Deep Learning for Channel Coding via Neural Mutual Information Estimation

PARTH DODHIA | SHAAN UL HAQUE

180070041 | *180070053*

COURSE: CS419(M)

INSTRUCTOR: PROF. ABIR DE

Abstract - End-to-end deep learning for communication systems, i.e., the complete process of training the embeddings for the messages, dense NN to convert the embeddings into signal constellation and training the decoder to decode the received corrupted signal is carried out at one go. These encoder-decoder systems can achieve bit error rates which come close to practical baseline techniques if they are used for over-the-air transmissions. This is promising since complex encoding and decoding functions can be learned on the fly without extensive communication-theoretic analysis and design, possibly enabling future communication systems to better cope with new and changing channel scenarios.

1 Introduction

1.1 Some Theory

One of the biggest achievements in the field of science was communication over large distances. Sitting in one corner of the world we can talk to other person who might be sitting on the opposite side of the world without significant delay. But how is this achieved?? Below, we show a high level generic communication model.

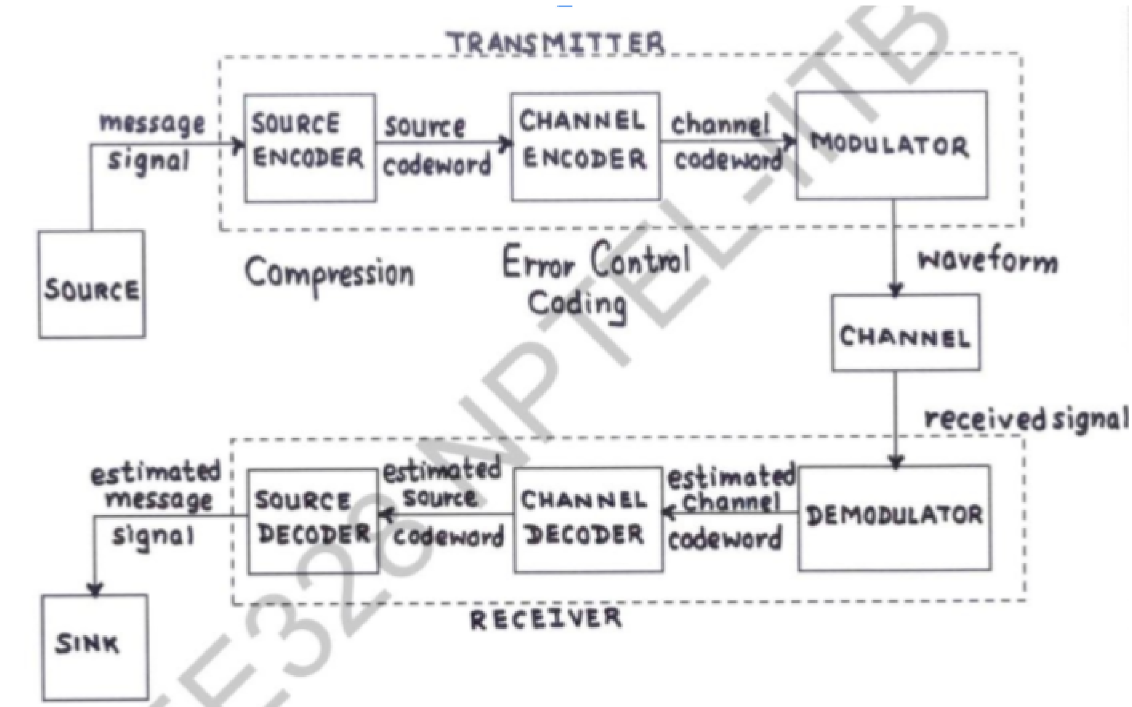

Figure 1: Generic communication model[2].

