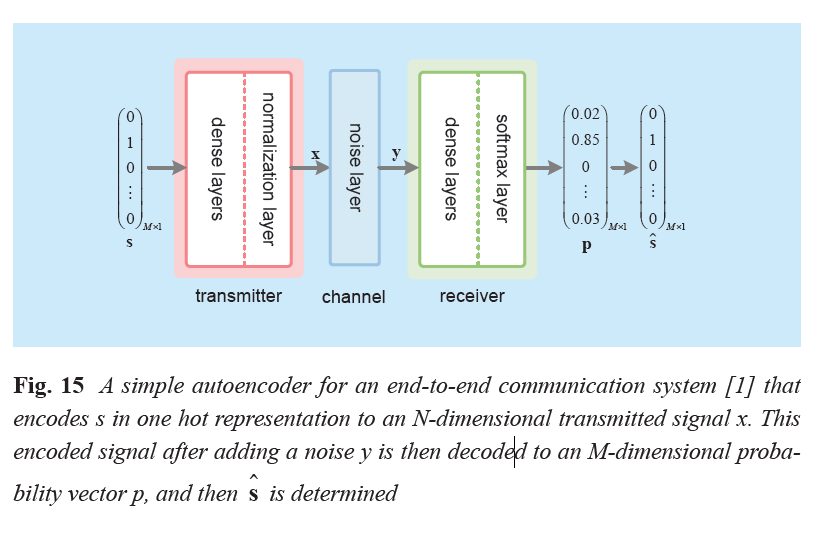


Fig. 2

1) *Autoencoder Simulation Algorithm:*

With the explanations of the autoencoder, we present the corresponding pseudo-code in Algortithm1.

**Algorithm 1:** Autoencoder

GenerateBatch of one-hot vectors:

Define autoencoder and its layer:

|  |  |
| --- | --- |
| Layer | Output dimentions |
| Input |  |
| Multiple dense layers + ReLU |  |
| Dense layer + linear |  |
| Normalization |  |
| Noise |  |
| Multiple dense layers + ReLU |  |
| Dense layer + softmax |  |

Train the autoencoder:

Feed the autoencoder with

Features = ; Labels =

/\****Note:*** The variance is a fixed value: 7 dB.

Training is done using Adam [27] with learning rate 0.001. \*/

Plot learned constellations

/\****Note:*** If the constellation is for higher dimension like (7,4) autoencoder, we should use a two-dimensional t-distributed stochastic neighbour embedding (t-SNE) [28] to depict the higher dimensional signal representations. \*/

Calculate the block error rate (BLER) versus different value. And plot the BLER curve.

[27] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, 2015, pp. 1–15.

[48] L. V. D. Maaten and G. Hinton, “Visualizing data using t-SNE,” *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.