



Fig. 1: The architecture of the implemented AE-NOMA for two-users. Each FCNN[ℓ] represents a fully connected layer with ℓ nodes. The main structure which consists five hidden layers each with 32 neurons is replicated at Tx, Rx1 and Rx2. The Tx, however, has a side block (Sub-Network 2) which is used to adjust power allocation for each symbol.

II. SYSTEM MODEL

Consider a downlink NOMA system involving a base station (BS) communicating with two users. All nodes are equipped with single antennas. The user with weaker (stronger) channel is referred to as the weak (strong) user. The channels between the BS and the weak and strong users are respectively denoted by h_1 and h_2 satisfying $|h_1|^2 \leq |h_2|^2$. Both channel gains are known at the BS, but each user only knows its own channel gain. All channel gains are complex-valued and independently drawn from a continuous distribution.

In a NOMA system, the BS employs superposition coding to concurrently transmit messages to both users. The transmitted signal x is defined by $x = \sqrt{\alpha P} s_1 + \sqrt{\bar{\alpha} P} s_2$, where s_1 and s_2 are independent and identically distributed complex Gaussian signals with zero mean and unit variance, $\mathcal{CN}(0, 1)$. Here, P represents the BS's transmit power budget, $\alpha \in [0, 1]$ is the fraction of power allocated to the weak user, and $\bar{\alpha} \triangleq 1 - \alpha$. The received signals at the users are expressed as

$$y_k = h_k x + n_k, \quad k \in \{1, 2\} \quad (1)$$

where the noises n_1 and n_2 are independent and identically distributed circularly-symmetric complex Gaussian random variables with zero mean and unit variance, $\mathcal{CN}(0, 1)$.

The weak user decodes its message by treating the interfering signal from the other user as noise. In contrast, the strong user first decodes the weak user's message, treating its own interfering signal as noise, and then applies SIC to decode its own message. The combined use of superposition coding and SIC decoding with Gaussian codebooks results in achieving the capacity region of this channel.

III. NETWORK STRUCTURE

A. Autoencoder's Structure

An autoencoder consists of two parts, an encoder and a decoder. The AE processes a binary input vector $\mathbf{s} \in [0, 1]^d$

by first mapping it to a hidden representation $\mathbf{u} \in [0, 1]^{d'}$ through a deterministic mapping $\mathbf{u} = f_\theta(\mathbf{s}) = \sigma(W\mathbf{s} + \mathbf{b})$, where $\theta = \{W, \mathbf{b}\}$, W is a $d' \times d$ weight matrix, and \mathbf{b} is a bias vector. Here, $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function and $\sigma(\mathbf{x}) = [\sigma(x_1), \dots, \sigma(x_d)]^T$. The resulting latent representation \mathbf{u} is then transformed back to a reconstructed vector $\hat{\mathbf{s}} \in [0, 1]^d$ in the input space, given by $\hat{\mathbf{s}} = g_{\theta'}(\mathbf{u}) = \sigma(W'\mathbf{u} + \mathbf{b}')$ with $\theta' = \{W', \mathbf{b}'\}$. For each training sample $\mathbf{s}^{(i)}$, the AE maps it to a corresponding $\mathbf{u}^{(i)}$ and a reconstruction $\hat{\mathbf{s}}^{(i)}$.

The proposed AE-NOMA network, illustrated in Fig. 1, consists of one transmitter (Tx) encoder and two decoders each at one of the receivers, i.e., Rx1 and Rx2. Each encoder and decoder, include a fully connected neural network (FCNN) composed of a sequence of fully connected layers and some residual connections. As seen in Fig. 1, the encoder (Tx) comprises a main block and a secondary network, referred to as Sub-Network 1 and Sub-Network 2, respectively.

This main block, replicated at Tx, Rx1 and Rx2, consists of an input layer, five hidden layers, and an output layer. The hidden layers contain 32 neurons each, while the final layer (output layer) has two neurons representing the in-phase (I) and quadrature-phase (Q) components of the symbol designed by the autoencoder for transmission. In addition to the main block, the Tx has a side block (Sub-network 2) which helps to better adjust the I and Q components under a given average power constraint, thereby optimizing the use of the I/Q plane in a way similar to a quadrature amplitude modulation (QAM).

In the main block, we have also implemented residual connections to enhance learning capacity and improves performance without the need for additional parameters or a wider network. These shortcuts also play a crucial role in preserving gradients within the network. We have used the activation function *exponential linear unit* which provides a smooth, non-zero output for negative inputs, and can help improve learning stability and performance.