Autoencoder Based Communication Systems

"Leveraging Autoencoder-Based Architectures, including Variational Autoencoders, for Optimized Learning in the Convergence."

Rehaan Goenka
Department of Electronics Engineering
Indian Institute of Technology, Varanasi
Varanasi, India
rehaan.goenka.ece23@iitbhu.ac.in

Shreyansh Bajpai
Department of Electronics Engineering
Indian Institute of Technology, Varanasi
Varanasi, India
shreyansh.bajpai.cd.ece23@iitbhu.ac.in

Harshit Verma
Department of Electronics Engineering
Indian Institute of Technology, Varanasi
Varanasi, India
harshit.verma.ece23@iitbhu.ac.in

Aynikal Noha Riju
Department of Electronics Engineering
Indian Institute of Technology, Varanasi
Varanasi, India
aynikal.nohariju.ece23@iitbhu.ac.in

***Abstract*—Modern wireless communication systems are traditionally designed as a cascade of independently optimized signal processing blocks — including source encoding, channel coding, modulation, and equalization — each relying on analytically tractable models and simplifying assumptions. While effective under idealized conditions, this compartmentalized approach often leads to suboptimal performance in real-world environments, where channels exhibit complex, nonlinear, and time-varying characteristics that are difficult to model precisely.**

To address these limitations, autoencoder-based architectures have emerged as a transformative paradigm, enabling end-to-end learning of the physical layer using deep neural networks. By jointly optimizing both the transmitter and receiver in a single learning framework, autoencoders eliminate the need for explicit mathematical modeling of the channel, allowing the system to learn directly from data and adapt to real-world impairments such as fading, hardware non-idealities, and interference.

This data-driven approach not only learns robust modulation and coding schemes tailored to specific channel conditions, but also often surpasses traditional designs in terms of bit error rate, spectral efficiency, and energy efficiency. Furthermore, autoencoders offer intrinsic flexibility, enabling seamless adaptation to novel or dynamic environments, and unlock the potential for hardware-friendly, low-complexity inference, making them a compelling solution for next-generation intelligent communication systems.

On building upon autoencoders, we further extend the framework to Variational Autoencoders (VAEs), which introduce a probabilistic approach to latent space modeling. While traditional autoencoders are effective in learning compressed representations of data, they lack a structured latent space and the ability to generate diverse outputs from learned distributions. VAEs address these limitations by enforcing a prior distribution—typically a Gaussian—on the latent variables and using variational inference to approximate the posterior. This allows for both generative capabilities and improved generalization, making VAEs more robust in applications such as data synthesis, representation learning, and semi-supervised training.

I. Introduction

Wireless systems are on the cusp of a revolution. Gone are the days when transmitters and receivers were painstakingly engineered as a chain of isolated blocks—each optimized in a vacuum for a narrow set of channel assumptions. Today's real-world environments teem with nonlinear hardware quirks, unpredictable interference, and fading profiles that defy neat mathematical models. In this untamed landscape, classical designs are simply reaching their limits.

Enter deep autoencoders. In their groundbreaking 2017 work, O'Shea and Hoydis showed that we can rethink the entire physical layer as a single end-to-end learning problem: the transmitter, the channel, the receiver—united into one neural network that learns to communicate by minimizing reconstruction error. Remarkably, this learned "channel autoencoder" not only matched Hamming codes and sphere-packing bounds in simulated Additive White Gaussian Noise channels but also discovered entirely novel constellations tailored to complex propagation scenarios.

Building on these advances, our journey unfolds in two phases:

- **Reproducing the Channel Autoencoder.** We began by faithfully implementing O'Shea's seminal architecture, confirming its ability to learn robust modulation and coding schemes without any expert feature engineering.
- **Extending to Variational Autoencoders (VAEs).** To capture uncertainty and explore richer latent symbol representations, we augmented the autoencoder with a probabilistic bottleneck—dramatically improving generalization across unseen channel conditions and paving the way for flexible, data-efficient training.

This paper weaves together the power of deep learning and the dual imperatives of tomorrow's wireless networks. From pure communication prowess to environmental intelligence, our autoencoder-driven framework delivers a leap forward—transforming how radios think, adapt, and perceive the world.

II. Practical Applications

Autoencoders (AEs) and Variational Autoencoders (VAEs) are widely utilized in diverse domains due to their capability to learn efficient, compact representations of data. Below, we outline three key applications of each, with emphasis on their practical relevance.

A. Autoencoders (AEs)

1) **Denoising of Signals and Images:** Autoencoders are effectively used for denoising tasks by training the network to reconstruct clean data from corrupted inputs. This has proven valuable in medical imaging, audio signal enhancement, and preprocessing of satellite data.
2) **Anomaly Detection:** When trained on normal data patterns, autoencoders can detect anomalies by observing large reconstruction errors. This makes them suitable for applications in industrial fault monitoring, network security (e.g., intrusion detection), and health diagnostics (e.g., identifying irregular ECG signals).