We need not go into the details of the above model. The part which of concern to us is the channel encoder. We would assume that the source encoder has encoded the original message signal efficiently so as to produce most compressed form of data which is uniformly distributed. When the waveform from the modulator is passed through the channel to the receiver, it gets corrupted by noise distorting the waveform, perhaps sometimes so much that the receiver might decode it incorrectly. In order to minimize this effect we want to encode the message signal and add some redundancy so as to make it robust against noise. By adding this redundancy, we want that the message signals input to the channel be, in some sense 'far apart' so that there is no conflation in the received signals because of noise.

1.2 Literature Review

The generic communication channel model has a encoder - noise source - decoder structure, very similar to that of autoencoders used in machine learning. Building on this idea, the use of autoencoders has been proposed [3], but they have the drawback of requiring the **channel noise probability distribution** for training, and also a **differentiable channel model for backpropagation**.
Another approach is to use generative adverserial networks (GANs) [4] to produce an artificial channel model for approximating the true distribution, and then using it for end-to-end training [5][6]. This approach is very popular, but requires receiver design as well for end-to-end training. Reinforcement Learning has also been used to replace the differntiability constraint of backpropagation with a feedback link - the transmitter is treated as the agent, signal transmission being the action for which it receives reward via the feedback link [7]. One drawback of this model is the fact that RL algorithms require large amount of data to achieve good performance.
The approach we propose to implement is motivated by mutual information which is used to measure the capacity of a communication channel [1]. We implement a Mutual Information Neural Estimator to train the encoder independent of the decoder, thus solving the two key challenges of the techniques mentioned above and show that it achieves results as good as classical methods on Additive White Gaussian Noise Channels.

1.3 Basic Idea

Because of this redundancy, there is an upper limit to the bit transmission rate. From a communication theoretic perspective, the optimal transmission rate is a function of the mutual information **I(X;Y)** between input X(encoded waveform input to the channel) and output Y of a channel p(y|x). This optimal transmission rate is called as **Capacity** of the channel. For an Additive White Gaussian Noise(AWGN) channel the optimal transmission rate is given by:

$$\boldsymbol{C} = \underset{p(x):E[X^2]\leq P}{\arg\max}\ H(X) - H(X|Y) = \underset{p(x):E[X^2]\leq P}{\arg\max}\ \mathbf{I(X;Y)} = log(1+\frac{P}{\sigma^2})$$

Notice, we are introducing a constraint on the expected value of the random variable X^2 which is actually a **power constraint** on signal. Thus, mutual information is a good metric to evaluate the performance of our encoding scheme. However, the mutual information also dependents on the channel probability distribution. But instead of approximating the channel probability distribution itself, we will approximate the mutual information between the samples of the channel input and output and optimize the encoder weights, by maximizing the mutual information between them.

1.4 Channel Model

We consider a communication model with a transmitter, a channel, and a receiver. The transmitter wants to send a message $m \in \mathcal{M} = \{1, 2, ...M\}$ over a noisy channel using an encoding function $f(m) = x^n(m) \in C^n$ (n=1 in our case) to make the transmission robust against noise. Moreover, for every message $m \in \mathcal{M}$ assume an average power constraint $\frac{1}{n}\sum_{i=1}^{n} x_i^2(m) \leq P$ on the corresponding codewords. In this work, our communication channel is an AWGN channel such that the received signal is given as:

$$Y_i = X_i + Z_i$$

where Z_i is iid with $Z_i \sim C\mathcal{N}(0, \sigma^2)$. The probability of error is defined as:

$$P_e = \frac{1}{|\mathcal{M}|}\sum_{m=1}^{M} Pr(\hat{M} \neq m | M = m)$$

where $\hat{M}$ is the decoded signal.
The complete model of encoder decoder pair is shown below. The complete description of the model will follow in the next Experiment and Analysis section.