B. Variational Autoencoders (VAEs)

1) **Data Generation and Synthesis:** VAEs model the underlying data distribution, enabling the generation of new, diverse samples. Applications include synthetic image generation, molecule design in computational chemistry, and text synthesis in NLP.
2) **Semi-Supervised Learning:** By incorporating both supervised and unsupervised objectives, VAEs can utilize labeled and unlabeled data effectively. This is particularly beneficial in scenarios with limited labeled data, such as rare disease classification or low-resource language processing.

III. Autoencoders for End-to-End Communication Systems

A traditional communication system consists of three primary components: a transmitter, a channel, and a receiver. The transmitter aims to communicate one of $M = 2^k$ possible messages to the receiver over n discrete channel uses. This is achieved by mapping an input message $s \in \mathcal{M}$ to a real-valued vector $\mathbf{x} = f(s) \in \mathbb{R}^n$ via an encoding function f. The transmitted signal $\mathbf{x}$ must satisfy practical constraints such as energy limitations ($\|\mathbf{x}\|_2^2 \leq n$), amplitude bounds ($|x_i| \leq 1$), or average power constraints ($\mathbb{E}[x_i^2] \leq 1$). The transmission rate is then given by $R = \frac{k}{n}$ bits per channel use.

The channel is modeled by a conditional probability distribution $p(\mathbf{y}|\mathbf{x})$, which describes how the transmitted signal $\mathbf{x}$ is distorted by impairments like noise and fading to produce the received signal $\mathbf{y} \in \mathbb{R}^n$. The receiver uses a decoding function $g : \mathbb{R}^n \to \mathcal{M}$ to produce an estimate $\hat{s}$ of the original message s.

This classical system can be reformulated using the concept of autoencoders from deep learning. Originally developed for unsupervised learning tasks such as data compression and feature extraction, autoencoders consist of an encoder that compresses input data into a latent space and a decoder that reconstructs the original data. In the communication context, the autoencoder's objective is not to compress but to enable robust transmission over noisy channels. The transmitter and receiver are jointly trained to optimize end-to-end message reconstruction accuracy despite channel impairments.

In this autoencoder-based framework, the transmitter and receiver are implemented as neural networks. The input message s is represented as a one-hot encoded vector $\mathbf{1}_s \in \mathbb{R}^M$. This vector is passed through the transmitter network, which consists of multiple dense layers followed by a normalization layer that ensures compliance with physical transmission constraints. The output of the transmitter is the signal $\mathbf{x}$, which is then passed through a channel layer.

The channel layer is modeled using additive white Gaussian noise (AWGN) with variance $\sigma^2 = \left(2R \cdot \frac{E_b}{N_0}\right)^{-1}$, where $\frac{E_b}{N_0}$ is the energy per bit to noise power spectral density ratio. The receiver neural network processes the noisy received signal $\mathbf{y}$ through a series of dense layers, culminating in a softmax layer that outputs a probability distribution over the M possible messages. The predicted message $\hat{s}$ corresponds to the index with the highest probability. The model is trained using the categorical cross-entropy loss between the true one-hot encoded message and the predicted distribution, optimized via stochastic gradient descent or adaptive optimizers like Adam.

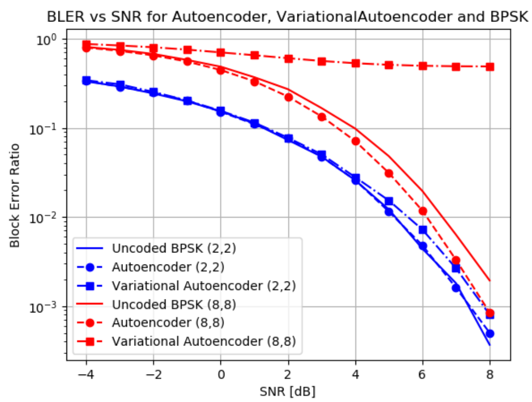
Experimental results demonstrate that the autoencoder can learn effective end-to-end communication strategies that rival conventional systems. For example, when compared to a binary phase-shift keying (BPSK) system with a Hamming (7,4) error-correcting code under either hard-decision or maximum likelihood decoding, the autoencoder achieves comparable or superior block error rate (BLER) performance. Importantly, the autoencoder achieves this without explicit knowledge of traditional modulation or coding schemes, instead learning these jointly from data. Additionally, deeper transmitter architectures facilitate better convergence and generalization due to the richer parameter space, helping the model escape suboptimal local minima.

Training is typically performed at a fixed E_b/N_0 value (e.g., 7 dB). Larger batch sizes during training tend to improve stability and final performance. The number of trainable parameters varies depending on the input-output dimensions and network depth, ranging from tens of thousands to over a hundred thousand parameters.

The learned modulation schemes provide further insight into the model's behavior. In small-scale systems like (2,2) or (4,2), the autoencoder often discovers conventional constellations such as QPSK or 16-PSK, albeit potentially with geometric rotations. Changing normalization strategies or constraints can yield irregular but robust constellations, such as hybrid pentagonal-hexagonal layouts, while maintaining low BLER. Dimensionality reduction techniques such as t-SNE applied to the receiver's input space reveal clear clustering of received signal points, highlighting the autoencoder's ability to learn discriminative features that are resilient to channel noise.

For larger message sets, one-hot encoding becomes inefficient. A more scalable alternative is to represent messages using binary vectors of dimension $\log_2 M$, using sigmoid activations and losses such as binary cross-entropy or mean squared error. However, this introduces challenges in training complexity and memory requirements, particularly for very large M.

A key advantage of the autoencoder paradigm is its adaptability to arbitrary channels. It can learn robust encoding and decoding strategies even when the channel model is unknown or analytically intractable. This makes deep learning-based autoencoders a powerful and flexible tool for the design of future communication systems.



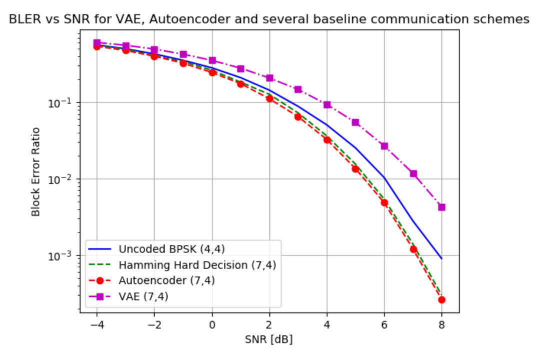


Fig. 1. Top: Autoencoder architecture. Bottom: VAE structure.

IV. Variational Autoencoders (VAE)

A. General Overview

A Variational Autoencoder (VAE) is a generative model that utilizes probabilistic graphical models with latent variables. It consists of two components: the encoder, which approximates the posterior distribution over latent variables, and the decoder, which reconstructs the data from these variables. The VAE learns a low-dimensional representation of data and generates samples through this representation.

The model treats both data and latent variables probabilistically using conditional distributions. The joint distribution is factorized as:

$$p(x, z) = p(z)p(x|z),$$

where $p(z)$ is the prior over latent variables, and $p(x|z)$ is the decoder. The encoder approximates the posterior $p(z|x)$ with a variational distribution $q(z|x)$.

The VAE optimizes the Evidence Lower Bound (ELBO), which minimizes the divergence between the approximate and true data distributions. This allows the decoder to generate realistic data samples. For tasks like classification, VAEs model conditional distributions $p(y|x)$.

VAEs can be extended to Deep Latent Variable Models (DLVMs), where neural networks parameterize the distributions, enabling the model to approximate complex data distributions.

In summary, VAEs combine probabilistic modeling and neural networks to learn and generate data by mapping observed data into a latent space and reconstructing it.

B. Underlying Mathematical Operations and Loss Functions

The Additive White Gaussian Noise (AWGN)channel is a widely used model to represent signal corruption in communication systems. For a signal with z dimensions, Gaussian corruption with noise power σ_n^2 per component is modeled as:

$$\hat{z} = z + n, \quad n \sim \mathcal{N}(0_m, \sigma_n^2 I_m),$$

where 0_m is a zero vector of dimension m, and I_m is the identity matrix of dimension $m \times m$.

Given a Gaussian prior $p(\hat{z}) = \mathcal{N}(0_m, \sigma_0^2 I_m)$, the Kullback-Leibler (KL) divergence for the AWGN channel can be computed as:

$$D_{\mathrm{KL}}(q_\phi(\hat{z}|x) \| p(\hat{z})) = \frac{1}{2\sigma_0^2} \sum_{j=1}^{m} \hat{z}_j^2 - \frac{m}{2}\left(1 - \frac{\sigma_n^2}{\sigma_0^2}\right) + \log\left(\frac{\sigma_n^2}{\sigma_0^2}\right).$$

Depending on the symbol representation (e.g., one-hot or binary), this expression can be combined with the appropriate objective function to train the model in the AWGN channel.