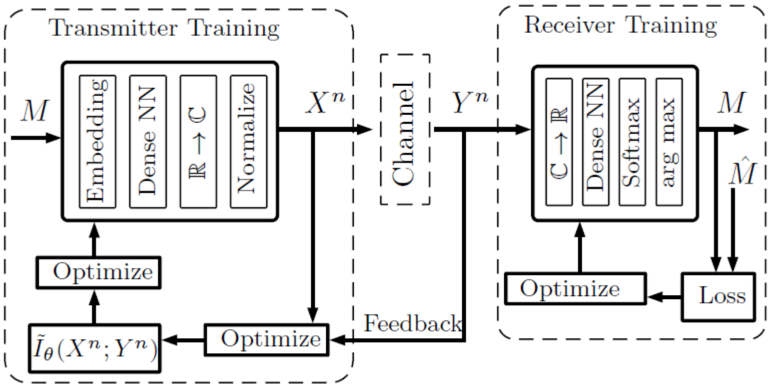

Figure 2: Complete Auto Encoder Model [1]

1.5 Neural Estimation of Mutual Information

The mutual information between two random variables X and Y is calculated as :

$$I(X;Y) = H(X) - H(X|Y)$$

It can be interpreted as the reduction in entropy or uncertainty of X upon receiving Y. For a communication channel with message signal X as input, we would like this reduction to be large to be able to estimate X from Y. The expression for $I(X;Y)$ can be simplified to:

$$I(X;Y) = \int_{x,y} p\,(x,y)\, log \left(\frac{p\,(x,y)}{p\,(x) p\,(y)} \right) dx\, dy$$

This equal to the KL divergence between the distributions $p(x, y)$ and $p(x)\, p(y)$
Since both the joint and marginal distributions are unknown, we choose an alternate method to evaluate $I(X;Y)$. Since

$$D_{KL}(P||Q) = \sup_{g} \mathbb{E}_P\,[g(X,Y)] - \log\, \left(\mathbb{E}_Q\,[exp(g(X,Y)]\right)$$

for a neural network T_θ with inputs X and Y, representing a subset of the possible choices for $g(X,Y)$,

$$D_{KL}(P||Q) \geq \sup_{\theta} \mathbb{E}_P \left[T_\theta(X,Y)\right] - \log\, \left(\mathbb{E}_Q \left[exp(T_\theta(X,Y)\right]\right)$$

T_θ is also commonly known as the Mutual Information Neural Estimator (MINE) in this setup. Thus, we can train the MINE network and use it to train the encoder instead of approximating the channel distribution. The neural network for the MINE estimator is shown below:

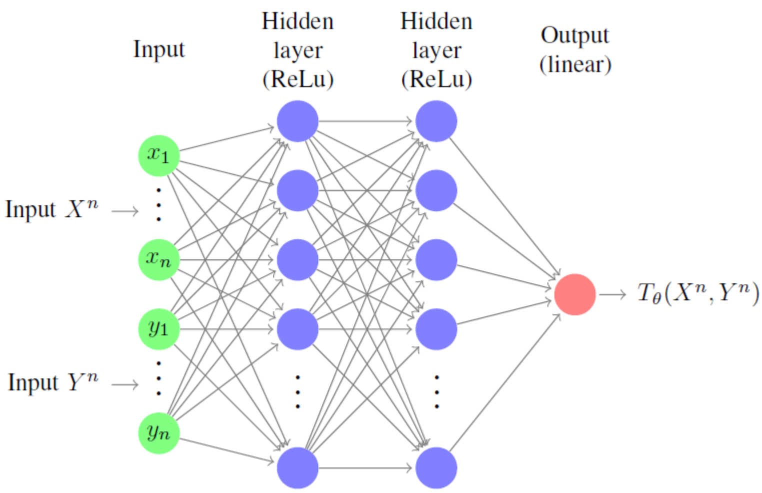

Figure 3: Neural Network to estimate Mutual Information[1]

2 Training Methodology

Our neural network setup is shown below:

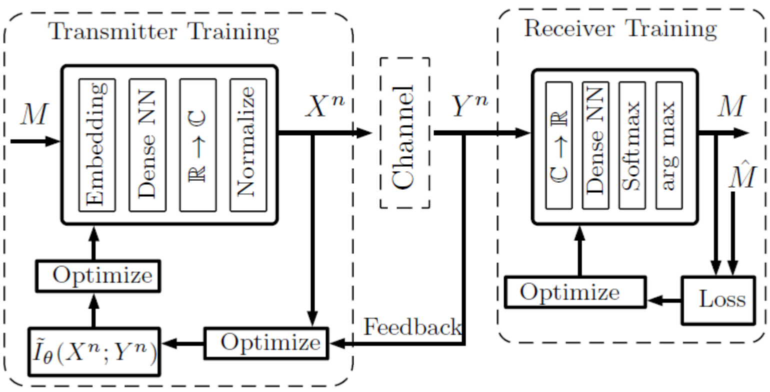

Figure 4: Complete Auto Encoder Model[1]

On the left hand side we have the encoder-MINE network, followed by the channel and the receiver network. We took the number of messages(M)=16 and for training the SNR(signal to noise ratio) was taken to be 7dB. Other details such as number of epochs, batch size, etc. are not mentioned here so as to make the report short and relevant. We first train the MINE and encoder networks **alternatively** using the following data: A batch of message signals X and channel outputs Y is collected and passed through the MINE network with the objective of approximating the empirical mutual information:

$$\hat{I}(X;Y) = \frac{1}{k}\sum_{i=1}^{k} T_\theta(x_i, y_i) - \log\left[\frac{1}{k}\sum_{i,j\in[k]:i\neq j} exp(T_\theta(x_i, y_j)\right]$$

Then, X, Y are passed through the encoder for maximising $\hat{I}(X;Y)$. This alternative process is repeated till we get a good encoder which can be verified by looking at its constellation. The following objective function needs to maximized.

$$\arg\max_\phi \arg\max_\theta \hat{I}_\theta((X_\phi^n(m); Y))$$

Finally, the encoder is fixed and the decoder is trained with the aim of minimising probability of symbol error. Since decoder needs to classify the received signal the loss function is sparse categorical loss. The model parameters are listed in the table below.

Parameters	Encoder	MINE	Decoder
Number of layers	4	3	2
Activation/Name	Embedding, Elu, None, Norm Layer	ReLU, ReLU, None	Elu, Softmax
Number of nodes	M(=16), M, 2	256, 256, 1	M, M

3 Analysis of Experimental Results

We can get a good idea about the encoder performance by looking at the signal constellation. For binary modulation schemes, we can transmit two independent real values (using two basis signals) for each message, which are then perturbed by channel noise before reception.
The decoder in the receiver decides a decision region for each message in the 2-dimensional plane (with components corresponding to the amplitude of received basis signals) for classifying each message as one of the (here) 16 possible inputs.
Thus an ideal encoder would 'place' each message signal in the 2-d plane (by assigning 2 real values to it) so as to satisfy the energy constraint and optimise the maximum a-posteriori estimate, which corresponds to distance between messages for a Additive White Gaussian channel. The obtained constellation of encoder outputs is shown below:

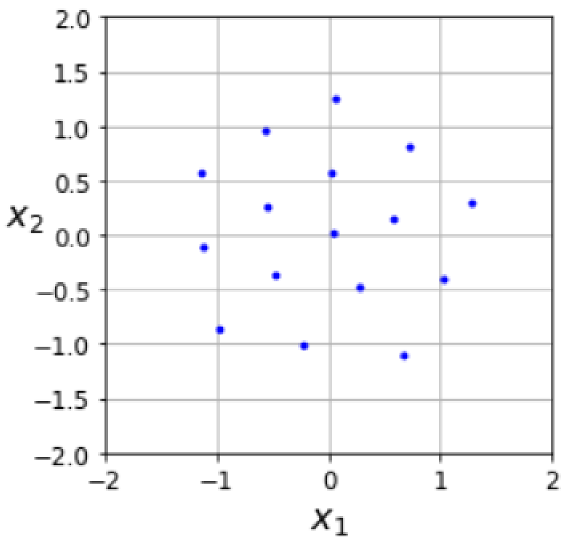

Figure 5: Learnt Signal Constellation.