For one-hot encoding as seen in previous works, the Evidence Lower Bound (ELBO) objective to maximize can be computed as:

$$\mathbb{E}_{x \in X}\left[\log p(x) - \frac{1}{2\sigma_0^2} \sum_{j=1}^{m} \hat{z}_j^2 + \frac{m}{2}\left(1 - \frac{\sigma_n^2}{\sigma_0^2}\right) + \log\left(\frac{\sigma_n^2}{\sigma_0^2}\right)\right].$$

Since the noise power per component σ_n^2 and the prior variance σ_0^2 are constants, the final objective to maximize simplifies to:

$$\max_{\theta_T, \theta_R} \mathbb{E}_{x \in X}\left[\log p(x) - \frac{1}{2\sigma_0^2} \sum_{j=1}^{m} \hat{z}_j^2\right].$$

The first term in this objective represents the negative of the categorical cross-entropy. Previous works focused on optimizing this term under a constant training Signal-to-Noise Ratio (SNR). The second term relates to the power used for signaling, and its optimization encourages a tradeoff between minimizing transmit power and maximizing reconstruction likelihood. If training is done under constant SNR, the second term becomes a constant, recovering the objective used in earlier works.

In Autoencoder (AE)-based communication systems, the typical objective is to maximize reconstruction likelihood while controlling the signal power via a normalization layer. The optimization objective becomes:

$$\max_{\theta_T,\theta_R} \mathbb{E}_{p(x)}\mathbb{E}_{q(\hat{z}|x)}\left[\log p_{\theta_R}(x|\hat{z})\right]$$

subject to

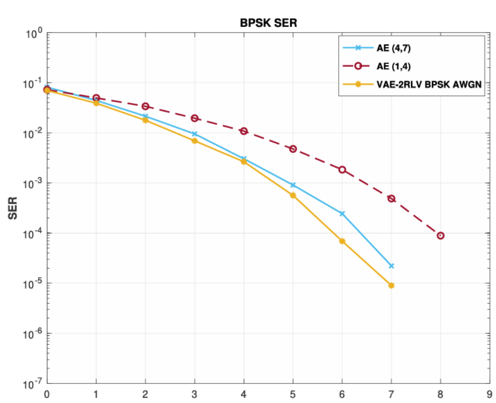
$$\mathbb{E}_{p(x)}\hat{z}^T\hat{z} = m\sigma_n^2\gamma,$$

where γ is the training SNR, and λ_L is a Lagrange multiplier. The optimization problem can be rewritten as:

$$\max_{\theta_T,\theta_R} \mathbb{E}_{p(x)}\mathbb{E}_{q(\hat{z}|x)}\left[\log p_{\theta_R}(x|\hat{z})\right] - \lambda_L\mathbb{E}_{p(x)}\hat{z}^T\hat{z}.$$

Comparing this with the objective derived earlier, we observe that AE-based models use a normalization layer to control the power used for signaling. These models optimize reconstruction likelihood while imposing a hard constraint on signal power. The approach in this work, however, introduces a soft constraint instead of a hard one, allowing for easier optimization.

Recent research suggests that hard constraints may not lead to the desired performance in deep learning models. Soft constraints, like the ones used here, provide differentiable penalties, improving optimization and computational efficiency. In contrast, previous methods with hard constraints (e.g., normalization layers) have been shown to result in more complex computations with limited practical benefits.



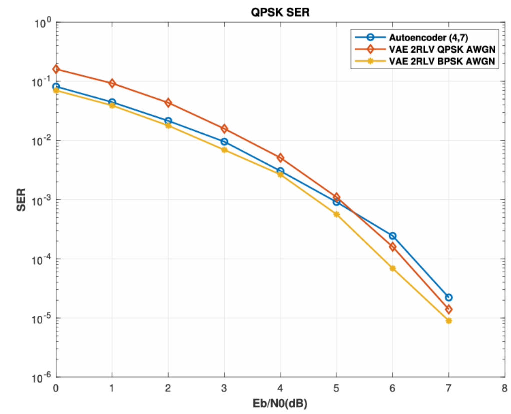


Fig. 2. Top: SER performance of the BPSK VAE vs AE baseline schemes. Bottom: SER performance BPSK and QPSK of the VAE scheme under AWGN vs AE baseline schemes

References

[1] T. O'Shea, "An introduction to deep learning for the physical layer," *arXiv preprint arXiv:1702.00832*, 2017.

[2] T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2002.

[3] R. M. Gagliardi and S. Karp, *Optical Communications*, 2nd ed. Hoboken, NJ, USA: Wiley, 1995.

[4] H. Meyr, M. Moeneclaey, and S. A. Fechtel, *Digital Communication Receivers: Synchronization, Channel Estimation, and Signal Processing*. Hoboken, NJ, USA: Wiley, 1998.

[5] E. Bourtsoulatze, D. B. Kurka, and D. Gündüz, "Deep joint source-channel coding for wireless image transmission," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 3, pp. 567–579, Sep. 2019.

[6] E. Nachmani, Y. Be'ery, and D. Burshtein, "Learning to decode linear codes using deep learning," in *Proc. 54th Annu. Allerton Conf. Commun., Control, Comput.*, Sep. 2016, pp. 341–346.

[7] E. Nachmani *et al.*, "Deep learning methods for improved decoding of linear codes," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 119–131, Feb. 2018.