We can clearly see the way messages are placed symmetrically to simultaneously satisfy the energy constraint and keep large distance between signals.
After training the encoder, we train the decoder using cross entropy loss to minimize probability of symbol error. The plot of average symbol error as a function of SNR for the MINE-Cross entropy (MI CE) system and the classical 16-QAM system is shown below:

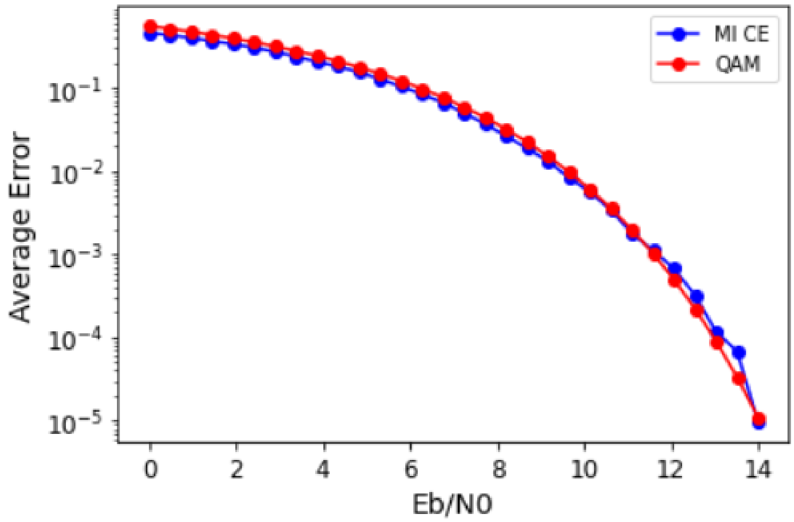

Figure 6: Average Error vs SNR compared to QAM.

We can see that the MI CE system performs just as good as 16 QAM. The huge implication of such a result is that while 16 QAM is designed after rigorous mathematical analysis and optimisation of the MAP estimate for Gaussian Noise, the MI CE estimator is simply trained by sampling the channel outputs. So, if the noise statistics were to change, the optimal signal constellation for the encoder output could change completely and would have to be determined by analysis. On the other hand, we can simply retrain the MI CE system using new samples.

4 Conclusion

In any conventional modulation technique, knowledge of the channel distribution is essential. Along with that we need to derive the average probability of symbol error for the designed scheme by going into the gory mathematical analysis.
We observe that in our method without going into the complex mathematical analysis or knowing the channel distribution, channel encoding can be done efficiently with error rate similar (rather slightly less for low SNR) to the conventional encoding schemes. Literature review suggests that not only for M=16 or for lower values of M but this scheme works for large number of message signals also with similar results. But one of the major drawback of this approach is large number of sample size for accurate mean estimation. This assumption may not be true always and it remains an open question on how to tackle situations where the number of samples are also low.

5 Usage of Code

- The code does not need installation of any additional packages. It can be easily run on Google Colab or a machine with latest version of Tensorflow installed on it.

6 References

[1] R. Fritschek, R. F. Schaefer and G. Wunder, "Deep Learning for Channel Coding via Neural Mutual Information Estimation," 2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), 2019, pp. 1-5

[2] EE328 Digital Communication NPTEL series.

[3] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," IEEE Transactions on Cognitive Communications and Networking, vol. 3, no. 4, pp. 563–575, Dec 2017

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Advances in neural information processing systems, 2014, pp. 2672–2680.

[5] H. Ye, G. Y. Li, B.-H. F. Juang, and K. Sivanesan, "Channel agnostic end-to-end learning based communication systems with conditional GAN," arXiv preprint arXiv:1807.00447, 2018

[6] T. J. O'Shea, T. Roy, N. West, and B. C. Hilburn, "Physical layer communications system design over-the-air using adversarial networks," arXiv preprint arXiv:1803.03145, 2018.

[7] F. A. Aoudia and J. Hoydis, "End-to-end learning of communications systems without a channel model," arXiv preprint arXiv:1804.02276, 2018.